Multiobjective Optimization of Crewed Spacecraft Supportability Strategies

by

Andrew Charles Owens

B.S., Rice University (2012)
S.M., Massachusetts Institute of Technology (2014)

Submitted to the Department of Aeronautics and Astronautics
in partial fulfillment of the requirements for the degree of

Doctor of Philosophy in Space Systems

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

February 2019

Author . . . .
Department of Aeronautics and Astronautics
January 23, 2019

Certified by . . . .
Olivier L. de Weck
Professor of Aeronautics and Astronautics and Engineering Systems
Thesis Supervisor

Certified by . . . .
Jeffrey A. Hoffman
Professor of the Practice of Aeronautics and Astronautics

Certified by . . . .
Brian C. Williams
Professor of Aeronautics and Astronautics

Certified by . . . .
William M. Cirillo
Senior Research Engineer, NASA Langley Research Center

Accepted by . . . .
Sertac Karaman
Associate Professor, Aeronautics and Astronautics
Chair, Graduate Program Committee

Multiobjective Optimization of Crewed Spacecraft Supportability Strategies

by

Andrew Charles Owens

Submitted to the Department of Aeronautics and Astronautics
on January 23, 2019, in partial fulfillment of the
requirements for the degree of
Doctor of Philosophy in Space Systems

Abstract

Future crewed missions present a logistical challenge that is unprecedented in human spaceflight. Astronauts will travel farther from Earth than ever before, and stay in space for longer, without access to regular resupply or the option to abort and return home quickly in the event of an emergency. Under these conditions, supportability – that is, the set of characteristics of a system that drive the amount of resources required to enable safe and effective system operations – will be a much more significant driver of system lifecycle properties than it has been in the past. Many of the strategies that are currently used to mitigate risk for human spaceflight will no longer be available, feasible, or effective. To enable the human exploration missions of the future, new supportability strategies must be identified, characterized, developed, and implemented.

This dissertation addresses this problem by developing and presenting a new methodology for modeling and multiobjective optimization of supportability strategies, minimizing mass and maintenance crew time requirements subject to constraints on risk. The supportability strategy optimization problem is encoded as a multiobjective Constraint Optimization Problem (COP), with a set of decision variables defining a range of supportability strategy options related to level of maintenance, On-Demand Manufacturing (ODM), commonality, redundancy, and distributed functionality. A supportability model is developed which enables evaluation of the mass and crew time associated with a given assignment to those decision variables, or the lower bounds on those objective values associated with a partial assignment.

The resulting model, called Mass, Crew time, and Risk-based Optimization of Supportability Strategies (MCROSS), advances the state of the art in space mission supportability analysis by enabling holistic, rapid, multiobjective optimization and evaluation of the tradeoffs between mass, crew time, and risk for future missions. Model outputs are verified against results from Monte Carlo simulation, and validated via comparison to an existing state-of-the-art NASA supportability model and to flight maintenance data from the International Space Station (ISS). MCROSS is then demonstrated using two case studies, one based on a notional system and the other examining the ISS Oxygen Generation Assembly (OGA). The notional case study is used to validate optimization results against the Pareto frontier identified via full enumeration. The second case study demonstrates the application of this methodology to a real-world system, showing that MCROSS can identify supportability strategies offering lower mass and crew time options than current approaches. A series of sensitivity analyses are also presented to demonstrate the application of MCROSS in an iterative design process. These results, and the associated analysis capability, provide a powerful analysis tool that can help inform system development and mission design by characterizing tradeoffs between mass, crew time, and risk, along with the underlying strategy decisions. The results and implications of this research are discussed, along with assumptions and limitations. Finally, the contributions of this research are summarized along with potential areas of future work.

Thesis Supervisor: Olivier L. de Weck
Title: Professor of Aeronautics and Astronautics and Engineering Systems

Acknowledgments

Many people have supported me and made this research possible over the past six and a half years. It is impossible to list them all here, but I would especially like to thank:

My advisor, Oli, who welcomed me into this group back in 2012, provided valuable feedback on this dissertation, and has given me the opportunity to thrive in this research as I explore space mission supportability.

My committee, Brian, Jeff, and Bill, and thesis readers Dave and Chel, who have also provided valuable insights, advice, and feedback for this research.

All of the other professors, administrators, and staff at MIT who have helped along the way, especially Julie, Beth, and Bill.

Everyone in the NASA family who has helped over the years, gathering data, refining models, discussing research, discussing the future of human spaceflight, and hosting me as a visiting researcher. Special thanks goes out to Molly, Sara, Mike, Niki, and many others in the Space Mission Analysis Branch, Crew and Thermal Systems Division, In-Space Manufacturing Project, ISS Program Office, Reliability and Maintainability Working Group, ISS Operations Support Officers, Advanced Exploration Systems, Space Technology Mission Directorate, and Human Exploration and Operations Mission Directorate. You have all helped me gain a greater understanding of the context we operate in and the challenges of future missions (technical and otherwise), and I look forward to continuing to work with you to build the future of human space exploration.

I'd like to give a very special thanks to everyone in the Space Mission Analysis Branch at Langley Research Center, especially Bill, who has been an incredible mentor through the NSTRF program, and Kandyce and Chel, who along with Bill have provided invaluable guidance as I developed this research. I'd also like to especially thank Paul and Kevin, who have been extremely supportive as I finished this research and prepare to transition into the workforce. The SMAB family is an incredible group of people, and I look forward to joining you all soon for more strategic analysis,

concept studies, mission development, and – of course – plenty of board games.

The community at the International Conference on Environmental Systems, which has provided excellent feedback on this research over the years.

Eric, Jingkai, and Simon, who provided invaluable help with setting up OpSat to work with my evaluation code.

Friends, near and far, who have provided support, feedback, and encouragement, especially Veronica, Sam, Jess, Anne, Matt, Alex, George, Eric, Parker, Sydney, Narek, Margaret, Marc, Bobby, David, Jack, Steven, Julie, David, Irene, Gwen, Nalin, Larissa, Matt, Kevin, Chris, Melanie, and all the others, too numerous to name, from SERG, 33-409, Festivus, Hampton Roads board gamers, Lionspool FC, D&D, Green Street, Ashdown, and AeroAstro.

Finally, I'd like to thank Mom, Dad, Michael, and all of my other family members who have encouraged and supported me throughout this long process. You taught me to dream big and work hard, and now here I am – doing what I've dreamed of doing for as long as I can remember.

This dissertation is dedicated to Granddaddy, who passed away in 2016, and Abia, who passed away in 2019.

It is also dedicated to my niece, Emily – born on the anniversary of the first time a human flew into orbit, she'll see and do more incredible things in her life than we can even begin to imagine today.

This research was supported by a NASA Space Technology Research Fellowship, grant number NNX14AM42H. My graduate tuition has also been supported by grants from the Emerging Space Office and the Hitachi Autonomous Driving Systems Architecture Tradeoff Analysis project.

"It is good to renew one's wonder," said the philosopher.
"Space travel has again made children of us all."

Ray Bradbury
The Martian Chronicles (1950)

THIS PAGE INTENTIONALLY LEFT BLANK

Contents

THIS PAGE INTENTIONALLY LEFT BLANK

List of Figures

THIS PAGE INTENTIONALLY LEFT BLANK

List of Tables

THIS PAGE INTENTIONALLY LEFT BLANK

List of Algorithms

THIS PAGE INTENTIONALLY LEFT BLANK

Symbols and Acronyms

$\hat{\imath}$	Imaginary Unit ($\sqrt{-1}$)
$\mathbb{R}$	Set of Real Numbers
$\mathbb{Z}_{>0}$	Set of Positive Integers
$\mathbb{Z}_{\geq 0}$	Set of Non-Negative Integers
α	Gamma Shape Parameter
β	Gamma Scale Parameter
γ	Mass Scaling Factor
$\gamma_{\Lambda r}$	Redundancy Effectiveness Factor
$\gamma_{\Lambda odm}$	On-Demand Manufacturing Failure / Action Rate Factor
γ_{odm}	On-Demand Manufacturing Mass Overhead Factor
γ_r	Redundancy Mass Overhead Factor
Δm	On-Demand Manufacturing Feedstock Mass Discretization Factor
Δt	Crew Time Discretization Factor
ε	Failure Rate Error Factor
ε_C	Crew Action Error Factor
ϵ_{max}	Maximum Numerical Characteristic Function Inversion Error
ϵ_{pt}	Point-Wise Error for Numerical Characteristic Function Inversion
$\epsilon_{pt,max}$	Maximum Point-Wise Error for Numerical Characteristic Function Inversion
ζ	Baseline Processing Rate
ζ'	Scaled Processing Rate
η_{max}	Maxmum Functional Distribution

$\hat{\eta}$	Total Number of Modules (Distributed Functionality)
θ	Component Indicator
ϑ	Variable Order
κ	K-Factor
Λ	Component Failure Rate
λ	Failure Rate (Deterministic Value)
$\bar{\lambda}$	Mean Failure Rate
$\bar{\lambda}_C$	Mean Crew Action Rate Rate
$\bar{\lambda}_\theta$	Component Effective Mean Failure Rate
$\bar{\lambda}_{C\theta}$	Component Effective Mean Crew Action Rate
Λ_C	Component Crew Action Rate
$\Lambda_{C\varrho}$	Replaceable Item Crew Action Rate
$\Lambda_{C\omega}$	Orbital Replacement Unit Crew Action Rate
Λ_ϱ	Replaceable Item Failure Rate
Λ_ω	Orbital Replacement Unit Failure Rate
μ	Lognormal Scale Parameter
ν	Maintenance Coverage Decision Variable
ξ	On-Demand Manufacturing Decision Variable
$\hat{\xi}$	Manufacturability
ξ_ϱ	On-Demand Manufacturing Decision (Replaceable Item)
ξ_ω	On-Demand Manufacturing Decision (Orbital Replacement Unit)
ϱ	Replaceable Item
σ	Lognormal Shape Parameter
τ	Mission Endurance
Υ	Gamma-Distributed Failure Rate Approximation
ϕ	Item List
ψ	Subsystem Indicator
ω	Orbital Replacement Unit List
A	Adjacency Matrix
a	Maintainability Parameter

C	Commonality Decision Variable
$\hat{C}$	Potential Commonality Matrix
C_U	Unique, Nonempty Rows of Commonality Matrix
D	Descendant Matrix
d	Duty Cycle
f_n	Probability Mass Function Estimate
i^+	Most Valuable Item (for Marginal Analysis)
ℓ	Component Life Limit
ℓ_ϱ	Replaceable Item Life Limit
ℓ_ω	Orbital Replacement Unit Life Limit
$\mathcal{M}$	Mission Description
m	Component Baseline Unit Mass
m'	Scaled Component Mass
m_{CM}	Corrective Maintenance Mass
$\hat{m}$	Spare Part Unit Mass
m_{inf}	Infimum Mass
m_{PM}	Preventative Maintenance Mass
m_{prev}	Mass of Previous Allocation
m_ϱ	Replaceable Item Unit Mass
m_S	System Mass
m^*	Optimal Spares Allocation Mass
m_{tot}	Total Mass
$\mathrm{E}[M_\phi]$	Item Expected Mass
m_ω	Orbital Replacement Unit Mass
N	Number of Corrective Maintenance Replacements (Random Variable)
n	Spares Allocation
N_C	Number of Corrective Maintenance Actions (Random Variable)
n^*	Optimal Spares Allocation
$N_{\Delta m}$	Number of Units of On-Demand Manufacturing Feedstock (Random Variable)

$N_{\Delta t}$	Number of Crew Time Units (Random Variable)
$n_{i,next}$	Next Allocation Level for Spare i
n^{LB}	Spares Allocation Lower Bound
n_o	Number of Observed Events (for Bayesian Updates)
n_{odm}	Number of Feedstock Units for On-Demand Manufacturing
n_{PM}	Number of Preventative Maintenance Actions
n_{POS_R}	Number of Units of Feedstock Requried to Achieve POS Requirement
$n_{\hat{s}}$	Number of Samples for Numerical Characteristic Function Inversion
n^{UB}	Spares Allocation Lower Bound
$\mathcal{P}$	Modeling Parameters
p	Parent (for Indenture Structure Definition)
$\hat{p}$	Negative Binomial Parameter (Probability of Success)
POS_R	Required Probability of Sufficiency
POS_S	Probability of Sufficient Spare Parts
$POS_{S,prev}$	Probability of Sufficient Spares for Previous Allocation
POS_T	Probability of Sufficient Crew Time
Q	Quantity Matrix
q	Quantity
$\hat{q}$	Quantity Per Application
q_{ϱ}	Replaceable Item Quantity
q_{ω}	Orbital Replacement Unit Quantity
R	Redundancy Matrix
r_i	Redundancy Decision Variable
r_{max}	Maximum Redundancy
$\mathcal{S}$	System Description
$\hat{s}$	Sampling Vector for Numerical Characteristic Function Inversion
T	Total Crew Time Required for Maintenance (Random Variable)
t	Crew Time Required for Maintenance (Component)
T_{CM}	Crew Time Required for Corrective Maintenance (Random Variable)
t_{max}	Maximum Allowable Crew Time for Maintenance

t_o	Operating Time (for Bayesian Updates)
t_{PM}	Crew Time Required for Preventative Maintenance
t_ϱ	Crew Time Required for Maintenance (Replaceable Item)
$\mathrm{E}[T_\phi]$	Item Expected Crew Time
t_ω	Crew Time Required for Maintenance (Orbital Replacement Unit)
u	Level of Maintenance Decision Variable
v_i	Marginal Value
W	Replaceable Item Content Matrix
$\mathcal{X}$	Supportability Strategy Decision Variables
$\circ$	Hadamard Product (Element-wise Matrix Multiplication)
$^{\circ -1}$	Element-wise Matrix Inverse
$^{\circ 2}$	Element-wise Matrix Square
$\lceil \cdot \rceil$	Ceiling (Rounding Up)
$\lfloor \cdot \rfloor$	Floor (Rounding Down)
$\Gamma(\cdot)$	Gamma Function
$\varphi_X(s)$	Characteristic Function for a Random Variable X
$\mathrm{diag}(\cdot)$	Diagonalization
$\mathrm{E}[\cdot]$	Expected Value
$\exp[\cdot]$	Element-wise Matrix Exponential
$F_X(x)$	Cumulative Distribution Function for a Random Variable X
$f_X(x)$	Probability Density Function for a Random Variable X
$\mathcal{F}$	Discrete Fourier Transform
$\mathcal{F}^{-1}$	Inverse Discrete Fourier Transform
$\hat{f}(\cdot)$	Fourier Transform
$\ln[\cdot]$	Element-wise Matrix Natural Logarithm
$\max_R$	Maximum Taken Along Rows of Matrix
$\mathrm{Var}[\cdot]$	Variance
$\mathrm{Gamma}(\alpha,\ \beta)$	Gamma Random Variable
$\mathrm{Logn}(\mu,\ \sigma)$	Lognormal Random Variable
$\mathrm{NB}(r,\ p)$	Negative Binomial Random Variable

Poisson$(\lambda\tau)$	Poisson Random Variable
AES	Advanced Exploration Systems
AIAA	American Institute of Aeronautics and Astronautics
ALSSAT	Advanced Life Support Sizing Analysis Tool
AM	Additive Manufacturing
ARFTA	Advanced Recycle Filter Tank Assembly
ASA	American Statistical Association
ASM	Aircraft Sustainability Model
BRAWL	Bayesian Reliability Average Weighted Likelihood Prior
CDF	Cumulative Distribution Function
CFR	Constant Failure Rate
CMAM	Comparative Maintenance Analysis Model
CM-h	Crewmember-Hour
CMMTTR	Crew Member Mean Time To Repair
COP	Constraint Optimization Problem
DA	Distillation Assembly
DRA	Design Reference Architecture
EBO	Expected Backorder
ECLSS	Environmental Control and Life Support System
ECTM	Exploration Crew Time Model
EFA	External Filter Assembly
EMAT	Exploration Maintainability Analysis Tool
EMTT	External Maintenance Task Team
EVA	Extravehicular Activity
EXAMINE	Exploration Architecture Model for In-Space and Earth-to-orbit
FCA	Firmware Controller Assembly
FCPA	Fluids Control and Pump Assembly
FDM	Fused Deposition Modeling

GA	Genetic Algorithm
GAO	Government Accountability Office
GRDF	Generalized Resilient Design Framework
h	Hour
HEO	Human Exploration and Operations
ICES	International Conference on Environmental Systems
ICOM	Integrated Crew Operations Model
IDFT	Inverse Discrete Fourier Transform
IMLEO	Initial Mass in Low Earth Orbit
ISM	In-Space Manufacturing
ISRU	In-Situ Resource Utilization
ISS	International Space Station
JSC	Johnson Space Center
kg	Kilogram
L&M	Logistics and Maintenance
LaRC	NASA Langley Research Center
LBP	Lower Bounding Principle
LEO	Low Earth Orbit
LMI	Logistics Management Institute
LoC	Loss of Crew
LoM	Loss of Mission
LoV	Loss of Vehicle
LRU	Line-Replaceable Unit
MADS	Modeling Analysis Data Set
MCROSS	Mass, Crew time, and Risk-based Optimization of Supportability Strategies
MDC	Maintenance Data Collection
METRIC	Multi-Echelon Technique for Recoverable Item Control

MFOP	Maintenance Free Operating Period
MFOPS	Maintenance Free Operating Period Survivability
MMOD	Micrometeoroids and Orbital Debris
MPI	Mutually Preferential Independence
MSFC	Marshall Space Flight Center
M-SPARE	Multiple Spares Prioritization and Availability to Resource Evaluation
MTBF	Mean Time Between Failures
NASA	National Aeronautics and Space Administration
NB	Negative Binomial
NRC	National Research Council
ODM	On-Demand Manufacturing
OGA	Oxygen Generation Assembly
OPTimIS	Operations Planning Timeline Integration System
ORU	Orbital Replacement Unit
P(LoC)	Probability of Loss of Crew
P(LoM)	Probability of Loss of Mission
P(LoV)	Probability of Loss of Vehicle
PACT	Probability and Confidence Trade-space
PCPA	Pressure Control and Pump Assembly
PDF	Probability Distribution Function
PMF	Probability Mass Function
POS	Probability of Sufficiency
PRA	Probabilistic Risk Assessment
PSN	Probability of Spare when Needed
QPA	Quantity Per Application
R&R	Remove & Replace
RAP	Redundancy Allocation Problem
RFTA	Recycle Filter Tank Assembly

RMAT	Reliability and Maintainability Assessment Tool
RMWG	Reliability and Maintainability Working Group
RSA	Rotary Separator Accumulator
SMAB	Space Mission Analysis Branch
SPA	Separator Plumbing Assembly
SRU	Shop-Replaceable Unit
SSF	Space Station *Freedom*
SSM™	Spacecraft Sustainability Model™
UAV	Unmanned Aerial Vehicle
UGF	Universal Generating Function
UPA	Urine Processor Assembly
WPA	Water Processor Assembly
WRS	Water Recovery System
WSTA	Wastewater Storage Tank Assembly

THIS PAGE INTENTIONALLY LEFT BLANK

Chapter 1

Introduction, Background, and Motivation

Future crewed missions present a logistical challenge that is unprecedented in human spaceflight. Astronauts will travel farther from Earth than ever before, and stay in space for longer, without access to regular resupply or the option to abort and return home quickly in the event of an emergency. Under these conditions, supportability – that is, the set of characteristics of a system that drive the amount of resources required to enable safe and effective system operations – will be a much more significant driver of system lifecycle properties than it has been in the past. Epistemic uncertainty, or uncertainty in the values of key parameters such as component failure rates, will be a more significant driver of risk on these longer missions, and must be considered during mission planning; analyses that neglect epistemic uncertainty will tend to underestimate the amount of resources required for a given mission. Finally, many of the strategies that have been used in the past and are currently used to mitigate risk for human spaceflight missions will no longer be available, feasible, or effective. New supportability strategies will need to be identified, characterized, developed and implemented to enable the human exploration missions of the future [1–10].

This dissertation addresses this problem by developing and presenting a methodology for modeling and multiobjective optimization of supportability strategies, minimizing mass and maintenance crew time requirements subject to constraints on the allowable level of risk. The supportability strategy optimization problem is formulated as a multiobjective Constraint Optimization Problem (COP), with a set of decision variables defined to encode options for a wide range of supportability strategies including level of maintenance, On-Demand Manufacturing (ODM), commonality, redundancy, and distributed functionality. A supportability model interprets assignments to those decision variables in order to calculate key supportability characteristics for a given system and mission. These characteristics are defined in greater detail in this chapter, and include the expected amount of crew time required for maintenance as well as the total mass (including system, spares, and maintenance items) required to achieve a desired level of risk mitigation, including risks from both aleatory and epistemic uncertainty. This risk requirement is captured in terms of a required Probability of Sufficiency (POS), defined in the following section.

The resulting analysis tool, called Mass, Crew time, and Risk-based Optimization of Supportability Strategies (MCROSS), advances the state of the art in space mission supportability analysis by enabling holistic, rapid, multiobjective evaluation of the tradeoffs between mass, crew time, and risk. The underlying supportability model provides improved estimates of maintenance crew time by separating its demand process from the one used for spare parts, and includes the risk of insufficient crew time in overall risk assessments. A novel methodology for assessing ODM feedstock mass allocations and their associated POS is also included, which allows ODM to be considered alongside traditional, discrete spares during maintenance resource allocation. In addition, MCROSS performs all required calculations without using Monte Carlo analysis. This means that candidate solutions are evaluated very rapidly, facilitating large-scale supportability strategy tradespace exploration and optimization. Overall, the methodology presented in this dissertation provides a powerful analysis tool to help inform technology development, system design, and mission planning and enable human exploration beyond Low Earth Orbit (LEO).

This chapter motivates this research and provides key background information regarding the supportability challenges of future human spaceflight. Section 1.1 defines supportability and its impacts on system development, mission planning, and operations. Section 1.2 defines and describes the concept of a supportability strategy, which is the set of architecture, design, and operational decisions that influence the supportability of a system. Section 1.3 describes past experience with human spaceflight mission supportability and compares that history to the challenges presented by future missions, and Section 1.4 discusses the need for formal optimization of supportability strategies. Finally, Section 1.5 summarizes the identified capabilities needed to solve the supportability problem, characterizing the domain motivation for this research.[a]

1.1 Supportability

Supportability is a term describing the ease with which a particular system can be supported during a particular mission – that is, the ease with which the system can be operated and maintained, and the amount of logistics and other resources required to support those tasks. A system's supportability is a function of a broad set of system characteristics that drive the the amount of logistics and support resources required to enable safe and effective system operations. These resources include physical resources such as spares, maintenance items, and consumables required to enable system functionality, as well as temporal resources such as crew time for maintenance activities. Some resources demands are deterministic, such as regularly scheduled maintenance. In this research, these deterministic, regularly scheduled demands are referred to as preventative maintenance. Other demands, however, are stochastic, driven by random failures and the associated corrective maintenance. The response to random failures is referred to as corrective maintenance. Replacement items used for preventative maintenance are referred to as maintenance items, while those used for corrective maintenance are referred to as spare parts [1, 2, 11].

The supportability characteristics of a system define the relationship between

[a] Portions of this background research were also published as a review paper at the 2017 American Institute of Aeronautics and Astronautics (AIAA) SPACE Forum [10].

the amount of resources required and the probability that those resources will be sufficient, known as POS [12, 13]. The challenge of supportability management is to create systems that balance risk and resource requirements to acceptable levels. A successful system has supportability characteristics that allow mission planners to mitigate risk to acceptable levels without requiring excessive amounts of resources.

Some examples of general system characteristics that influence supportability include:

- Reliability: the probability that an item will perform its intended function for a given period of time under a given set of operating conditions [1, 14–16].
- Maintainability: the ease with which a system can be maintained, as a function of the set of system, component, and crew characteristics that allow for or reduce the cost and difficulty of maintenance activities during operations [1–3, 16].
- Repairability:[b] the ease with which a component can be repaired (i.e. restored to functional status) by the crew after a failure occurs [1].
- Redundancy: the incorporation of multiple copies of a system element (which can be of similar or dissimilar design, as long as they accomplish the same function) in order to mitigate the impact of a failure on system operations [1, 16].
- Sparing Philosophy: the set of decisions regarding spare parts for a particular system and mission, including which items, assemblies, and/or components within the system are replaceable – meaning items that will be swapped out for spares when a failure occurs – as well as operational decision such as the number and type of spare parts manifested [1].

Each of these characteristics is determined by a set of interrelated design and operational decisions that must be made – whether explicitly or implicitly, consciously or unconsciously – during system development, deployment, and operations. Once made, all of these decisions together define the supportability of the system.

[b] Also spelled “reparability” by some sources.

It is important to note that supportability is a holistic, emergent characteristic of the entire system, and does not result from any single decision alone, nor is it fully described by the supportability characteristics of any single subsystem. Coupled, nonlinear interactions between the risk/resource trades for all system elements drive the overall supportability characteristics of the system. As a result, holistic systems analysis is required to assess supportability and understand the trades between risk and resources for a given system and mission.

1.1.1 Importance of Supportability Assessment During Development

Decisions made early in system development can have a very significant impact on overall system characteristics, particularly with regard to cost. For example, it is estimated that in general 80% of total system lifecycle cost will be "locked in" after only 20% of that cost has been spent [17]. From the perspective of supportability specifically, Montgomery [18] notes that operational support costs typically account for 60-80% of total program costs. By definition, these costs occur during operations, after a system has been completed and deployed. However, the decisions that defined those costs – or, more precisely, the cost-risk tradespace available to system operators – are often made much earlier, during concept development and system design activities.

Systems that neglect supportability and logistics considerations during the system architecture and design process may, once deployed, be significantly (and often unexpectedly) more expensive to operate than systems that account for supportability considerations during these early phases of the system lifecycle. Without careful analysis, some issues may not become apparent until after hardware is manufactured and integrated, by which point design changes may be prohibitively expensive to implement and more complex and expensive operations strategies may be the only remaining viable option to mitigating risk. For example, a lack of interchangeability and accessibility for some systems on the Space Shuttle increased the complexity of operations and maintenance activities after the system was deployed. However, early-stage supportability analysis and lessons learned from past experience can be used to inform the development of better systems. For example, partially as a result of the Space Shuttle supportability issues described above, logistics and supportability was more effectively included during early International Space Station (ISS) concept development [18].

One key example of the impact of supportability analysis on system design is the Space Station *Freedom* (SSF) External Maintenance Task Team (EMTT) Report, also known as the Fisher-Price study [19]. The EMTT was formed in 1990 after a preliminary analysis of external maintenance crew time demands for the SSF (a precursor to the ISS) found that, given the then-current system design, planned crew time allocations may be exceeded by an order of magnitude. The EMTT performed an in-depth analysis of the expected number of failures and amount of maintenance crew time that could be expected for SSF, based on estimates of system characteristics. That analysis provided valuable information to system designers by highlighting and characterizing the prohibitively high crew time demands of the then-current design and informing a set of recommendations for changes to system design and operations planning to reduce external maintenance crew time requirements [19, 20].

Supportability has already been a challenge for past human spaceflight efforts, and future exploration missions will be even more challenging, as described in Section 1.3. Under those conditions, supportability will be a much more significant driver of crewed space system lifecycle properties than it has been in the past. Strategies that have been effective for past missions may not be as effective – or even feasible – for future missions. It is therefore critical that supportability be carefully considered early in mission planning and system development efforts in order to enable safe and cost-effective human exploration of space beyond LEO [1, 3–7, 10].

1.1.2 Key Metrics

The three primary metrics associated with supportability are:

- Mass: the mass of spare parts and maintenance items, as well as any system mass that changes as a function of the specific supportability strategy applied.
- Crew Time: the amount of time, typically measured in crewmember-hours (CM-h), that the crew must spend on maintenance activities.
- Risk: the risk associated with maintenance, typically captured in terms of the probability that the amount of resources available (i.e. spares, maintenance items, and crew time) will or will not be sufficient for a given mission.

The supportability characteristics of a system define the relationship between these three metrics for a given mission. Each of these metrics is discussed in the following subsections. While these metrics tend to be primary considerations during supportability analysis, it is important to note that they do not capture all impacts of supportability-related decisions on a given mission. Additional considerations that can impact mission-level metrics (i.e. cost, risk, performance, and schedule) are discussed in Section 1.1.3.

A typical approach taken in supportability assessment (e.g. [3, 7, 12, 21–23]) is to specify a required POS and determine the amount of resources required to achieve that POS. Alternatively, the amount of mass and/or crew time available could be specified and the associated POS calculated. Descriptive statistics of resource requirements such as median or expected values could also be used to describe resource requirements (e.g. [19]), though this approach does not rigorously examine risk. Overall, there are many different approaches to examining supportability and using that information to guide the development of better systems and missions.

Mass

One of the primary impacts that supportability has on mission architecture is its impact on mass, and maintenance logistics mass in particular. Maintenance logistics mass – meaning the mass of spare parts and maintenance items – is a significant driver of overall logistics for overall logistics mass, particularly for long, logistically isolated missions [1, 3]. The supportability characteristics of a system determine the amount of maintenance logistics that are required to support a system to a desired POS, or, conversely, the level of POS that can be achieved within a given logistics mass constraint. For exploration systems, one objective of system design is to minimize logistics mass, since reduced logistics can enable reduced launch requirements, less expensive/complex in-space transportation systems, and/or an increased cargo capacity for science and utilization objectives, thereby increasing the value of the mission or reducing its cost to feasible levels. In fact, mass reduction may be necessary to enable a mission in the first place, particularly for destinations like Mars, which have very high transportation costs and delta-v requirements [1, 3, 5].

In addition, some strategies for improving the supportability characteristics of a system may increase the mass of the system itself. For example, redundant components could be added in order to allow the system to continue operating after a failure. These redundant components may or may not increase the maintenance logistics mass associated with a system, since the mass of spares and maintenance items may be higher, but the number of replacements required may be lower. However, they do increase the initial system mass. In addition, some supportability strategies may require that additional equipment be added to the system. For example, ODM could reduce the logistics mass required to cover random failures by allowing undifferentiated raw materials to be converted into spare pats as needed, but this approach typically requires some kind of manufacturing system, which increases initial system mass [24, 25]. In general, it is important to consider the impacts of supportability decisions on baseline system mass in addition to logistics.

Crew Time

Crew time is a critical, valuable, and limited resource during space missions. The crew must spend time to support utilization activities such as science and exploration, since these activities which are typically a key part of the value proposition of the mission. The crew must also perform a range of activities required to maintain crew health, including sleeping, eating, and exercise. However, crew time is also necessary for spacecraft maintenance. Any time spent on maintenance detracts from time available for utilization, and therefore one objective of supportability design is to minimize the amount of crew time spent on maintenance [1, 3, 7, 8, 26, 27].

Experience with maintenance on LEO space stations has shown that maintenance crew time is a significant challenge. In practice, maintenance crew time demands have been higher than expected, and as a result system productivity has been lower [8, 26, 27]. For example, the amount of crew time associated with maintenance of the ISS Environmental Control and Life Support System (ECLSS) during station operations has exceeded the amount specified in design documents by more than an order of magnitude. It was initially estimated that only approximately 1 hour per week of crew time would be required, but ISS ECLSS maintenance has historically accounted for approximately 13 to 15 hours of crew time per week [26, 27]. ISS flight maintenance data, presented as part of model validation in Chapter 5, indicate that at least part of this significant crew time underestimate is likely driven by underestimates of both the amount of crew time required per maintenance action and the number of maintenance actions required, as captured by component failure rates.

Each time a failed item is replaced, maintenance crew time is required to execute the maintenance action and install a new spare. These are known as Remove & Replace (R&R) events. However, it is important to keep in mind that maintenance crew time can be used even if a spare or maintenance item is not used. For example, some systems may require troubleshooting, inspection, or cleaning, actions that do not require R&R but nevertheless can contribute a significant amount to overall crew time demand. Examination of the aforementioned ISS flight maintenance data (described in greater detail in Chapter 5) shows that 26% of maintenance crew time spent on key ECLSS systems over an eight-year period from 2008 to 2016 was related to events that did not involve R&R of a failed item. Every maintenance action that expends mass resources (i.e. spares and maintenance items) also requires crew time, but not every maintenance action that requires crew time also requires mass resources. When assessing maintenance crew time demands, it is important to include these non-R&R events in order to avoid underestimating crew time. Failure rates are typically used to describe the distribution of the number of times that a particular item will fail and require replacement. This research introduces an analogous crew action rate (described in greater detail in Chapters 3 and 4) to describe the distribution of the number of times that the crew must spend maintenance crew time on an item, regardless of whether R&R was required.

The high maintenance crew time demand experienced on the ISS is indicative of a particular challenge for future missions, given that the ISS supportability strategy was specifically designed to minimize crew time spent on maintenance and repair by packaging components into Orbital Replacement Units (ORUs). This approach saves crew time by simplifying maintenance and repair activities, but it also increases logistics mass by reducing the mass efficiency of spare parts allocations [8, 28]. Implementing maintenance at a lower level is a commonly-discussed supportability strategy (discussed further in Section 1.2.1) that can enable more efficient maintenance and repair logistics, thereby reducing logistics mass; however, it also tends to have the effect of increasing the amount of crew time necessary for maintenance and repair activities [1, 29]. Similarly, the use of redundant components could reduce crew time by allowing systems to recover from a failure without requiring a crew action, but those redundant components tend to increase the mass of the system. While shifting to a lower level of maintenance, or other changes to strategy, may provide much-needed reduction in spares mass for future missions, any new approaches to supportability must carefully consider their impact on crew time, and by extension their impact on mission productivity and performance [1, 7].

Risk

Supportability is fundamentally related to the tradeoff between risk and resources, and therefore assessment and mitigation of risk is a core element of supportability management. Risk is defined as the combination of the probability that an event will occur and the impact that will result if it does occur [30]. For human spaceflight, risk is typically evaluated via estimation of the probabilities associated with a set of particular events, sometimes called outcomes, using Probabilistic Risk Assessment (PRA). At a high level, the three outcomes most often examined for crewed missions are Loss of Mission (LoM), Loss of Vehicle (LoV), and Loss of Crew (LoC). The associated probabilities are, respectively, Probability of Loss of Mission (P(LoM)), Probability of Loss of Vehicle (P(LoV)), and Probability of Loss of Crew (P(LoC)); these probabilities are key risk metrics for human spaceflight missions. Mission planning will often involve the specification of a threshold on one or several of these probabilities in order to define a requirement [31]. For example, Apollo program risk requirements specified that P(LoM) be no greater than 0.01 and P(LoC) no greater than 0.001 on any given mission [32]. NASA's Commercial Crew Transportation System certification requirements specified that the mean P(LoM) and P(LoC) be no greater than 1 in 55 (0.0182) and 1 in 270 (0.0037), respectively, for a 210-day ISS mission [33]. (Note that, in that application mean values of an uncertainty distribution are used, with that distribution representing epistemic uncertainty on the actual probability.) This probability requirement can then be used to inform design/operational decisions and logistics requirements, or a range of sensitivity analyses can be used to examine interactions between design decisions, risk, and operational cost/performance.

The relevant critical event in the context of supportability analysis is a failure to maintain the system because of a lack of spare part or because of insufficient crew time. This risk is typically described using POS, defined as the probability that the amount of maintenance resources provided for a given mission is sufficient to cover all maintenance demands during that mission [12, 13]. In traditional usage, POS refers to the probability of having sufficient spare parts; however, for the purposes of this research the term is expanded to include the probability of having sufficient crew time to perform maintenance. Note that, while the probability associated with risk is the probability of a negative event occurring (in this case, insufficient spares or crew time), POS is defined as the probability that the negative event does not occur – that is, POS is a probability of success, not failure. For long, logistically isolated missions with limited abort and resupply capability, the ability of the crew to maintain systems is much more critical than it has been in the past, since a failure to maintain systems could result in LoC [1, 4]. The system cannot be maintained unless there are sufficient spares and crew time for maintenance, and therefore (in the absence of contingency options) POS can be thought of as a bound on P(LoC):

$$P(LoC) \geq 1 - POS \tag{1.1}$$

If contingency options exist, such as an abort capability that can return the crew safely to Earth by ending the mission prematurely, then POS can be thought of as providing a bound on P(LoM) rather than P(LoC). In either case, a P(LoC) or P(LoM) requirement can similarly be thought of as a bound on required POS, for planning purposes. For example, if a mission to Mars has a requirement that P(LoC) be no greater than 1 in 1000 (i.e. the probability of crew survival must be greater than 0.999), then in the absence of contingency options the POS associated with critical systems such as ECLSS must be greater than 0.999. Otherwise, the probability of failed maintenance due to lack of resources – and therefore loss of system function, resulting in loss of crew when there are no contingency options – would be higher than the specified acceptable risk. If there are contingency options such as secondary systems or open-loop life support supplies on board, then the POS associated with those options must also be considered, given the lack of abort and resupply options for Mars missions.

In general, the consequence associated with a failure to execute all required maintenance can be mapped to LoM, LoV, or LoC as a function of the characteristics of the particular mission being examined, and the relationship between supportability risk and overall mission risk should be adapted to the circumstances at hand. The strategy used to address POS should also be informed by mission context, specifically with regard to the cost associated with reducing the diverse set of risks that contribute to P(LoC), P(LoM), and/or P(LoV). POS is only one source of risk for deep-space missions, and it is addressed primarily via spares selection, crew time planning, and the range of supportability decisions made during development. Other sources of risk, such as those associated with launch, landing, and in-space propulsion systems, are addressed by other investments. Decisions related to all of these risk factors should be balanced in order to achieve mission objectives within budget and schedule constraints.

1.1.3 Other Mission-Level Impacts

In addition to the specific mass, crew time, and risk metrics discussed in Section 1.1.2, supportability impacts overall mission characteristics in a number of ways, described in the following subsections. Supportability impacts are binned in terms of their impact on high-level mission cost, performance, schedule, and risk. This section is not intended to be an exhaustive discussion of supportability impacts. Instead, it presents an overview of key areas that should be considered during supportability analysis and mission planning in order to provide context for this research.

Cost

As noted in Section 1.1.2, one of the primary impacts that supportability has at the mission level is the relationship between risk and logistics mass. Mass is a commonly-used proxy metric for cost in space mission design, due to the (traditionally) high cost associated with launch into space and the difficulty associated with moving massive payloads between different destinations in space [34]. Initial Mass in Low Earth Orbit (IMLEO) – which represents the total mass that must be launched into LEO for a given mission, including systems, logistics, and propellant – is typically used as a cost proxy for interplanetary missions such as Mars campaigns. For example, IMLEO was a key metric used to evaluate options for NASA's Design Reference Architecture (DRA) 5 [35–37].

However, It is important to bear in mind that mass is an imperfect proxy for cost, since different elements of the same mass may have significantly different costs. In particular, raw consumables such as water and oxygen are likely significantly cheaper per unit mass than complex spare parts and other components for systems designed to recycle water. In addition, recent shifts in the launch industry have begun to reduce the average cost per kilogram of sending payloads to orbit, reducing the importance of launch mass as a factor in overall mission cost [38, 39]. While mass-based analysis provides a means to assess the relative costs of various options – particularly for early concept studies and mission design activities – without requiring detailed cost assessment of the elements within them, these results must always be considered in the context of the actual cost required to develop and procure the desired hardware.

Supportability-related changes to system design may incur significant development costs, including both the design (and/or redesign) of systems for maintainability/reliability as well as testing to confirm the supportability characteristics of the system. While more complex systems may provide improved supportability characteristics, the expense involved in developing them and demonstrating that they function to the desired level of reliability may outweigh the cost associated with a simpler, heavier system. For supportability in particular, the amount of test time required to characterize and validate a system's supportability characteristics can be very significant, and therefore expensive. While mass is the primary cost metric used in this dissertation, programmatic decisions related to technology investment and system design must always be careful to ensure that the cost of developing a new capability is justified by the benefits that the capability can provide.

Performance

One core objective of human exploration missions is the safe delivery a crew to an exploration destination in order to perform utilization activities such as completing exploration objectives and scientific investigations. Utilization activities output is one metric for mission performance; the more utilization that can be done, the more productive and higher-performing the mission is. Utilization activities require time from the crew, and therefore (as noted in Section 1.1.2) the amount of crew time available for these activities, or utilization time, is a metric for the performance of a given mission. Missions with more utilization time can achieve higher performance through greater crew productivity, which results in higher scientific output. Conversely, a reduction in available utilization time corresponds to a reduction in mission performance due to a reduction in productivity. If the crew is to perform utilization activities at an exploration destination, they need to have time to devote to those activities. Any crew time spent on maintenance activities reduces the amount of crew time available for utilization. In addition, excessive maintenance crew time can impede the crew's ability to prepare for and execute critical operations, thus potentially reducing the effectiveness of those operations and overall mission performance.

Maintenance demands to not simply impact the amount of crew time available for utilization, however. In the severely mass-constrained operating paradigm of human spaceflight, maintenance logistics also impact the amount of cargo available for utilization. Every kilogram of an exploration vehicle's cargo capacity that is used for maintenance is a kilogram of mass not spent on utilization. Conversely, every kilogram spent on cargo and utilization is a kilogram not spent reducing risk for the mission. A supportability strategy that reduces the amount of maintenance logistics required to meet a desired level of risk mitigation could allow a mission to carry more equipment for scientific investigation, technology demonstration, or other utilization purposes, thus increasing the performance and value of the mission. Good supportability strategies manage the balance between risk and logistics requirements in order to ensure that the needs of the mission are met in a cost-effective manner.

Schedule

Effective supportability analysis requires knowledge of the supportability characteristics of a system. One critical supportability characteristic for each component in the system is the failure rate, which is a parameter used to define the distribution of time to failure for that component and/or the distribution of the number of failures that will occur in a given period of time. Failure rate is sometimes also represented by its inverse, Mean Time Between Failures (MTBF), which represents the average time that a component will operate without failing. Unfortunately, failure rates cannot be measured directly. Instead, they must be estimated based on analogous systems and/or testing and operational experience. Increased uncertainty results in increased logistics mass requirements, as well as increased risk for future missions [40]. Testing time can reduce this uncertainty, but the impacts of testing campaigns on development schedules should be carefully considered.

1.1.4 Uncertainty Drives Risk

At its core, supportability analysis and design involves an assessment of tradeoffs between risk and resources and the impact of various design decisions on that balance. Risk is driven by uncertainty, which in this case is uncertainty in the number of times that a particular item will fail or require crew intervention. The exact number of spares or amount of maintenance crew time that will be required for a particular mission cannot be known before that mission occurs, but supportability analysis provides a method to assess the probability that a particular allocation of spares and crew time will be sufficient, and that POS can be traded against the cost associated with providing resources at those levels.

Aleatory and Epistemic Uncertainty

One particular challenge for space system supportability is the fact that both aleatory and epistemic uncertainty are present in the problem in significant amounts, and both must be accounted for in order to have an accurate assessment of risk. Aleatory uncertainty is the natural randomness inherent to a particular process, while epistemic uncertainty arises from a lack of knowledge about that process [31]. For example, if fair dice (or dice with a known bias) are rolled, the uncertainty associated with the outcome of that roll is aleatory. However, if the dice are loaded and the bias is unknown, then there is additional, epistemic uncertainty present in the outcome due to the lack of knowledge of the probabilities associated with each possible outcome.

In the context of supportability analysis, aleatory uncertainty is related to the distribution of the number of failures that a particular item will experience during the mission, or the number of maintenance actions (and therefore the amount of maintenance crew time) that will be required for that item, given a known failure rate or crew action rate. Epistemic uncertainty, on the other hand, describes the uncertainty in the value of the rate itself. Unlike other parameters (such as mass), failure rates and crew action rates are parameters describing random component behaviors, and cannot be directly measured. Instead, they must be estimated based upon comparison to similar items and/or statistical analysis of observed system behavior. This estimation is never perfect, and in the context of space systems there is often a relatively low amount of empirical data available and a correspondingly large amount of epistemic uncertainty present in a particular estimate.

Aleatory uncertainty is an inherent characteristic of the system itself which cannot be modified without making changes to the system, such as selection of different components or a different system design. Epistemic uncertainty, on the other hand, can be reduced through experience, even if the system itself remains unchanged. As more data are gathered – for example, as more system operating time is accumulated – estimates of failure and crew action rates can be refined to reduce the level of epistemic uncertainty present. For example, the ISS has provided a critical testbed for understanding the behavior of key systems in flight conditions, both in terms of validating system performance and in terms of providing data to refine failure rate estimates. Improvements in failure rate estimates enabled by ISS experience have had a significant impact on the projected spares required for future exploration missions, and have helped to uncover and mitigate a significant amount of previously-hidden risk in the system due to underestimated failure rates. However, even after nearly two decades of continuous crewed operations in LEO, a significant amount of uncertainty remains which must be accounted for in supportability analyses [40].

Both aleatory and epistemic uncertainty have significant impacts on the risk associated with a particular allocation of supportability resources. Analyses that neglect epistemic uncertainty tend to significantly underestimate risk, particularly for longer missions, and therefore underestimate the amount of resources that would be required to achieve a desired level of POS. Therefore, it is critical that supportability analysis and design efforts include epistemic uncertainty [9, 10, 41].

Unknown Unknowns

The supportability-related contributions to risk described above are only some of the many contributions to overall system risk, and analysis techniques that estimate POS or other risk metrics are only as valid as the data that are used as inputs, and only capture risks that are included in the model. Even when perfectly accurate data are used, these models can only account for known or anticipated factors. Unknown or unanticipated effects can have significant implications for system supportability that must be kept in mind during system development. For example, the ISS ECLSS system has experienced several instances of component failure and/or degraded performance due to unanticipated issues [25, 42], including significantly reduced component lifetimes due to dust and debris impingement [43, 44], seal material degradation [45], component manufacturing and assembly errors [45], or even changes in the concentration of calcium in crew urine due to physiological changes resulting from the microgravity environment [45, 46]. Distributions on the values of failure rates or other parameters can be used to account for some of this uncertainty, but unknown risks will always remain.

For future missions to new destinations, unknowns resulting from the exploration destination itself – such as resource availabilities and/or environmental conditions and their effects on the crew and mission hardware – will be even more prevalent. These "unknown unknowns" are, by definition, unknown. However, historically they have had a detrimental impact on system supportability by introducing additional risk factors that have to be dealt with using operational workarounds, often reducing the performance or increasing the logistics costs associated with a particular system. In many cases, the solutions that have been implemented to adapt to unforeseen circumstances were only possible because they occurred in LEO, where a solution developed using ground-based resources can be (relatively) easily shipped to orbit and supply chains can be adapted to provide additional logistical support if necessary. In a logistically isolated deep-space environment, these options will likely not be available, and therefore the impact of unknown unknowns will typically manifest in the form of additional, unanticipated risk. Therefore, probabilistic assessments of risk should be seen as a bound, rather than a guarantee; when risk factors beyond the scope of the analysis (known and unknown) are included, the actual risk will almost certainly be higher. In addition, system testing will be much more critical for these future missions in order to reduce uncertainty and try to identify these problems before they appear in a mission context, where there may be no workable solution.

1.2 Supportability Strategy Decisions

A supportability strategy is the collection of architecture, design, and operational decisions that influence the supportability of a system and, by extension, the system characteristics described in Section 1.1. These decisions each have different, often coupled, and sometimes conflicting impacts on mass, risk, crew time, and other characteristics of the system. They must be made at various points in the system development timeline, from concept generation and architecture synthesis, through design and testing, all the way to operations. Key supportability strategy decisions for human spaceflight missions are described and discussed in this section, based on descriptions in various sources and lessons learned from previous supportability analyses [1, 9, 10, 19, 25, 40, 41, 47].

1.2.1 Level of Maintenance

Level of maintenance, also known as indenture or level of repair, refers to the level in the parts hierarchy (system, subsystem, assembly, component, etc.) at which maintenance actions are executed [1, 12]. For example, the Air Force uses the terms Line-Replaceable Unit (LRU) and Shop-Replaceable Unit (SRU) to refer to items that are maintained on the flight line or in a maintenance shop, respectively [12]. Spaceflight applications typically refer to ORUs in the place of LRUs. Grouping components into ORUs can result in a more maintainable system by simplifying maintenance activities and reducing the amount of time required for maintenance, as discussed in Section 1.1.2 [8, 28]. However, higher-level maintenance reduces the mass-efficiency of maintenance; when components are grouped together into ORUs, the failure of a single component within that ORU can force the replacement of all components when the entire ORU is replaced. Items that may still have a significant amount of useful lifetime remaining before they fail are replaced because of the failure of some other item in the same ORU.

In contrast, lower-level maintenance can result in significant reductions in spares mass requirements, since lower-level maintenance is more mass-efficient. When maintenance actions can be executed at the component rather than assembly level, all other components within the assembly that are still functioning can remain in service and the mass of elements that are replaced is lower than it would be in a higher-level maintenance paradigm [1, 29]. In addition, there may be more potential for commonality between lower-level items such as valves, sensors, fans, hoses, and manifolds than there is between higher-level assemblies. Increased commonality enabled by lower level maintenance may provide additional benefits (see Section 1.2.2).

However, lower level maintenance can increase the complexity of system development and maintenance operations. In order to enable effective lower-level maintenance, systems must be designed to enable access to and replacement of lower-level components, rather than collecting components into convenient boxes. Design for maintainability will place additional constraints on the physical design of the system from the perspective of crew access, tool clearances, potential hazards (e.g. sharp edges, containment for toxic materials, etc.), and sensing requirements for diagnostics. Crews must have the knowledge and tools to execute maintenance activities at a lower level, including diagnostic of system failures to identify failed components and removal/installation of those components. There may be additional risk to the crew during more complex maintenance operations, as well as an increased risk of unsuccessful maintenance. Increased complexity is also likely to increase the amount of crew time required for maintenance, just as reduced complexity resulting from the implementation of higher-level maintenance using ORUs is intended to reduce maintenance crew time. Decisions regarding level of maintenance must carefully balance their impact on logistics mass, crew time, and the challenges of design for lower-level maintenance [1].

1.2.2 Commonality

Commonality is a maintenance strategy in which replaceable items from different parts of the system are designed to be interchangeable, meaning that a single type of spare can cover multiple types of failures. The use of common items reduces the total number of spares that are required, since each additional spare covers multiple sources of risk. In addition, common items can enable scavenging or cannibalization of items from some systems to provide contingency spares for other, higher-criticality systems. Finally, commonality can simplify inventory management and maintenance procedures by reducing the number of different types of items that must be accounted for, as well as the number of different tools that may be required to perform maintenance [1, 48].

Common items do not have to be identical, but they do need to be interchangeable with regard to both form and function. Commonality can impose additional constraints on the design process, forcing an item to conform to requirements associated with multiple locations in the system. As a result, the final design may not be as optimal in terms of local performance as it could be if it were specific to a single installation point. However, the overall system characteristics may be significantly improved as a result of the logistics benefits of commonality. In some cases, the loss of local optimality in item design is more than made up for by the global optimality enabled at the system level [1].

Commonality can increase the risk of common cause failures, since the same design is used multiple times within the system. A common cause failure is a failure that occurs in different parts of a system for the same reason [49]. When the same design is reused, failure modes within that design – whether they arise from design flaws, manufacturing flaws, unanticipated interactions, or any other source – become common failure modes across multiple systems. As a result, failure modes within common items can be more critical than those in individualized items, since they may impact wide swaths of system functionality. This increased criticality should be taken into account during testing and evaluation of proposed common item designs.

However, the use of commonality increases the population of a given type of item that is active in the system at a given time, providing more data for for testing and potentially helping to accelerate reliability growth and uncertainty reduction. As a result, improvements to system reliability and/or failure rate estimates may be achievable faster than they would be for a set of individual items. Therefore, while the impact of a failure mode within a common item may be greater, the ability of system designers to detect and understand/correct that failure mode is likely increased. Commonality also reduces the number of different types of items that must be designed and evaluated, which may simplify development and testing for the system. During system development, designers must carefully balance the impact of additional constraints and common cause risks against the logistics mass reduction and simplification of development, testing, and operations that may result from the implementation of commonality.

1.2.3 Redundancy

Redundancy addresses reliability issues in the same way as spare parts – additional copies of certain system elements are provided so that if one fails, another can take its place. However, with redundancy the replacement item is already within the system, and no crew time is required to execute a maintenance action to install a new spare when failure occurs. This provides an additional benefit in that redundant components can recover system functionality without crew intervention, thus significantly lowering risks associated with loss of critical functions.

Redundancy can be implemented at many levels within the system, from individual components all the way to entire redundant systems. Level of redundancy considerations have similar implications to level of maintenance, described in Section 1.2.1. Specifically, while high-level redundancy may provide the ability to switch over to an entirely different system or subsystem when a failure occurs, it is not a mass-efficient way to improve POS for the system [29]. In addition, redundancy may consist of identical copies of a system, or of different systems that accomplish the same function. The latter approach, known as dissimilar redundancy, has the benefit of reducing the potential for common cause failures.

Analysis shows that multiple spares are typically needed to mitigate risk for most components [22, 29]. The particular number of spares required varies as a function of component and mission characteristics, and the optimal spares allocation for a particular system will almost certainly not simply provide the same number of spares to each item, or to large groups of items. Since each redundant instance of a system or subsystem effectively supplies only one additional spare for all items within that system or subsystem, multiple redundant copies would be necessary to provide complete risk mitigation. Spare parts are typically implemented at a lower level than redundancy, and are therefore typically a more mass-efficient approach for risk mitigation. When combined with spare parts, however, redundancy can provide a backup capability to maintain system function while maintenance activities are completed, giving the crew more time to respond to failures. Thus, while redundancy may be valuable to preserve critical system functions during maintenance downtime, it is unlikely to eliminate the need for spare parts as an effective supportability strategy. Decisions regarding redundancy – including the level at which redundancy is to be implemented, whether it is similar or dissimilar, and how many redundant elements to include – impact system mass, risks associated with loss of function, and crew time required for maintenance [1].

Another potential approach to is to use analytic rather than physical redundancy. That is, a system could be designed to be reconfigurable in the event of a failure, so that other systems could provide some level of functionality after a primary system fails. This approach is distinct from dissimilar redundancy since the secondary systems would be primarily designed for other functions. As a result, it is likely that the result would be a degraded system state where limited resources and systems are rebalanced to prioritize critical functions (or at least partial restoration of critical functions), potentially at the expense of other, less-critical functions. Analytic redundancy could provide options for reducing the consequences of an inability to maintain critical systems, thereby loosening the relationship between POS and mission-level metrics like P(LoC) and P(LoM) and potentially enabling relaxed constraints on POS. Put another way, a system that can reconfigure to adapt to failures can accept a lower POS for a specific system while still providing an acceptable level of risk mitigation with regard to that system's functionality. However, this type of redundancy requires much more detailed examination of the primary and potential secondary functionalities provided by spacecraft systems, or combinations/alterations of those systems, and is therefore considered beyond the scope of this research. Supportability analyses such as the ones described in this dissertation can be used in conjunction with more detailed examination of analytic redundancy options within a particular system in order to assess the potential value of such an approach, in terms of the amount that a POS constraint could be relaxed and the resulting impacts on logistics requirements.

1.2.4 Distributed Functionality

The physical instantiation of spacecraft systems are associated with a particular function or set of functions, and the mass of those systems is often driven by the functions that they provide. For example, ECLSS systems are sized to process life support consumables at a particular rate in order to support the crew. Changes in consumables processing requirements, such as a higher water demand, can significantly impact the mass of ECLSS systems. One potential strategy for reducing logistics mass is to split larger subsystems into multiple, identical modules, each of which handles a portion of the total load. Reducing the load on each specific module results in smaller modules, though splitting a subsystem does increase the number of instances of each of those smaller units in the system. For any ORUs within these distributed subsystems, this strategy effectively replaces a single large ORU with a set of smaller ORUs that are common with each other.

System mass scaling laws, such as those used by the Advanced Life Support Sizing Analysis Tool (ALSSAT) [50], typically specify some portion of the component's mass as being variable as a function of load, while the rest is fixed (this relationship is described in equation 3.2 on page 124). As a result, halving the load on a system does not typically halve the size of that system, and distributing a subsystem across multiple modules will typically increase the emplaced mass of that system. However, the total mass required for the set of spares covering the multiple modules of that distributed system can be less than the total mass that would be required using monolithic spares. From a crew time perspective, however, distributed functionality will increase the number of maintenance events (since more instances of the system are operating), and therefore will increase the amount of time that the crew will have to spend maintaining that particular system. Overall, the potential mass benefits of distributed functionality must be weighed against the increased crew time requirements and potentially increased system complexity.

1.2.5 On-Demand Manufacturing

ODM, also referred to as In-Space Manufacturing (ISM), can enable the on-demand, in situ production of useful items from raw material feedstock and/or low-level components. In general, this approach is called ODM in this dissertation, while specific technology and/or in-space implementations are referred to as ISM. This "just-in-time" manufacturing has the potential to significantly reduce logistics mass requirements by enabling the use of common raw materials to cover a variety of potential failures on-demand, eliminating the need to specialize spares and maintenance items to particular components [10, 24, 25, 51, 52].

ODM is effectively an extension of commonality (Section 1.2.2), and has similar effects in terms of broad risk coverage from common resources. However, a key difference is that where commonality of spare parts imposes interchangeability constraints on the design of the part itself, ODM imposes the need for common materials and for manufacturability within the constraints of ODM technology available to the mission. This commonality of material, rather than commonality of design, can enable the application of the benefits of commonality more flexibly than traditional implementation, especially as manufacturing technologies and capabilities advance [24, 25].

Recent technological advancements in Additive Manufacturing (AM), colloquially known as 3D printing, have opened the door to more advanced manufacturing capabilities in space than have previously been available, due to the fact that AM processes are almost entirely automated (meaning they do not require excessive crew time) and relatively resource-efficient when compared to traditional, subtractive manufacturing techniques. However, it is important to note that ODM does not necessarily have to make use of AM, though there are many benefits to AM which make it an attractive option. An extensive review of potential applications of AM for spaceflight and the benefits that might result was published by the National Research Council (NRC) in 2014 [53]. The first 3D printer in space, the 3D Printing in Zero-G Technology Demonstration Mission, was delivered to the ISS in 2014 and has demonstrated the feasibility of ISM using Fused Deposition Modeling (FDM) in a microgravity environment; initial results of this technology demonstration are reported by Prater et al [54].

In addition to enabling commonality of material and the resulting savings in logistics mass, ODM can enable adaptable maintenance logistics. Traditional spare parts are specialized, covering specific failure modes (or sets of failure modes, for common components). In contrast, raw materials and an ODM capability can be used to cover a wide variety of potential failure modes, and specific allocations of resources to specific failure modes do not have to be made before launch. When traditional, individualized spares are used, an unexpectedly high failure rate in a single component could exhaust the spares allocated to that item before the end of the mission (potentially resulting in LoC, LoM, or LoV), even if other items experience unexpectedly low failure rates and have a surplus of components. ODM enables the adaptive reallocation of maintenance resources, since raw materials are not specialized. If one item requires fewer spares than are expected, resources that might have been used for that item can be reallocated to cover demand from an item that requires more spares than expected. As a result, ODM helps mitigate the impact of epistemic uncertainty [25]. In addition, ODM provides the flexibility for the crew to manufacture items that may not have been planned for, providing a powerful contingency option [25, 55–57]. ODM can also offer several other benefits, including simplified inventory management, reduced logistics stowage volume requirements, and opportunities to re-optimize system designs without launch packaging or loading constraints [24, 25, 51, 53, 55, 56].

One of the most significant benefits of ISM may be that it enables recycling and/or In-Situ Resource Utilization (ISRU) for maintenance logistics. Recycling has been used to great effect for the reduction of ECLSS consumables requirements on past spaceflight missions, and ISRU has the potential to reduce logistics demands even further for crewed missions to the Moon and Mars [42, 58–61]. When ODM is available, recycling and ISRU can be applied to spare parts in the same way that it is currently applied to water, oxygen, and propellant. Failed components may be converted into raw materials that could be used to manufacture new components, and new components may be manufactured on site using local materials. Significant technology development is required before this approach can be fully implemented. However, such a capability would be a revolutionary for maintenance logistics, drastically reducing the amount of mass required for long-endurance missions [25, 47, 51–53, 55, 62–64].

ISM is still a relatively new capability, and a significant amount of development effort remains before ODM techniques can be deployed aboard space systems in a mission-critical setting. This development and testing cost must be accounted for in future mission planning when ODM is considered. In addition, ODM introduces new risks that must be carefully examined. For example, if ODM is to be used to manufacture critical components, then the ODM system may become a common failure mode across several systems. Thus, the reliability and maintainability of such a system is a critical consideration. In addition, additional mass and volume required to enable ODM must be accounted for, including the manufacturing system itself as well as any logistics that it may require for operations.

The manufacturing capability of any ISM system is a critical concern. This includes available materials, precision, surface finish, quality, and reliability of the manufacturing process. ISM capabilities place constraints on the type and quality of part that can be manufactured in space. However, it is important to bear in mind that the operational and logistical context of parts manufactured in space is different from that of parts manufactured on Earth and launched into space, and therefore the optimal design may change – especially if material recycling and/or ISRU are available. Therefore, component designs may change, adapting to the constraints of an ISM system or to the different context provided when material can be recycled or produced in situ.

Decisions regarding the implementation of ODM – including selection of specific components to produce in space, which manufacturing capability to develop, whether or not to use recycling/ISRU, and what changes to component and system design may be implemented to make full use of ISM – must carefully balance the cost of ISM/recycling/ISRU system development, the cost of any design changes to components to enable ISM, as well as additional logistics and risk against the potential logistics savings or risk mitigation provided by an ISM capability. A more in-depth discussion of ISM and its potential benefits for human exploration missions is presented by Owens and de Weck.[25]

1.3 Past Experience and Future Challenges

In order to understand the supportability challenges of future missions, it is useful to put them in the context of past human spaceflight experience. Figures 1.1 and 1.2 show the mission endurances of eight major human spaceflight programs throughout history, including Mercury, Gemini, Apollo, Skylab, the Space Shuttle, Salyut, Mir, and the ISS, compiled from various sources [65–71]. Mission endurance is defined as the time that the system supports a crew independently (i.e. without resupply), based on the definition provided by Do [42]. The distinction between mission endurance and mission duration is critical for supportability analysis, because it is endurance, not duration, that drives risk. Duration is a more commonly-used metric, and it typically refers to the total amount of time that a system or crew spends in space. For example, the ISS – the current state-of-the-art in long-duration human spaceflight systems – has supported a continuous human presence in LEO since November 2, 2000 [72]. This represents a mission duration of over 18 years. However, the ISS is resupplied from the ground on a regular basis, and as a result its mission endurance is much shorter, on the order of a few months.

The distinction matters because endurance dictates the time horizon over which logistics planning must be conducted. Supportability demands are driven by stochastic processes, and longer planning time horizons result in greater uncertainty. This uncertainty increases the amount of resources that must be allocated to mitigate risk. While long-duration, short-endurance systems like the ISS can be resupplied from the ground to replenish stores of spare parts as they are used – thus reducing uncertainty around logistics requirements, since demand can be filled and refilled after it is needed, so only a relatively small buffer of spares is required – long-endurance systems must carry all required spares for the entire system. While the total number of failures that will actually be experienced may be low, uncertainty regarding which specific items will fail, and how many times, means that a significant amount of mass is still required to reduce risk. For example, it is estimated that more than 95% of spares allocated to cover random failures on the ISS will not be used. However, a large number of spares (and consequently large amount of mass) must be carried for risk mitigation since there is no way to know ahead of time which specific spares will be needed [1].

Endurance is also a driving factor when considering the criticality of different failure modes. In LEO, the availability of rapid abort and resupply options means that the crew are well-protected in the event of an emergency. While a failure event that forces an abort may result in LoM, only a subset of potential failures – specifically, those with a short time to critical impact after failure, such as a loss of pressure event – are likely to result in LoC in LEO. On long-endurance missions with limited abort capability, a much larger set of potential failures directly threaten crew survival, due to the lack of this contingency option. Under these circumstances, the ability of the crew to maintain spacecraft systems, and therefore the supportability of the system, will be much more critical [1, 4].

For sortie missions, such as the Mercury, Gemini, and Apollo missions, mission endurance is equal to the time between launch and landing. For space stations (i.e. Skylab, Salyut, Mir, and ISS), mission endurance is equal to the time between docking events while the station is crewed. 525 distinct mission endurances are presented here, ranging from less than a day to nearly 120 days. The missions included in this dataset range from the flight of Alan Shepard aboard *Freedom 7* (Mercury-Redstone 3) on May 5, 1961 to the arrival of the Cygnus CRS-10 ISS resupply mission on November 19, 2018. Figure 1.1 shows these endurances in a timeline over the past six decades, while Figure 1.2 organizes these endurance periods by program.

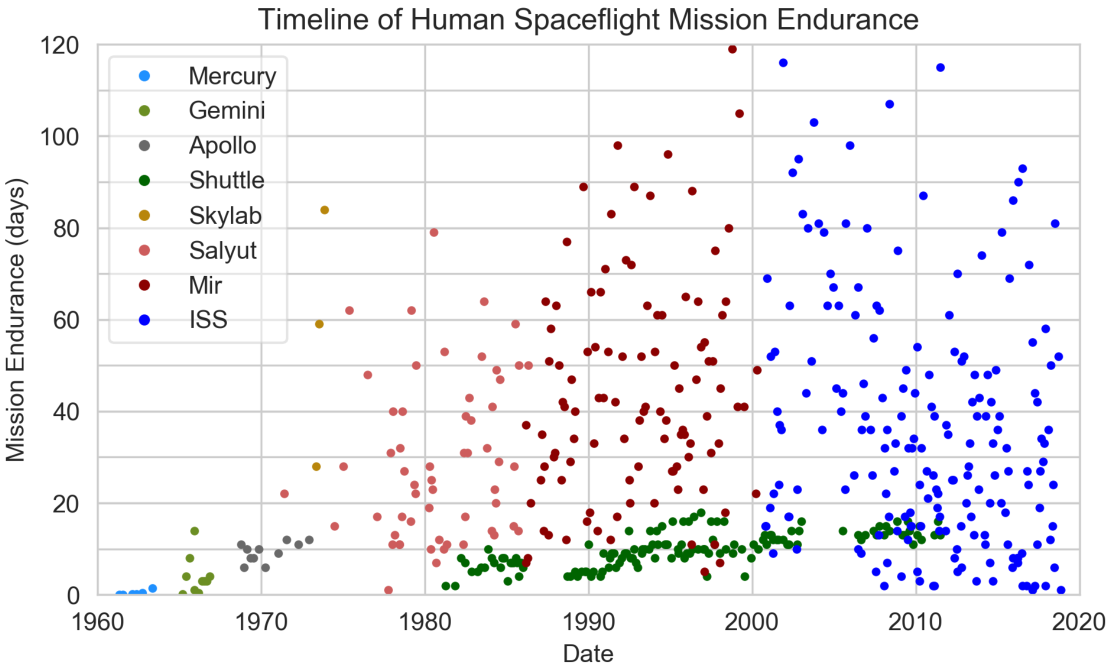

Figure 1.1: Timeline of human spaceflight mission endurance, color-coded by program [65–71]. Each point represents the start date of a period of time that a crewed spacecraft operated without receiving supplies from Earth, plotted at a height equal to the mission endurance. For sortie missions, mission endurance is equal to the time between launch and landing; for space stations, each endurance period is defined as the time between docking events.

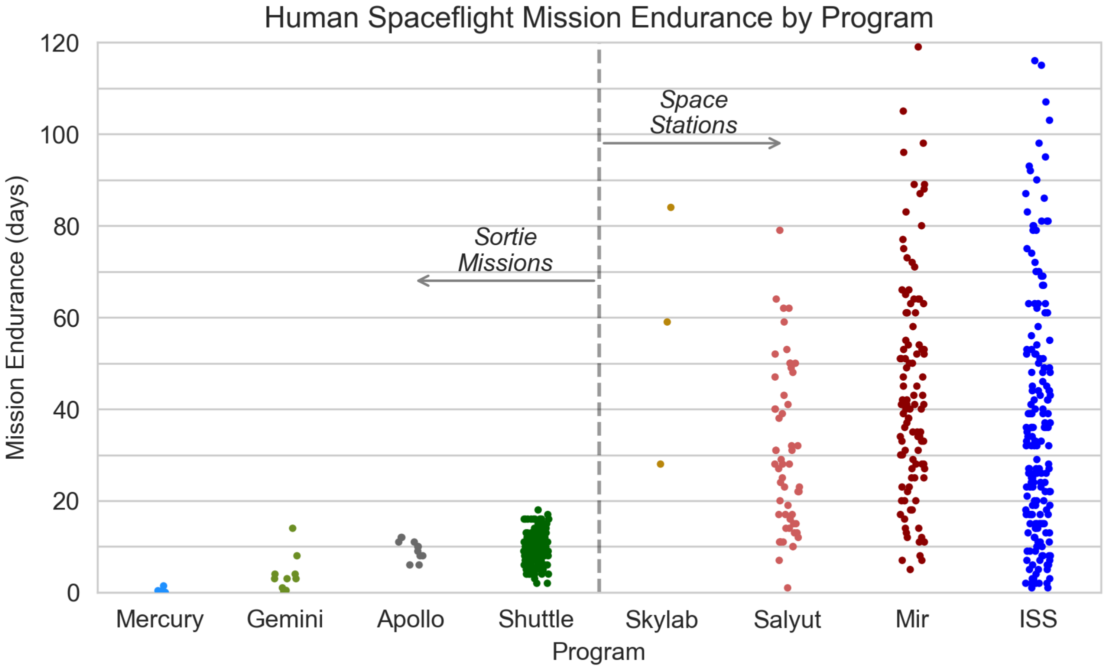

Figure 1.2: Human spaceflight mission endurance by program [65–71]. Each point represents an endurance period within the specified program (a random amount of lateral jitter is added to reduce overlap between points). Sortie missions are shown on the left, while space stations are shown on the right. For sortie missions, mission endurance is equal to the time between launch and landing; for space stations, each endurance period is defined as the time between docking events.

Table 1.1: Summary of historical crewed mission endurances, showing the average and maximum for each program. For space stations, this corresponds to the time between visiting vehicle arrivals. Values are rounded to the nearest day (except for the average mission endurance for the Mercury program, which was less than 1). The time period during which the program's crewed flights occurred is also indicated.

Program	Period of Crewed Flight Activity	Endurance (days)	
		Average	Maximum
Mercury	1961-1963	<1	1
Gemini	1965-1966	4	14
Apollo	1968-1972	9	12
Shuttle	1981-2011	10	18
Skylab	1973-1974	57	84
Salyut	1971-1986	30	79
Mir	1986-2000	44	119
ISS	2000-Present	36	116

The record for the longest period of time that a crewed spacecraft has gone without resupply is held by Mir, which had a period of independent operations lasting 119 days from 1998 to 1999. Between the arrival of Progress-M 40 on October 26, 1998, and Soyuz-TM 29 on February 22, 1999,[c] Cosmonauts Gennadi Padalka and Sergei Avdeyev lived aboard Mir for nearly four months without resupply from Earth [70].[d] The next two longest mission endurance records are held by the ISS at 116 days (November 28, 2001 to March 24, 2002) and 115 days (July 10, 2011 to November 2, 2011), respectively [71]. The longest mission beyond LEO, Apollo 17, lasted for over 12 days, from December 7-19, 1972 [67]. No crewed spacecraft have traveled beyond LEO since that time. The maximum and average mission endurance for each program endurance are reported in Table 1.1.

For space stations, the mission endurances shown in Figures 1.1 and 1.2 and summarized in Table 1.1 represent periods of time between docking events during which the station was occupied. Some of these docking events were uncrewed cargo delivery vehicles, while others (specifically, Shuttle and Soyuz) delivered both crew and cargo. While the Space Shuttle was a large vehicle capable of delivering a significant

[c] Dates associated with Russian space programs are given in Moscow Time.

[d] Cosmonaut Yuri Baturin arrived at Mir with Padalka and Avdeyev, but returned to Earth after approximately 12 days [70].

amount of cargo and logistics to the ISS at the same time as crew, the Soyuz cargo capacity is much more limited. Data are not available showing the amounts of logistics delivered on each of these flights; however, it may be that Soyuz docking events do not “reset” the endurance clock, since they primarily deliver crew to the station rather than cargo. When Soyuz dockings are removed from the dataset, the longest period of time between docking events increases to 160 days, which occurred on Mir between Progress-M 40 and Progress-M 41 (October 26, 1998 to May 4, 1999) [70]. The longest period of time that the ISS has gone between resupply vehicles, excluding Soyuz, was 154 days between Progress 12 and Progress 13 (August 30, 2003 to May 27, 2004) [71].

The purpose of these data is not to suggest that past and current human spaceflight systems *could not* operate for longer periods of time in LEO without requiring resupply, but simply that they *did not*. These programs operated under different constraints and in a different context than future exploration missions, and therefore it is expected that they would be optimized to that environment. In LEO, resupply from the ground is readily available at all times, and it makes sense to structure an operational plan around that capability. However, these data provide necessary context for examining the challenges of future human spaceflight and how the mission endurances required to go to Mars differ from past experience.

Future human spaceflight missions beyond LEO will be unlike any previous human spaceflight experience. Though it is not always the immediate next step in human spaceflight programs, the horizon goal of human spaceflight is to land humans on Mars and return them safely to the Earth [73]. On missions such as this, crews will fly farther from home than ever before, and remain in space for longer. Resupply and abort options will be heavily constrained, if they are available at all. Without resupply options, mission endurance will be significantly increased from that of past missions. Figure 1.3 shows the same data as Figure 1.2, with a column added to show the 1,200-day Mars mission endurance in comparison to past experience [11, 74]. The endurance required for a Mars mission dwarfs all previous human spaceflight experience. In fact, the total time in orbit accumulated by the Space Shuttle fleet –

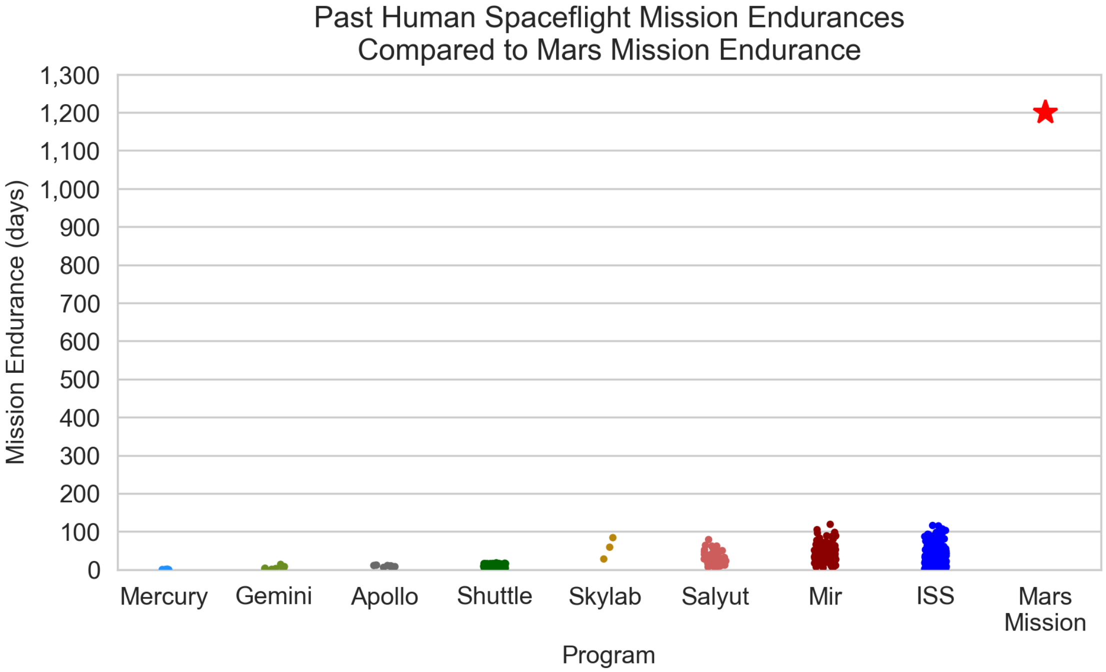

Figure 1.3: Past human spaceflight mission endurances compared to the endurance required for a Mars mission.

1,323 days over 135 missions [75] – is just over 10% longer than the time that a system must support the crew independent of Earth for a single Mars mission. Overall, the supportability strategies used by current and previous systems all take advantage of short mission endurances, readily-available abort options, and/or regular resupply from Earth to manage risk to acceptable levels without requiring excessive spares mass. These options will not be available on future missions.

The longest time period that a crew has operated in logistical isolation in space is 119 days; the longest period of time that a crew has spent beyond LEO is 12 days. In contrast, a mission to Mars will require a system that can support the crew in space – with no timely abort or resupply capability – for 1,000 to 1,200 days [11, 74]. That is, the endurance required for a mission to Mars is an order of magnitude greater than the longest mission endurance in past experience, an two orders of magnitude greater than the longest beyond-LEO mission in human spaceflight history. These values are plotted in Figure 1.4. In addition, when abort options are limited and the

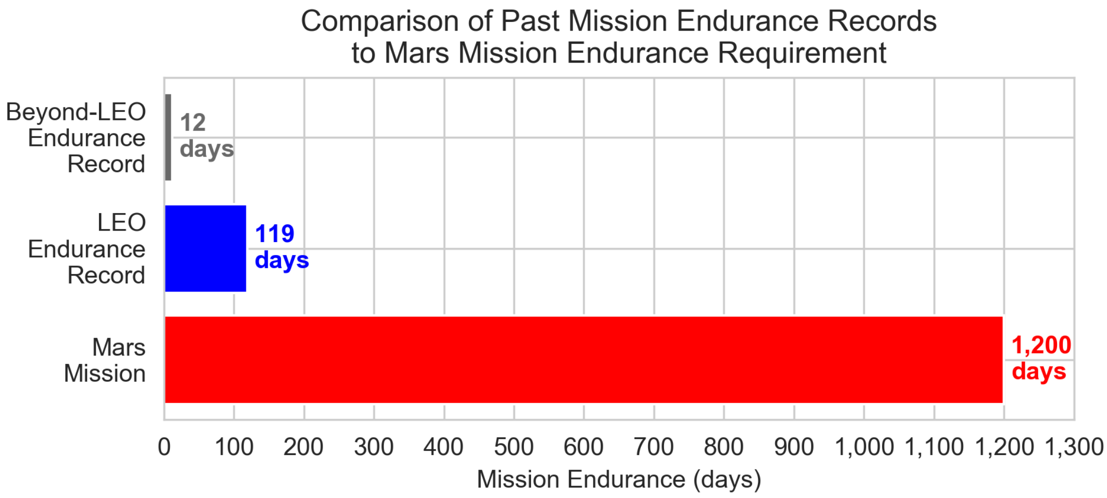

Figure 1.4: Comparison of historical mission endurances to Mars mission endurance requirements.

crew does not have a way to quickly return to Earth in the event of an emergency, the criticality of various system failure modes increases. The combination of these factors – long endurance missions without timely options for abort or resupply – creates an unprecedented supportability challenge for future missions. Strategies that have supported human spaceflight operations in LEO and on short trips to the Moon in the past will not be effective for future missions to Mars or other beyond-LEO destinations. New approaches to supportability will need to be developed [1, 3–9].

1.4 Optimization

A supportability strategy consists of a variety of elements, described in Section 1.2. These decisions are complex, often coupled, and have conflicting influences on a large, multiobjective tradespace. The systems required to enable the human exploration missions of the future will push the limits of engineering capability, and require efficient solutions that minimize mass and maintenance crew time. The large, complex tradespace defined by supportability strategy decisions, as well as need for high-quality solutions, motivates the need for systematic optimization, rather than manual exploration of options. In addition, the significantly different operating context of future missions means that heuristics and expert opinion based on the past several decades of experience in LEO may not be easily extensible to this new challenge. While modeling and optimization alone cannot solve these supportability challenges, they can provide key insights, support engineering creativity, and help guide programmatic decision-making by allowing analysts to explore various options and generate quantitative assessments of the potential costs and benefits associated with each one.

During the search for optimal supportability strategies, it is important to consider as many design variables concurrently as possible. It is unlikely that a true optimum will be found by optimizing each variable alone and then combining the results. Artificially constraining the solution space by arbitrarily fixing a subset of design variables, rather than leaving them free to be optimized, may result in a solution of lower quality than one that could have been found if the full solution space were available. For example, Agte [76] demonstrated that optimization of Unmanned Aerial Vehicle (UAV) design that considered static design variables and component failure rates concurrently found solutions with higher system availability and/or lower flyaway cost than solutions found when either type of variables were considered in isolation.

This effect is formalized in the concept of a relaxation, or the Lower Bounding Principle (LBP) of optimization: given two sets $A,\ B \in \mathbb{R}^n$ (where $A \subseteq B$), a set of decision variables $x \in B$, and an objective function $f(x)$ to be minimized, the minimum objective value achieved when x is only restricted to B is guaranteed to be less than or equal to the minimum objective value found when x is restricted to A:

$$\min\{f(x) \colon x \in B\} \leq \min\{f(x) \colon x \in A\} \qquad A \subseteq B \tag{1.2}$$

Put another way, an optimal solution found when searching the larger space (B) is guaranteed to be at least as good as one found when searching a subset of the space (A), and may in fact be better [77, 78].

Therefore, it is important that the search for optimal supportability strategies examine a wide range of options concurrently in order to find high-quality solutions. However, for tractability it is also important to keep the size of the design space (or of the search space) at a scale that can be feasibly explored in a reasonable amount of time. The combination of various supportability strategy decisions defined in Section 1.2 can quickly result in an enormous tradespace when large systems are examined. Well-defined constraints (eliminating known infeasible solutions) and efficient optimization algorithms (quickly identifying potentially optimal vs. known non-optimal regions of the tradespace) are essential for the solution of large-scale problems. Rapid evaluation of candidate solutions, and/or bounds on objective values for those solutions, is also critical.

The balance between the fidelity and scope of the evaluation method and tradespace should be carefully considered in the context of the goals of a particular analysis. During early development activities, where the tradespace is large and parameter uncertainty is high, more emphasis may be placed on rapid, coarse evaluation and broad tradespace search. As the system matures, emphasis will likely shift to higher-fidelity models exploring more specific tradespaces, which may themselves be smaller, or may simply be defined at a higher fidelity. Methodologies that provide the ability to rapidly identify Pareto-optimal solutions in large tradespaces (hundreds of thousands of candidate solutions, or more) with mid-fidelity models will provide a powerful analysis capability to inform system development.

1.5 Summary and Analysis Needs

Supportability will be a greater challenge for future human spaceflight missions than ever before, and will therefore be a larger driver of system and mission design. Crews and their systems will be logistically isolated from Earth for an order of magnitude longer than any previous mission, and will not have the ability to abort and return to Earth quickly in the event of an emergency. Epistemic uncertainty in failure rates will be a more significant source of risk due to this extended mission endurance.

This challenge is unprecedented in human spaceflight, and critical maintenance resources such as spares mass and crew time will have to be carefully managed in order to mitigate risk to acceptable levels without incurring excessively or even prohibitively high cost or negatively impacting mission productivity. A wide variety of supportability strategy decisions will affect the characteristics of these future missions, and the strategies that have been used for past missions will not be feasible for missions beyond LEO. In addition, the experience gained via LEO operations may not be directly applicable to future missions, due to significant changes in mission constraints and operating contexts. New strategies must be developed and implemented, and they must be carefully selected to minimize the amount of mass and maintenance crew time required to mitigate risk.

Therefore, there is a need for quantitative, multiobjective modeling and optimization of spacecraft supportability strategies in order to support future system development. This modeling capability should be able to assess the mass and crew time required to achieve a desired level of risk as a function of system and mission characteristics and the selected supportability strategy, and provide information regarding the tradeoffs between these metrics and the decisions that drive them. It must also have sufficient flexibility to assess a wide range of supportability strategies concurrently. The capability to examine strategies enabled by new technology developments, specifically ODM, will be particularly important for informing investment and system design decisions related to those options. Both aleatory and epistemic uncertainty must be considered, since both can be significant drivers of risk. Finally, since the supportability strategy tradespace is large and involves various conflicting and coupled decisions, these models and related optimization approaches must be able to systematically and rapidly evaluate potential options in order to identify Pareto-optimal supportability strategies that minimize mass and crew time for a given risk threshold.

1.6 Dissertation Outline and Chapter Summaries

To address the challenges laid out in Section 1.5 and the research gap identified in Chapter 2, this dissertation presents a novel methodology for multiobjective supportability strategy optimization. A multiobjective COP is defined with decision variables encoding a range of supportability strategy options to enable formal optimization. A set of methods for interpreting full or partial assignments to those decision variables to assess mass and expected maintenance crew time a given system and mission is also described. The underlying supportability evaluation model incorporates aleatory and epistemic uncertainty into risk evaluation, and provides several novel supportability modeling capabilities, including

- Improved maintenance crew time estimates via separation of the stochastic processes representing demand for spares and crew time;
- Improved risk assessment via inclusion of the probability of sufficient crew time for maintenance;
- The ability to evaluate the relationship between ODM feedstock mass supply levels and risk, which enables consideration of ODM as a supportability strategy alongside other options; and
- Reduced evaluation time for demand distributions via use of numerical characteristic function inversion techniques instead of Monte Carlo simulation.

The following sections provide brief summaries of each chapter of this dissertation.

Chapter 1: Introduction

This chapter presented the motivation for this research, as well as key background information to provide context. Supportability was defined and discussed in terms of its importance during early system design, relationship to key system metrics, and relationship to uncertainty, with particular attention paid to the importance of epistemic uncertainty. The set of decisions that make up a supportability strategy were also described and discussed, along with their impacts on mass, crew time, risk, and other metrics. Historical human spaceflight mission endurances are used to provide context and highlight the differences between past experience and future mission requirements. Finally, the need for systematic optimization is discussed and the overall challenge is summarized in order to highlight analysis needs.

Chapter 2: Literature Review

Chapter 2 reviews the relevant literature in the context of the need identified in this chapter. A variety of supportability models are examined and discussed in terms of their ability to assess mass, crew time, and risk, to include epistemic uncertainty, and to optimize over a range of supportability strategies. This review finds that a the majority of models are designed to assess individual supportability strategies rather than optimize the underlying decisions. When supportability strategy optimization is provided, it is limited and focused on one particular element of strategy at a time. As a result, the primary gap identified for this research is the lack of supportability strategy optimization capability. In addition to this optimization gap, a number of gaps are identified related to modeling capability. Specifically, no models for the relationship between POS ODM feedstock levels exist in the literature other than previous work related to this dissertation. Existing maintenance crew time modeling techniques use failure rates to assess crew time demands, and as a result do not capture crew time associated with non-R&R events. In addition, the probability of sufficient crew time is not included in the evaluation of overall POS or in spares allocation optimization to achieve a given POS requirement. Finally, all models that do evaluate the distribution of crew time requirements do so using Monte Carlo simulation, which is computationally expensive and limits the extent to which those techniques can be applied to large tradespaces. These research gaps, along with the challenges described in this chapter, are used to define the research question for this work:

> *What is the Pareto-optimal set of supportability strategies that minimize mass and crew time required for maintenance for future long-endurance crewed space missions, given a particular*

system, mission, and risk requirement?

Chapters 3 and 4: Methodology

Chapters 3 and 4 present the supportability optimization methodology developed by this research, which answers the research question and fills the identified research gaps. Chapter 3 formally defines the supportability strategy optimization problem as a multiobjective COP, providing a detailed mathematical definition of inputs, decision variables, objectives, and constraints. The set of inputs describe a particular system and mission, and the decision variables are designed to encode a range of supportability strategy decisions – level of maintenance, ODM, commonality, redundancy, and distributed functionality – and provide a formal representation that enables automated, systematic strategy generation, tradespace exploration, and optimization. 21 sets of constraints define feasible assignments to those decision variables and acceptable objective values, reducing the size of the tradespace to improve tractability. The supportability strategy optimization problem is presented in equation 3.36 on page 148.

Chapter 4 presents the methodology used to solve the supportability strategy optimization problem. This chapter defines the process for defining the ORU list, given a full assignment to the decision variables (i.e. a candidate supportability strategy). The ORU list enumerates and characterizes all ORUs that are considered for maintenance, and fully defines the supportability characteristics of the system. A supportability evaluation model is presented that determine the mass and maintenance crew time requirements of a given ORU list. This model, which consists of separate crew time and mass models, incorporates most capabilities of existing supportability analysis models (e.g. spares allocation optimization, expected crew time assessment) alongside the novel analysis capabilities listed at the beginning of this section. These novel capabilities provide the capability to asses ODM, as well as improved crew time and risk assessment. This chapter also describes the multiobjective branch and bound optimization strategy implemented to solve the COP defined in Chapter 3, including a process for initializing a candidate Pareto frontier, methods to calculate lower bounds on mass and crew time given a partial assignment to the decision variables, and variable/domain ordering heuristics used during optimization. The supportability strategy optimization problem formulation described in Chapter 3 and the evaluation methodology described in Chapter 4 are implemented in an analysis tool called MCROSS. The key novel contributions of this methodology are summarized in Section 4.4.1 on page 4.4.1.

Chapter 5: Verification and Validation

Chapter 5 presents an in-depth verification and validation of the supportability model developed in the previous two chapters. Model verification is performed via comparison to large-scale Monte Carlo simulation. Specific aspects of the model that are investigated include the negative binomial approximation used for maintenance demands, the process for combining sets of components to define ORUs, the approximation used to represent the impact of redundancy, and the numerical characteristic function inversion technique used to assess ODM feedstock and crew time distributions. Each of these assessments demonstrates that the model produces results that are in close agreement with Monte Carlo simulation in the region of interest for most supportability analyses (i.e. POS values greater than 0.9). The largest source of error is the redundancy approximation, but this error is most pronounced in the low POS range, which is typically not relevant to mission analysis. Future work will seek to improve this approximation.

The second portion of Chapter 5 uses ISS flight maintenance data and an existing state-of-the-art supportability model to validate the corrective maintenance-related outputs of MCROSS. An assessment of the validity of specific ORU failure rates, crew action rates, and maintenance crew times is presented, since all of these are key inputs to the model. This input validation – which is a domain contribution of this dissertation, since there is no known in-depth, ORU-by-ORU examination of parameter validity – shows that, while many estimates are accurate, there are several cases of underestimated failure rates and maintenance crew times within the dataset that can have a significant impact on model outputs. The corrective maintenance crew time model presented in Chapter 4 is validated via comparison to the methodology used by Exploration Maintainability Analysis Tool (EMAT) as well as observed ISS maintenance crew time requirements. EMAT is one of the most capable supportability models found during the literature review, and is regularly used to inform NASA technology investment, system development, and mission planning decisions. This comparison shows that the MCROSS crew time analysis methodology – which separates crew time demand and spares demand into two different stochastic processes – provides crew time estimates that are closer to observed values than current techniques. Comparison of spares mass estimates generated by MCROSS and EMAT show close agreement. In addition, statistical analysis of spares allocations generated by MCROSS show that, given valid ORU characteristics, the POS estimates provided by the model are in statistical agreement with actual spares demand observed on the ISS. Overall, the verification and validation performed in Chapter 5 demonstrate that the methodology presented in this research is sound, given valid data.

Chapter 6: Notional Case Study

Chapter 6 demonstrates the application of MCROSS to a notional system, and validates the optimization process by comparing the Pareto frontier identified by MCROSS to one identified via full enumeration of the tradespace. This comparison shows that the optimizer is identifying all Pareto-optimal strategies correctly, and in conjunction with the model validation presented in Chapter 5 validates the overall methodology. A comparison of optimization and enumeration shows that optimization reduced the number of evaluations to 15% of the total tradespace, and reduced the amount of time required for analysis by over 86%, indicating that the optimization techniques used here can significantly improve the efficiency of supportability analyses. A series of sensitivity analyses examining changes in POS and endurance are conducted to demonstrate application of the methodology in a notional iterative design process.

Chapter 7: ISS Oxygen Generation Assembly Case Study

Chapter 7 demonstrates the application MCROSS to examine a real-world system, namely the ISS OGA, on a notional 1,200-day Mars mission. The tradespace associated with this system is much larger than the one examined in Chapter 6, and this case study illustrates the value of formal supportability strategy optimization. The Pareto frontier is identified and characterized in comparison to the mass and crew time that would be required if the current ISS strategy were used. The current strategy does not appear on the Pareto frontier. This analysis demonstrates that MCROSS can identify supportability strategies that provide reduced mass and crew time requirements over current practices, though it is important to note that these results only capture supportability-related considerations and there are likely other factors that influence design. A series of sensitivity analyses are again performed, examining changes in POS, endurance, maintainability, and ODM system mass. These sensitivities are structured to represent the application of MCROSS in an iterative design process, showing how the outputs of the methodology developed in this research can be used to inform technology investment, system design, and mission planning. Overall, MCROSS provides the ability to examine hundreds of thousands of supportability strategies in order to identify and characterize optimal solutions across a range of analysis cases in a matter of hours. A comparable analysis using previous methodologies would require extensive effort to define analysis cases, and would likely be prohibitively time-consuming without a formal supportability strategy encoding scheme such as the one described in Chapter 3.

Chapter 8: Discussion

Chapter 8 presents a discussion of this research. The methodological and domain contributions are presented, along with their impacts and value to supportability analysis. Key assumptions and limitations are also discussed, including factors that are beyond the scope of the current model and limitations resulting from data availability. Finally, a brief overview of applications of this methodology to date is presented, illustrating the various ways that this approach has already been used to inform technology investment, system design, and program planning.

Chapter 9: Conclusions

Chapter 9 presents the conclusions of this research. Key contributions are enumerated and summarized, and a range of areas for future work are identified. This includes extensions and improvements to the model itself, as well as development of interfaces to facilitate use of MCROSS in conjunction with other models examining other elements of space mission design, such as transportation or habitation system analysis. Overall, supportability analysis and optimization is a critical capability to enable future human space exploration, and the novel multiobjective optimization methodology presented in this dissertation provides a key capability to help inform system and mission development.

THIS PAGE INTENTIONALLY LEFT BLANK

Chapter 2

Literature Review

This chapter reviews and discusses relevant literature related to spaceflight supportability modeling and optimization, including assessment of risk, logistics mass, and crew time. As discussed in Section 1.5, the challenges of future crewed exploration missions motivate the need for a supportability analysis and optimization capability that can

- Quantitatively assess the maintenance logistics mass and crew time required to achieve a desired risk objective for a given system and mission, as a function of the selected supportability strategy;

- Provide sufficient analysis flexibility to enable the concurrent examination of a broad range of potential supportability strategies, including strategies such as ODM that are enabled by new technologies (e.g. ISM);

- Account for both aleatory and epistemic uncertainty when assessing risk; and

- Enable systematic, multiobjective supportability strategy optimization and tradespace exploration, identifying Pareto-optimal strategies that minimize mass and maintenance crew time requirements for a given risk threshold.

Each supportability analysis technique identified in the literature is briefly described and assessed in the context of these analysis needs.

The references reviewed in this chapter are organized into broad categories based upon their relevance to this research. Section 2.1 describes models that include an assessment of spares mass, and Section 2.2 describes models that assess the crew time required for maintenance. Note that some references are discussed in both sections; when this is the case, the model in question is primarily described in Section 2.1 (since for the most part these models are primarily designed to evaluate spares mass) and the elements of the model related to crew time evaluation are described in Section 2.2. Section 2.3 describes modeling and optimization of reliability- and supportability-related system design characteristics and maintenance strategies. Section 2.4 discusses ODM modeling. Note that there is no section which purely examines risk, since risk is an integral part of supportability mass and crew time analysis; instead, risk (usually captured in terms of system availability or POS) is discussed in the context of its impact on other metrics where appropriate. Finally, Section 2.5 identifies and characterizes the gap in the literature, and Section 2.6 presents the research question, based on this gap, that is addressed by this dissertation.

2.1 Mass Models

2.1.1 Heuristics

One of the most common supportability models is also the simplest, which estimates spares requirements as a fixed percentage of system dry mass per year of operations. This heuristic model is based on an analysis of historical spares requirements data for Skylab, Salyut, Mir, and the Space Shuttle, as well as a projection of ISS maintenance requirements under assumed failure rates, conducted by Leath and Green in 1993 [28]. Under this model, the spares mass required per year of operations is estimated (conservatively) to be equal to 5% of the dry mass of the system that spares are being provided for [79].

The dry mass percentage heuristic is particularly useful during the early phase of concept development, when there may not be much more information available about a system other than an initial dry mass estimate. The ability to include a rough estimate of spares requirements during early analyses is valuable, but this approach has significant limitations when it comes to informing system development.

The primary issue with this heuristic is that the spares mass estimate is based only on mission duration and system dry mass. Supportability strategy decisions have no impact, and therefore this approach cannot be used in supportability optimization or tradespace exploration. In addition, there is no assessment of the level of risk associated with the calculated spares mass, nor is there an assessment of crew time. Therefore this heuristic also cannot be used to examine trades between these factors.

More importantly, however, the 5% heuristic is an empirical estimate based on data from LEO space stations. These stations operated in a context that is very different from that of future exploration missions, as discussed in Section 1.3. As a result, it is likely that this heuristic is neither applicable nor accurate for missions with longer endurance and fewer opportunities for resupply and/or abort.

2.1.2 METRIC and Related Inventory Models

Sherbrooke [80] developed the Multi-Echelon Technique for Recoverable Item Control (METRIC) at the RAND corporation in 1966 in order to inform Air Force maintenance inventory decisions. METRIC is a multi-item, multi-echelon model, meaning that it examines inventory for multiple different types of spares at multiple locations within a supply chain, such as operating bases and supply depots. The challenge of inventory management is to determine the number of spares to provide at each supply location (base, depot, etc.) in order to support the system to the greatest degree possible within a given cost constraint, where each LRU or SRU has a specific (fixed) cost associated with it. The core methodology of METRIC formed the basis for several maintenance logistics models for Air Force and NASA applications, described below. The overall philosophy and methodology underlying each of these models is described in detail by Sherbrooke [12]. Though there are variations in capabilities between the various models that have grown out of the METRIC methodology, the core methodology is nearly identical in all variants, and therefore these models are described together.

METRIC Variants

Muckstadt [81] expanded upon METRIC in 1973 by introducing MOD-METRIC, which added the capability to examine multi-indenture problems – that is, inventory problems where an item can be maintained at multiple levels, which may depend upon the location at which maintenance is executed. An "indenture structure" refers to a hierarchical description of the parts of a system at the different levels where maintenance could occur, such as component, sub-assembly, assembly, subsystem, etc. Within the MOD-METRIC framework, a particular maintenance location within the multi-echelon model is associated with a particular level of within the hierarchy of the system. For example, the Air Force refers to items that can be removed and replaced on aircraft at an operating base as LRUs, while the items that make up an LRU and can be replaced in a depot or maintenance shop are known as SRUs [12, 81]. (ORUs are analogous to LRUs for spacecraft.) However, the MOD-METRIC approach restricts the model for maintenance demands (i.e. item failures) to Poisson processes, where previously METRIC had facilitated the use of a range of distributions, including the negative binomial, to describe the demand process [12, 80]. The multi-indenture problem examined by MOD-METRIC is similar to, but distinct from, the more general level of maintenance problem described in Section 1.2.1, as discussed at the end of this section.

Slay [82][a] and Graves [85] further expanded upon the METRIC framework to improve accuracy by including both the expected value and variance of demand rates, modeling demands using a negative binomial distribution. This model, called VARI-METRIC, enabled examination of multi-item, multi-echelon, multi-indenture problems without the restriction to Poisson-process demand models [12, 84]. Slay et al. [83] implemented METRIC and VARI-METRIC approaches in the Aircraft Sustain-

[a] The original Slay report [82] is an internal LMI working paper, the text of which could not be found during this literature review. However, the approach is described in depth by Slay and his co-authors in another report [83], as well as by other authors in other sources [12, 84, 85]. This literature review is therefore based upon those descriptions, which provide sufficient detail to examine Slay's method in the context of this research.

ability Model (ASM), which was developed at the Logistics Management Institute (LMI) in the 1990s to inform Air Force maintenance inventory management. ASM enables evaluation of multi-item, multi-echelon, multi-indenture inventory optimization problems, including the impacts of commonality, variations in item criticality, and cannibalization (that is, the removal of parts from an operational system in order to provide spares for a different system). These last three factors – commonality, criticality, and cannibalization – are options for inputs to the model that impact the results of inventory optimization, but they are not themselves optimized.

In addition, the VARI-METRIC approach used in ASM enables the consideration of epistemic uncertainty in failure rate values. Specifically, the distribution resulting from a Poisson process with an uncertain, gamma-distributed parameter is negative binomial; therefore, the distribution of the number of failures of a particular item with an uncertain, gamma-distributed failure rate can be calculated using the negative binomial distribution [83, 86, 87]. While Sherbrooke [80] had discussed use of the negative binomial distribution in his original presentation of METRIC, that approach was based primarily upon the idea of using a two-parameter distribution (i.e. the negative binomial) to fit the expected value and variance of an observed number of failures in order to develop the demand model; this produces a more accurate model than the pure Poisson model, which only considers the expected value [80, 80, 84]. The VARI-METRIC approach, on the other hand, explicitly applies the negative binomial as a model for random failures with an uncertain rate [82, 83].

Kline and Sherbrooke [88, 89] developed the Multiple Spares Prioritization and Availability to Resource Evaluation (M-SPARE) model to evaluate spare parts requirements for SSF and inform NASA program management, particularly with regard to budget planning for spare parts. M-SPARE optimized for station availability directly rather than using Expected Backorder (EBO) (described below) as a metric, and provided an integrated maintenance logistics estimation and spares procurement planning model for the SSF program that included planned Shuttle resupply cycles, spares procurement lead times, varying criticality for different ORUs, and multi-objective assessment of logistics mass and cost as a function of whether spares were stored on-orbit or on the ground, to be launched on demand. M-SPARE is effectively a multi-item, multi-echelon spares logistics optimization tool, built on the METRIC methodology, that was implemented in a (relatively) user-friendly computer program for use in SSF project offices at NASA centers and prime contractor offices, as well as at the Canadian Space Agency, for logistics and budget planning [88, 89].

The specific objective applied in most METRIC-based models is EBO, which is the expected number of backorders which occur across all bases in a given period of time. System availability – defined as the expected portion of a fleet of systems that is not offline for maintenance at a random point in time, or the portion of time that a single system is operational (for single-system fleets) – is maximized when the sum of backorders across all spares is minimized. METRIC, MOD-METRIC, and VARI-METRIC all use EBO as the primary metric for spares allocation optimization, due to its computational convenience [12, 80, 81, 84]. ASM and M-SPARE use availability, defined as the probability that no system is rendered inoperative due to a lack of a spare, as the objective, though EBO could also be used [83, 88, 89].

General Modeling Approach

For each of these models (METRIC, MOD-METRIC, VARI-METRIC, ASM, and M-SPARE) the process of addressing a failure in a multi-item, multi-echelon, multi-indenture case (described as the "physics" of the problem by Sherbrooke [12]) is as follows. A collection of systems are in operation at a set of one or more bases, which have a set of supply depots. A depot may provide supplies to more than one base. Each system consists of a set of first-indenture items, called LRUs (also known as ORUs, for spacecraft applications); each LRU consists of a set of second-indenture items, called SRUs. When a failure (i.e. a maintenance demand) occurs, the failed LRU is removed from the system. If a spare for that LRU is available at the base, it is used to replace the failed LRU and the system resumes operation. Otherwise, a backorder for that item exists – that is, a backorder is an instance of unsatisfied demand for a spare at the base level, a demand which causes a system to be non-operational due to a lack of spares. This backorder exists until it can be addressed, either by the delivery of another LRU spare from a depot, or by repair of the failed LRU at the base itself. If the LRU can be repaired at the base, typically by swapping out lower-level SRUs within the LRU, then the repair is scheduled. This causes a demand for a spare SRU, which may similarly cause a backorder and require ordering of new spares from the depot. Otherwise, the LRU may be sent to a depot for lower level maintenance. Time delays exist for maintenance actions and for shipment of spares from the depot to the base, and the time required for a spare to become available when a backorder exists impacts the amount of time that the system which failed remains offline for maintenance [12, 80, 81, 84, 85].

METRIC and its descendants model demand for various spare parts using a stochastic process described by the expected value (and variance, if a compound Poisson or negative binomial model is used) of the number of demands that an item may exhibit in a given period of time. This demand model characterizes an objective value for each item as a function of the number of spares provided, which may be:

- EBO, in the case of METRIC, MOD-METRIC, and VARI-METRIC [80–85],
- Availability, in the case of ASM [83], and/or
- Probability of Spare when Needed (PSN), in the case of M-SPARE [88, 89].

This last metric, PSN, is similar to POS, but accounts for the possibility that the desired inventory level for a given ORU is not achieved at the beginning of a logistics cycle due to a lack of spare availability for a Shuttle resupply launch [88, 89]. The total EBO is the sum of EBO for each item at the operating base. For both availability and PSN, the overall system metric is the product of the metric values for each individual items. That is, system availability is the product of availability for each individual item, and system PSN is the product of PSN for each individual item. This product can be transformed into a summation by applying a logarithmic transform: the natural logarithm of system availability is the sum of the natural logarithm of item availability, and the natural logarithm of system PSN is the sum of the natural logarithm of the PSN for each individual item. This approach results in a separable objective function – that is, one where the system-level value is the sum of the contributions from each of the items being considered [12, 83, 88, 89].

The optimal inventory is then identified by iteratively adding the most valuable spare, defined as the spare which provides the largest increase in objective value (EBO or PSN) per unit cost (which can be captured as financial cost, mass, volume, or some weighted sum of these). This benefit per unit cost is called the marginal value of the spare, and the process of iteratively adding the spare with the highest marginal value is known as marginal analysis, and is described in greater detail as part of the methodology used in this research (see Section 4.2.3). A detailed description of marginal analysis is provided by Fox [90]. For concave objective functions, the solutions that are iteratively generated by marginal analysis are Pareto-optimal; the approach can also be adapted for non-concave functions, as described in Section 4.2.1. However, marginal analysis does not necessarily identify all of the solutions in the Pareto frontier; that is, the algorithm may "skip over" some solutions. Dynamic programming (such as the branch and bound approach used in this research; see Section 4.2.3) can be applied to find solutions that may have been missed by marginal analysis [12, 83, 90].

Discussion

METRIC-based models provide a powerful technique for identifying the optimal spares inventory for a given system, and trading the cost of that inventory against risk. Epistemic uncertainty in failure rates can also be accounted for by using negative binomial distributions to model failure demand for items with gamma-distributed failure rates. However, though the time required to transport spares from one location to another, or to send an LRU through a lower-level maintenance process at a depot, are factored into the model, the amount of crew time spent on maintenance activities such as removing and replacing LRUs is not included. For example, the time required to receive a new spare once a backorder occurs depends upon the time taken to receive a new spare from the depot, or the time required to repair a spare at the base. However, this time includes shipping times, waiting times, and other supply chain activities that are not demands on crew time. In addition, the amount of this time used during a particular logistics cycle is not a metric tracked by these models. As a result, the family of METRIC-based models do not provide an assessment of crew time required for maintenance, either in terms of the POS associated with crew time or the total amount of crew hours spent on maintenance.

Moreover, though the multi-indenture capability of these models does examine multiple levels of maintenance within the spares supply chain, it does not do so in the sense of designing different levels of at which to define ORUs, which is the desired capability for this application. In the METRIC framework, system maintenance is executed at the LRU level. A spare LRU may be installed in the system, or the failed SRU may be pulled from the LRU itself, effectively treating the LRU as a "system" in a nested maintenance process. This is different from the level of maintenance challenge described in Section 1.5, in which the objective is to inform the selection of the level at which maintenance is performed on the system itself, rather than the level of spares which flow through the maintenance logistics supply chain. In addition, the METRIC approach takes the supportability characteristics of the system – including a definition of LRUs, SRUs, etc. – as an input, and calculates the associated risk-cost curve for that supportability strategy. METRIC-based inventory optimization models have been expanded by Sherbrooke [12] and Sleptchenko and van der Heijden [91] to consider ORU-level redundancy as part of the optimization; these models are described in Section 2.3.1. Another METRIC-based model, Spacecraft Sustainability Model™ (SSM™) [21, 92] (which is described in Section 2.1.4 since it incorporates a crew time model and other capabilities beyond the METRIC models described in this section), uses a METRIC-based approach for inventory optimization and can evaluate the impacts of commonality, redundancy, and level of maintenance, though it does not optimize over those decisions. Overall, METRIC-based models do not optimize over decisions other than spares allocations and redundancy, and do not include a model for ODM.

2.1.3 ISS Program Supportability

The ISS Logistics and Maintenance (L&M) Office manages maintenance logistics for the ISS, evaluating supportability using a combination of quantitative and qualitative assessments to balance logistics and risk against a number of other programmatic considerations, including failure criticalities, stowage capability, resupply launch schedules, station docking/undocking traffic, and crew time availability. ISS L&M, supported by a Boeing L&M team, also maintains the Modeling Analysis Data Set (MADS), a database of ISS ORU characteristics to support these analyses, including epistemic uncertainty in failure rate estimates. These failure rate estimates are updated on a regular basis using a Bayesian technique that accounts for observed failures (or lack thereof). The results of these analyses are used to inform a variety of activities, including launch manifests, spares procurement planning, and station risk assessments [13, 93, 94].

Uncertainty in failure rate estimates is captured using a lognormal distribution, characterized by a mean and an error factor. The error factor, defined as the ratio between the 50th and 95th percentiles, is a characterization of the amount of uncertainty present in the estimate, representative of the width of the probability distribution. Initial mean failure rate estimates were derived from MIL-HDBK-217 [95], with the error factor estimated based on the level of uncertainty in the estimate. As more data are gathered and failure rate estimates are refined, the error factor will tend to shrink and the failure rate uncertainty distribution will tend to converge to the true value [13, 93, 94, 96].

In some cases, epistemic uncertainty is disregarded, and the mean value of the failure rate uncertainty distribution is used directly to assess the POS associated with a given number of spares using a Poisson distribution. When epistemic uncertainty is considered, Monte Carlo simulations are used to examine the number of failures for each ORU that would occur when the failure rate is drawn from the corresponding lognormal uncertainty distribution described above. Spare parts can then be allocated in order to ensure that a desired POS is achieved with a desired level of confidence, where confidence is defined as the probability that the actual POS is greater than the desired value, given epistemic uncertainty.

Monte Carlo simulation is implemented in a software tool called the Reliability and Maintainability Assessment Tool (RMAT), which includes random failures as well as scheduled maintenance in order to assess the resources required to maintain station operations [97, 98]. When Monte Carlo simulation is used, failures can follow a Weibull distribution in addition to an exponential distribution, enabling modeling of infant mortality and wear-out failures in addition to end-of-life replacements. A more user-friendly version of RMAT, called the Comparative Maintenance Analysis Model (CMAM), was crated by Soldon [98] to facilitate deeper understanding of ORU failure rate data as well as sensitivity analyses. RMAT and CMAM both output the number of failures experienced by each ORU, as well as the associated expected crew time required for maintenance activities, including both preventative and corrective maintenance [97, 98].

Alternatively, the Probability and Confidence Trade-space (PACT) Evaluation methodology described by Anderson et al. [13] may be used to enable more dynamic trades between POS and confidence for a given ORU (rather than specifying fixed target values for each). However, PACT does not include crew time evaluation.

Regardless of the specific tool or methodology used, the results of the analysis are used along with other programmatic considerations (e.g. budget, lead times, launch constraints and scheduling, and maintenance/repair capabilities) to inform procurement, resupply scheduling, and station maintenance planning [93]. However, the ISS supportability analysis methodology is focused on supporting an existing system, and therefore it does not include any optimization or tradespace exploration of the impact of different system designs and supportability strategies. In addition, it does not include an evaluation of the impact of ODM.

2.1.4 Spacecraft Sustainability Model™

Bachman and Kline [21, 92] developed the SSM™ at LMI to determine optimal spares allocations using the same approach as those applied by METRIC and its descendants (described above). SSM™ utilizes either Poisson or negative binomial demand distributions to calculate EBO and system availability as a function of spares allocation, and applies marginal analysis to identify optimal allocations using mass, volume, financial cost, or a weighted sum of factors as a cost metric. In addition, SSM™ evaluates the expected amount of crew time required to replace ORUs, based on the expected number of ORU replacements and the expected time required to perform the action (the crew time modeling approach used by SSM™ is discussed further in Section 2.2.3). Finally, the model incorporates the ability to evaluate the impacts of commonality, redundancy, and level of maintenance on the relationship between availability, mass, volume, and crew time. The first two (commonality and redundancy) are evaluated directly, while level of maintenance impacts are evaluated using parametric scaling factors [21, 92]. These considerations are parameters to be input to the model, however, and not variables that are optimized.

Specifically, SSM™ examines level of maintenance by applying two adjustment factors to an ORU that can be maintained at a lower level (i.e. at the SRU level): a size factor that represents the estimated reduction in spares mass/volume enabled by SRU replacement rather than ORU replacement, and a maintenance factor that estimates the increase in maintenance time required to perform SRU replacement on the ORU. The mass/volume and maintenance crew time associated with a given ORU are multiplied by these two factors, respectively, before spares optimization is performed. This framework can be cleanly applied in cases where all failures of a given ORU are due to the specified SRU. In cases where some portion of failures may not be correctable by replacing the designated SRU, SSM™ uses a weighted sum to determine the average mass/volume/crew time required per replacement, as an approximation [21]. As a result, SSM™ provides an approximate approach lower-level maintenance analysis that applies a correction factor to ORU characteristics in order to simulate the effects of lower level maintenance rather than directly examining the effect of having ORUs defined at a different level in the system indenture structure.

SSM™ does not include any evaluation of the impact of ODM. In addition, though the inputs to SSM™ (e.g. an ORU list with mass, volume, reliability, and other information about each item) can be adapted to examine a range of system designs, and the model does include evaluation of redundancy and commonality, the model is not structured to examine or optimize over a wide range of supportability strategies.

2.1.5 Exploration Maintainability Analysis Tool

The EMAT was developed by Stromgren et al. [3] to examine logistics and support requirements for human spaceflight beyond LEO. In its initial formulation, EMAT used a Monte Carlo approach, executing thousands of simulated missions and observing stochastic ORU failure and maintenance activities to generate samples of the number of failures experienced by each ORU, as well as the impact of those failures. ORUs can have redundant instances, which serve as pre-installed spares. Statistical analysis of simulation results provides estimates of key metrics such as P(LoM), P(LoC), and POS for a particular mission, system, and spares allocation. More recent iterations of EMAT have used a binomial distribution to assess the Cumulative Distribution Function (CDF) of the number of failures observed by each ORU, rather than Monte Carlo simulation, based upon the daily probability of failure. Marginal analysis is used to identify the minimum-mass spares allocation that achieves a desired POS requirement, as well as to generate curves showing the tradeoff between spares mass and POS [1, 3, 22, 99].

EMAT can be used with the Exploration Crew Time Model (ECTM) [7, 100] (described in Section 2.2.4) to assess maintenance crew time requirements using either Monte Carlo simulation or direct calculation via the expected number of failures for each ORU. The outputs of EMAT can also be used along with the Wald ISM model [47] (described in Section 2.4.1) to asses the impact of the ability to recycle and reconfigure material during the mission.

In addition, the impact of epistemic uncertainty in failure rate estimates can be assessed using an additional layer of Monte Carlo simulation. In this approach, ORU MTBFs are modified for each Monte Carlo simulation using a scaling factor drawn from an empirical uncertainty distribution. The baseline uncertainty distribution was generated by examining the ratio of MADS MTBF estimates from 2016 to the initial estimates [9, see Fig. 3 on p. 8], and samples are linearly scaled to account for different ORU error factors. The average POS from all Monte Carlo samples is then used in the place of the deterministic POS from the original model to optimize spares allocations using marginal analysis. This process is repeated until the Monte Carlo simulations indicate that a desired POS has been reached with a desired level of confidence, an approach similar to that described by Anderson et al. [13]. EMAT then outputs required spares mass as a function of POS and confidence [9].

EMAT is a powerful supportability analysis tool that is used to assess maintenance logistics, risk, and crew time and inform a variety of NASA activities, including mission planning, system design, and technology assessment (e.g. [11, 22, 74, 101–105]). However, it is designed for the evaluation of the supportability characteristics of a specified system in order to guide design activities, rather than for optimization of system supportability strategy. While the EMAT framework has the flexibility to examine a wide variety of supportability strategies via variation of the input ORU set and its characteristics, it does not have the ability to generate and explore supportability strategies and resulting ORU lists itself. In addition, the Monte Carlo approach used to evaluate crew time distributions and epistemic uncertainty can require a significant amount of computational time, limiting the feasibility of EMAT analyses in cases where a large number (e.g. hundreds or thousands) of candidate system architectures are to be examined.

As discussed in Section 2.4.1, though EMAT can evaluate ISM via the parametric model described by Wald [47], this approach only examines the value of ISM in terms of enabling recycling and reconfiguration of material. This ISM model is a post-processing of results from EMAT, and as a result the model does not include an assessment of the risk-associated impacts of ISM/ODM and the value of commonality of material. Similarly, the evaluation of crew time requirements using ECTM [7, 100] (Section 2.2.4) is also a post-processing of EMAT results, and therefore the risk implications of crew time are not included in the spares analysis. That is, the tradeoff between crew time and maintenance logistics mass is not examined, since the two are assessed independently and EMAT examines a fixed supportability strategy rather than a set of options.

2.2 Crew Time Models

2.2.1 External Maintenance Task Team Report

One of the most well-known maintenance crew time assessments in human spaceflight history is the 1990 report of the SSF EMTT, commonly called the Fisher-Price report [19]. The history and implications of this report are discussed in Section 1.3, while this section discusses the Fisher-Price report from a methodological perspective.

To assess crew time demands, the EMTT identified a list of SSF ORUs that would require an Extravehicular Activity (EVA) for maintenance (since, as its name suggests, the team was focused on external maintenance demands). The failure rate, K-factor, and crew time required for maintenance (including overhead) were characterized for each ORU, and a Monte Carlo analysis was performed to estimate the expected number of failures that would occur as a function of time. The expected number of failures was then multiplied by the maintenance crew time required per failure to estimate the expected amount of crew time required for maintenance [19, 20].

The EMTT report provided valuable results that informed SSF (and later ISS) development, and it also included a qualitative discussion and set of recommendations for various supportability strategies that could be taken to address the issue of excessive maintenance crew time requirements, particularly with regard to level of maintenance. However, the EMTT used an analysis approach based on expected values which does not allow an evaluation of the amount of risk associated with a particular amount of crew time. Only R&R events were included in the analysis, since the number of maintenance actions was calculated using ORU failure rates. As a result, the EMTT analysis does not capture crew time associated with non-R&R events – its results only include crew time associated with ORU failures. In addition, the methodology does not examine the tradeoff between crew time and mass, since a spares logistics calculation was not included. Finally, ODM impacts, epistemic uncertainty in failure rates, and optimization of supportability strategies were not included in the methodology.

2.2.2 RMAT / CMAM

As described in Section 2.1.3, RMAT and CMAM (which are, for the most part, different implementations of the same model) include an assessment of crew time requirements in addition to ORU spares requirements [97, 98]. However, since these models focus on the evaluation of a given set of ORUs, they evaluate a fixed supportability strategy and do not include the ability to optimize over a range of supportability options. In addition, as with EMTT, both RMAT and CMAM link crew time directly to ORU failures. As a result, only the crew time associated with removal and replacement of failed ORUs is considered. Historically, a significant amount of maintenance crew time has been required for non-replacement events, and therefore this maintenance crew time modeling approach is incomplete.

2.2.3 Spacecraft Sustainability Model™

As noted in Section 2.1.4, SSM™ evaluates the expected amount of crew time required for maintenance by combining the expected number of ORU replacements (which is calculated as part of the spares allocation process) with the expected time required to remove and replace each ORU. This crew time evaluation is affected by all of the same factors that are used for the spares mass evaluation, meaning that changes in level of maintenance and the implementation of commonality or redundancy impact crew time as well as mass [21, 92].

However, SSM™ does not evaluate the risk associated with any given amount of available crew time. That is, the risk that there is not enough crew time to complete required maintenance is not evaluated, and the probability of sufficient crew time for maintenance is not considered during spares optimization or overall availability assessment. In addition, the only crew time included in the model is crew time associated with removal and replacement of failed components, since both crew time and spares demands are driven by the same stochastic processes. Crew time associated with non-replacement events (which can be significant) is not evaluated, and thus the expected crew time evaluated by SSM™ captures only a portion of the total crew time required for maintenance.

2.2.4 Exploration Crew Time Model (EMAT Extension)

EMAT (described in Section 2.1.5) was initially developed to assess the relationship between mass and risk. However, it has since been augmented to enable maintenance crew time assessment via ECTM. Mattfeld et al. [7] and Stromgren et al. [100] developed ECTM to forecast crew time requirements on future Mars missions, including time demands from utilization, crew health, and maintenance activities. The initial version of ECTM was called the Integrated Crew Operations Model (ICOM) [7]. The model includes distributions for the amount of crew time spent on various activities based on ISS experience from ISS operations from July 20, 2011 (the day after the final Space Shuttle mission undocked from the ISS) to the end of 2016, based on historical entries in Operations Planning Timeline Integration System (OPTimIS), a detailed schedule used by the ISS Operations Planning team to develop and manage crew and ground control schedules for ISS operations. Data from ISS operations archives (covering the time period from 2011 to 2018) such as visiting vehicle traffic and EVA activities, as well as maintenance and repair activities recorded in the Maintenance Data Collection (MDC) database and MADS, were used to augment OPTimIS data and provide context for the various activities recorded in the crew schedule. Crew activities were categorized according to a detailed ontology, including category (i.e. work or non-work activities), sub-category (e.g. scheduled operations, operations preparation and conference), activity (e.g. vehicle operations, EVA, exercise, upkeep), sub-activity (e.g. vehicle loading/unloading, maintenance), and operation type (e.g. docking/undocking, corrective maintenance). This categorization enables an assessment of how much crew time was spent on different types of activities on the ISS, which can then be used to inform forecasts of how much time will be required on future missions [7, 100].

Whereas some crew activities (e.g. sleep, exercise) are relatively independent of system architecture, the amount of maintenance spent on crew time is strongly dependent on the supportability characteristics of the system itself. ECTM therefore evaluates maintenance crew time using EMAT (described in Section 2.1.5). ORU reliability data is used to estimate the expected number of failures for a given ORU, which is then combined with the estimated maintenance times for that ORU in order to generate an overall estimate of the amount of crew time spent on maintenance [22]. EMAT-based Monte Carlo simulation can also be used to assess the distribution of the total amount of crew time required and extract the amount of crew time associated with a particular confidence level. For example, Mattfeld et al. [7] take the amount of maintenance crew time with a 90% confidence level, meaning the amount of crew time that would need to be allocated, based on Monte Carlo simulation, to be sufficient 90% of the time.

From a supportability perspective, the maintenance crew time evaluation portion of ECTM can be thought of as an augmentation of EMAT that uses either the expected number of failures or Monte Carlo simulation to determine the expected amount and/or distribution of crew time spent on maintenance. This approach enables the evaluation of the relationship between crew time and risk, as well as forecasting of crew time requirements for the purposes of mission utilization planning. However, EMAT/ECTM uses ORU failure rates to assess both spares and crew time requirements, and therefore only includes crew time required for removal and replacement of failed ORU. As noted in Section 1.3, a significant amount of crew time is spent on activities that do not involve the replacement of ORU, including stochastic activities such as troubleshooting. ECTM does not include these activities in its assessment, and therefore likely underestimates maintenance crew time requirements. In addition, while the probability of insufficient crew time can be calculated using ECTM, it is not used to inform overall POS calculations or influence spares allocations. That is, the probability of sufficient crew time and the probability of sufficient spares are only considered separately, rather than as two contributors to the overall probability of sufficient maintenance resources. As a result, EMAT/ECTM effectively performs separate evaluations of mass, risk, and crew time associated with maintenance, rather than an integrated evaluation.

2.3 Design and Maintenance Strategy Optimization

This section describes modeling and optimization methodologies that focus on the supportability characteristics of the system itself, as opposed to the spares inventory models described in the previous sections. Some of these models, particularly those related to redundancy allocation, do include an assessment of spares requirements as part of their cost function, and therefore could be used to assess mass. However, they are described in this section rather than in Section 2.1 since their primary application is system design optimization rather than spares mass evaluation.

2.3.1 Redundancy Allocation

One supportability strategy element which has been the subject of significant research is the Redundancy Allocation Problem (RAP), in which the objective is to allocate redundancy levels to components and subsystems within a combined series-parallel system structure in order to maximize system reliability subject to constraints on other parameters such as cost and weight; alternatively the problem can be formulated to minimize cost subject to a constraint on reliability [106]. A variety of modeling and optimization techniques have been applied to RAP over the past several decades; however, the majority of related research focuses on the optimization of the characteristics of the system itself (cost, mass, availability, etc.), and logistical considerations such as spares inventories are not often considered. However, RAP modeling has been expanded in places to include the logistics impacts of redundancy decisions. Relevant examples of RAP modeling and optimization techniques are described below.

Tian et al. [107] present a model for joint optimization of reliability and redundancy for multi-state series-parallel systems, minimizing total cost while meeting a specified system availability requirement. Various pre-specified types of each component are available, each of which has different failure and repair rates, as well as different costs. In addition, different organizational strategies can be implemented to affect failure and repair rates; these organizational strategies, implemented using scaling factors on component failure/repair rates, represent decisions such as the allocation of more resources towards repair activities. The component failure model includes multiple states, and as such represents system degradation rather than the traditional binary operational/failed states. While it does not include logistics, this modeling approach does incorporate a multiobjective examination of different supportability design decisions, including both redundancy allocation as well as the selection of different levels of component quality/cost. A Markov model is used to represent component behavior and calculate component state probabilities, and Universal Generating Functions (UGFs) are used to calculate system state probabilities based upon the state probabilities of the constituent components and the structure of the system (i.e. the level of series and/or parallel redundancy). Component failures and repairs are assumed to follow an exponential distribution, and optimization is performed using a Genetic Algorithm (GA) [107].

Sleptchenko and van der Heijden [91] present a model for joint optimization of redundancy levels and spare part inventories. Redundant components may be in any one of three standby modes: "hot," "warm," or "cold," meaning that the standby component's failure rate is equal to the active component, less than the active component, or equal to zero, respectively. Both redundancy and sparing are implemented at the same level, so that when a failure occurs a redundant instance can maintain system function while maintenance takes place. A formulation using continuous-time finite-state Markov chains is applied to calculate exact availability, though the authors node that this approach does not scale well for larger systems and also present an approximation. The objective is to minimize costs, including the cost of installed systems as well as spares inventory costs, subject to a specified system availability requirement. The overall optimization uses a heuristic, nested approach in which – based on the observation that the addition of a redundant component tends to have a much larger impact on availability and cost than the addition of a spare – redundant components are iteratively added to the system and spares allocations for each level of redundancy are optimized using the same approach used by METRIC-based analyses (see Section 2.1.2) [91].

Sherbrooke [12] presents a METRIC-based model for joint optimization of redundancy and spares inventory, applying the EBO approach described in Section 2.1.2 to model the cost-availability curve for a k-out-of-N system – that is, a system in which there are a total of N components, k of which must be operational to maintain system function. The system is considered available if the number of backorders is less than $N - k$ (i.e. if there are at least k operational systems). The approach involves enumerating of all system configurations associated with each level of backorders (0, 1, 2, etc.), calculating the probability associated with each using the hypergeometric distribution, and adding up the probabilities associated with configurations that have less than $N - k$ backorders. This approach incurs a significant computational cost and does not scale well with the number of systems considered. However, de Smidt-Destombes et al. [108] expand upon the initial approach and present two simplified (and less computationally expensive) formulations. The first applies a recursive formula to calculate availability; the second, an approximation, uses a moment-matching technique to approximate the distribution of the number of backorders. In all cases, redundancy and sparing are assumed to be applied at the same level. Using any of these models – which relate levels of redundancy and spares inventory to availability – marginal analysis is applied to optimize the allocation of both spares and redundancy. That is, at each step in the procedure either the level of redundancy or the number of spares allocated to a particular item is increased by one until the desired availability level is achieved [12, 108].

The RAP methodologies described here have the ability to examine a range of redundancy options. The method described by Tian et al. [107] includes design decisions related to component reliability and repair time, while the methods described by Sleptchenko and van der Heijden [91], Sherbrooke [12], and de Smidt-Destombes et al. [108] include spares allocation optimization. However, while some of these approaches include the effect of maintenance downtime in their evaluation of system availability, none of them assess the amount of crew time required for maintenance or use it as an objective to be minimized. In addition, they do not include other supportability strategy decisions such as level of maintenance, commonality, distributed functionality, or ODM.

2.3.2 Multistate Analysis and Design

Agte [76], Agte et al. [109], and Borer and Agte [110] present a multistate analysis and design framework that uses Markov chains to examine the tradeoff between nominal system performance, availability, expected performance, and cost for long-endurance systems. In particular, this framework enabled concurrent examination and optimization of both static design variables and reliability characteristics. The framework was applied to the design of long-endurance UAVs, and demonstrated that this approach could identify designs with better system characteristics than optimization of either static design variables or reliability characteristics alone. However, the multistate analysis and design approach focused on the assessment of cost and risk for systems that could not be maintained during operations, and therefore it did not include an assessment of maintenance logistics or crew time requirements.

2.3.3 Joint Optimization of Design and Maintenance

Adjoul et al. [111] developed a technique for joint optimization of system design and maintenance strategy for multi-component systems that undergo a series of missions of fixed duration during which no planned maintenance occurs (but corrective maintenance can occur, if needed), each of which is followed by a system shutdown for preventative maintenance activities. The time between preventative maintenance shutdowns is known as the Maintenance Free Operating Period (MFOP). The reliability metric for the system under these conditions is the Maintenance Free Operating Period Survivability (MFOPS), defined as the probability that the system survives (i.e. does not fail) for the entire MFOP, given that it was operational at the beginning of that MFOP. System lifecycle cost is calculated using a combination of parametric models to assess initial cost and Monte Carlo simulation to assess maintenance demand and resulting costs. Initial cost considers the cost of each component (which may increase/decrease as a function of desired reliability and/or accessibility), as well as the number of redundant instances and any sensors used for component health monitoring. Maintenance costs include both corrective and preventative actions. Monte Carlo simulation is used to assess the number of random failures which occur for each component. The cost per maintenance action includes the cost of the replacement component as well as associated logistics and labor. For corrective maintenance, the cost of diagnosing the failure is also included. Variations in component reliabilities, level of redundancy, accessibility, and component health monitoring infrastructure, as well as preventative maintenance policy (i.e. which components are replaced at the end of each MFOP) are investigated by performing Monte Carlo simulation and assessing the expected lifecycle cost under a particular system design and maintenance strategy.

This approach assesses the cost of a particular system design and maintenance strategy using Monte Carlo simulation to observe the number of failures that occur during a series of simulated missions, and uses the results to determine the expected lifecycle cost. Spares and crew time (in this case, labor hours) are assumed to be available (at a cost) when needed, and there is no up-front allocation of maintenance resources. The risk-related portion of the model is captured by the maintenance policy, which involves the selection of components to replace during each preventative maintenance period (i.e. at the end of each MFOP) in order to achieve a desired MFOPS. As a result, this model does not examine the problem of selecting a limited allocation of spare parts before a mission, but rather focuses on optimizing scheduled preventative maintenance activities in order to minimize overall operating costs. The additional cost of making spares allocation decisions under uncertainty – i.e. before a failure occurs, as opposed to in response to a failure – is therefore not included, and as a result this modeling approach is more applicable to continuously operating systems with robust, always-accessible supply chains and regular maintenance overhauls than it is to logistically isolated deep-space missions. In addition, the model does not include any assessment of ODM.

2.3.4 Generalized Resilient Design Framework

Matelli and Goebel [112] applied the Generalized Resilient Design Framework (GRDF) to examine ECLSS resilience and inform system design. GRDF uses rule-driven Monte Carlo simulation to examine degradation and failure within a system over the course of a given mission. No maintenance or repair is considered. Random failures within the system can propogate according to a set of pre-defined rules, and a set of metrics are used to examine the resilience of the system, including, for example, the average operating time, time until failure, and probability of failure. While valuable for examining failure propagation within systems, as well as the likely outcomes of maintenance-free operations for risk assessment, the lack of maintenance activities within the model means that this model does not include logistics or crew time assessment, nor does it incorporate any assessment of ODM.

2.4 In-Space Manufacturing and Supportability

This section describes models relating ODM (sometimes referred to as ISM) to system supportability. ISM is a relatively new technology that enables ODM for deep space missions, as described in Section 1.2.5. As a result, the vast majority of discussions of ISM, or ODM for space missions, found in the literature focus on the development of ISM technology itself, or on a qualitative discussion of the potential impacts that it may have (e.g. [51–57]).

Three papers describing quantitative models of ISM impacts on space mission supportability were found in the literature: a parametric model developed by Wald [47], a working paper by Zhao et al. [113], and another parametric model developed by Sears and Ho [114]. The models presented in those papers are described below. Other models in the literature that assess the impact of ODM/ISM on spaceflight logistics are techniques that were developed as part of this research [24, 25]. These techniques have been refined and incorporated into the evaluation methodology described in

2.4.1 Wald 2015

Wald [47] developed a high-level parametric model to examine the effect of ISM on logistics requirements for a long-duration Mars transit habitat, focusing on the ability to recycle material and reconfigure components during the mission. This model was implemented within Exploration Architecture Model for In-Space and Earth-to-orbit (EXAMINE) [115], a parametric space mission architecture modeling, element sizing, and logistics framework developed in the Space Mission Analysis Branch (SMAB) at NASA Langley Research Center (LaRC). EXAMINE includes a broad set of capabilities that are beyond the scope of supportability analysis, but for logistics and supportability assessment purposes EXAMINE facilitates the estimation of a fixed initial system mass plus a time-dependent logistics mass. EXAMINE can also be paired with EMAT (Section 2.1.5) for a more detailed assessment of maintenance logistics [47, 115, 116]. The Wald ISM model assesses the mass savings that could be achieved by transforming products that might otherwise become waste (e.g. spent components, packaging, etc.) into useful items, thereby eliminating the need to launch the latter. Various levels of ISM capability could be evaluated by varying the amount and types of resources that could be recycled or reconfigured [47].

This parametric ISM approach focuses on the ability to recycle material, but it does not include the impact of ODM on spares allocations. Specifically, this model focuses on reducing initial mass allocations by transforming waste products into useful items, rather than by providing the benefits of commonality of material (described in Section 1.2.5) that reduce initial spares mass requirements directly. In addition, it does not include any evaluation of the relationship between logistics and risk.

2.4.2 Zhao et al. 2014

Zhao et al. [113] examined the break-even point of combined ISRU and spare parts manufacturing on Mars, calculated in terms of the total mass transported to

Mars. To assess spare parts requirements, individual spares were modeled as a mass flow at a continuous rate; that is, the stochastic spares demand process was simplified to a deterministic spares mass consumption rate. An ISM capability was assumed for a subset of items requiring spares, meaning that the spares mass for those items would be produced on-site rather than transported from Earth. The mass and power requirements of metal and plastic parts manufacturing systems, both of which use ISRU-derived feedstock, are estimated based on comparison to terrestrial systems. The break-even point for spares manufacturing on Mars is then calculated by comparing the mass profile over time for the no-ISM case, which has no initial production system mass, but has a higher time-dependent spares mass, to the case with ISM, which has a fixed initial production system mass but a reduced time-dependent spares mass [113].

The Zhao ISM model utilizes a simplified spares demand model that transforms a random process into a deterministic spares mass flow rate. As a result, it does not include any consideration of risk. In addition, it focuses on spares mass reduction via in-situ production using in-situ resources, rather than examining any impacts of commonality of material. This approach effectively considers an ISM system to be part of an ISRU system, and focuses on ISRU spares logistics rather than the benefits that could be provided by ISM even if ISRU is not available. Finally, this model did not examine other supportability strategy elements (e.g. commonality, level of maintenance), and did not include an assessment of crew time.

2.4.3 Sears and Ho 2018

Sears and Ho [114] presented a framework to evaluate the incorporation of ISM into a robotic on-orbit servicing operations. Within this framework, a mothership houses a spare parts supply for a fleet of satellites, along with robotic servicers that are deployed to those satellites to deliver and install spares when failures occur. When ISM is available, the mothership uses an AM system to manufacture replacement parts to replenish its stock when the servicer delivers a new spare to a failed satellite. Failed parts can be recycled into AM feedstock once they are returned to the mothership by the robotic servicer.

This model is similar to the one presented by Wald [47] (Section 2.4.1), in that it focuses on ISM as a means to enable material recycling, rather than examining it from the perspective of covering multiple failure modes using common raw materials. Launches from Earth are used to replenish stores of spares and/or feedstock in the mothership as needed, when supply levels fall below pre-determined levels. There is no evaluation of the tradeoff between risk and resource allocations. As such, the operating mode for this system is not applicable to deep-space missions. In addition, since this model focuses on robotic satellite servicing, crew time is not examined.

2.5 Summary and Identification of Research Gap

Table 2.1 summarizes the results of this literature review, focusing on the models most closely related to the objectives of this research. The heuristic approach [28, 79], METRIC-based models [12], ISS supportability models [13, 94, 98], SSM™ [21, 92], EMAT (which is assumed for the purposes of this discussion to include ECTM and the Wald ISM model) [3, 7, 47, 100], EMTT [19], and RAP approaches that include spares allocation optimization [12, 91, 108] are included in the table. The model for joint reliability-redundancy optimization described by Tian et al. [107], Multistate Analysis and Design [76, 109, 110], Joint Optimization of Design and Maintenance [111], and GRDF [112] are not included since they do not evaluate mass or crew time. Only the Wald ISM model [47] is included (as part of EMAT), since the models presented by Zhao et al. [113] and Sears and Ho [114] use a similar approach and only the Wald model is incorporated into a supportability evaluation framework that assesses mass and crew time.

For each model, Table 2.1 identifies the sections describing that approach as well as a few key references. The remaining rows in the table then indicate if and to what extent the model includes key objectives of this research. Cells are filled in green if the model completely fulfills the criteria developed as part of the motivation of this research (see Section 1.5). They are filled in red if the model does not, and in orange if

Table 2.1: Summary of literature review results.

Method	**Heuristic**	**METRIC**	**ISS Supportability**	**SSM**	**EMAT[a]**	**EMTT**	**RAP**
Section(s)	2.1.1	2.1.2	2.1.3, 2.2.2	2.1.4, 2.2.3	2.1.5, 2.2.4, 2.4.1	2.2.1	2.3.1
Key References	Leath and Green 1993; Grenouilleau 1999	Sherbrooke 2004	Anderson et al. 2012; Vitali and Lutomski 2004; Soldon 2004	Bachman and Kline 2004; Kline and Bachman 2007	Stromgren et al. 2012; Mattfeld et al. 2015; Stromgren et al. 2018; Wald 2015	Fisher and Price 1990	Sherbrooke 2004; de Smidt-Destombes et al. 2011; Sleptchenko and van der Heijden 2016
Mass	**Y**	**Y**	**Y**	**Y**	**Y**	**N**	**Y**
Crew Time	**N**	**N**	**P[b]**	**P[b]**	**P[b]**	**P[b]**	**N**
Risk	**N**	**P[c]**	**P[c]**	**P[c]**	**P[c]**	**N**	**P[c]**
Epistemic Uncertainty	**N**	**Y**	**Y**	**Y**	**Y**	**N**	**Y**
Level of Maintenance	**N**	**C**	**N**	**C**	**N**	**N**	**N**
Commonality	**N**	**C**	**N**	**C**	**N**	**N**	**N**
ODM	**N**	**N**	**N**	**N**	**P[d]**	**N**	**N**
Redundancy	**N**	**O**	**C**	**C**	**C**	**N**	**O**
Distributed Functionality	**N**	**N**	**N**	**N**	**N**	**N**	**N**

Y: Included in model
N: Not included in model
P: Partially included in model
C: Considered, but not optimized
O: Optimized

[a] EMAT is considered to include ECTM and the Wald ISM model
[b] Only evaluates crew time associated with ORU removal and replacement
[c] Does not include crew time in POS assessment
[d] Only assesses impacts of recycling; does not include commonality of material

the model partially accomplishes the desired objectives. The first three rows indicate whether the model evaluates maintenance mass and crew time, as well as whether risk is taken into account in that evaluation. The next row indicates whether epistemic uncertainty is taken into account during the calculation. The remaining rows are associated with the supportability strategies indicated in this research, and indicate whether or not that strategy is considered and/or optimized as part of the model. A strategy is considered in a model if it is one of the model inputs; it is optimized if the value of that input can be optimized in a single- or multi-objective fashion.

Note that nearly all of these models (with the exception of heuristic methods) could be used to evaluate various ORU sets with different characteristics, and therefore could be adapted to consider many strategies as a function of the supplied ORU list. However, this type of analysis would require an additional modeling effort to interpret a supportability strategy decision into an ORU list; that supportability strategy interpretation capability is not currently included in those models, and therefore they are not considered to include those factors in their evaluation.

The primary gap identified in this research relates to evaluation and optimization of a wide range of supportability strategies. As the last five rows of Table 2.1 indicate, most supportability models consider only a subset of supportability strategies (typically level of maintenance, commonality, and redundancy) as inputs. This is largely due to the fact that, for the most part, these models are designed to evaluate existing systems, and take a pre-defined ORU list as an input rather than taking in a generic system definition and a supportability strategy, then generating and evaluating an ORU list based on that strategy. The only models found in this review which optimize an element of supportability strategy while also evaluating mass (though they do not evaluate crew time) are the RAP models presented by Sherbrooke [12] (which is also a METRIC-based model) and Sleptchenko and van der Heijden [91], which optimize redundancy at the ORU level alongside spares allocations. No model considers the option of distributing functionality across multiple smaller modules, and the only model to include ODM as part of its mass evaluation is EMAT, which uses the Wald ISM model. However, this ODM evaluation only considers the impact of material recycling, and does not include the effects of commonality of material.

In fact, as noted in Section 2.4, none of the ODM models identified in this literature review evaluate the impacts of ODM beyond recycling or use of ISRU to generate raw materials for spares. Specifically, they do not evaluate the POS associated with a given amount of raw materials for ODM, and thus are not applicable for assessment of trades between risk and mass. The only models found during the literature search which evaluate commonality of material and other risk implications of ODM / ISM were developed as part of this research [24, 25], and form the basis for the ODM model described as part of the methodology presented in the next chapter. This lack of ODM models represents the second significant gap in the literature.

From a metrics perspective, only the ISS supportability models, SSM™, and EMAT evaluate both mass and crew time. However, it is important to note that, for all of the models identified in the literature, crew time is calculated by associating a crew time demand with each ORU failure. That is, crew time demand is driven by ORU failure rates, and as a result only includes crew time associated with removal and replacement of failed ORUs. As a result, these models do not include crew time spent on non-replacement activities such as inspection and troubleshooting, which can be a significant source of demand. All of these use a risk-based evaluation for mass, since spares are allocated in order to meet a POS or availability objective. However, in all of these models risk is only assessed as a function of the amount of spares allocated. That is, the probability of insufficient crew time is not included and these models only partially capture the risks associated with insufficient maintenance resources. The lack of non-replacement activities in crew time evaluation and the lack of probability of sufficient crew time in overall POS evaluation are the third and fourth gaps identified by this literature review.

Finally, all of the models that evaluate the distribution of maintenance crew time in addition to expected value (and some models that examine only expected value) do so via Monte Carlo simulation. This approach can be slow and computationally expensive, particularly if the values of interest are associated with very high or very low probabilities. While this approach is effective for evaluation of a single supportability strategy, or a handful of strategies at a time, the computational cost associated with Monte Carlo simulation can become prohibitively expensive when large supportability strategy tradespaces are evaluated, or when multiobjective optimization is performed which requires repeated evaluation of candidate strategies. As a result, these Monte Carlo evaluation approaches are not scalable to larger, more complex problems and present a barrier to systematic supportability strategy optimization and tradespace exploration.

In summary, the gaps identified by this literature review are as follows:

1. No supportability model found in the literature considers or optimizes over a wide range of supportability strategy options (level of maintenance, commonality, ODM, redundancy, and distributed functionality) in order to minimize mass and maintenance crew time.

2. No existing ODM models evaluate the relationship between the amount of raw materials supplied for ODM and the probability that the amount provided is sufficient. As a result, no existing ODM models capture the impacts of commonality of material on logistics mass and risk.

3. None of the models found in this literature review include time spent on non-replacement activities in their assessment of maintenance crew time requirements. Instead, all models that assess crew time do so using ORU failure rates and as a result only capture crew time associated with removal and replacement of failed ORUs.

4. No models found in literature consider the probability of sufficient crew time when evaluating whether a particular system and spares allocation fulfills a specified POS requirement.

5. All models evaluating the distribution of crew time use Monte Carlo simulation, which limits their extensibility to large-scale supportability strategy tradespace exploration and optimization.

2.6 Research Question

The research question addressed by this work is:

> *What is the Pareto-optimal set of supportability strategies that minimize mass and crew time required for maintenance for future long-endurance crewed space missions, given a particular system, mission, and risk requirement?*

In particular, the supportability strategy elements to be considered include level of maintenance, commonality, ODM, redundancy, and distributed functionality. The ODM evaluation will assess the POS associated with different levels of raw feedstock allocation, enabling the incorporation of ODM into overall spares allocation optimization. Crew time evaluation will include non-replacement events such as inspection or troubleshooting in addition to time spent on ORU removal and replacement, and the probability of sufficient crew time will be included in the evaluation of overall maintenance resource POS. Finally, both aleatory and epistemic uncertainty will be considered in all of these evaluations, and analysis will be executed without the use of Monte Carlo simulation. The resulting supportability model, described in Chapters 3 and 4, addresses all of the research gaps listed above.

Chapter 3

Methodology I: Supportability Strategy Optimization Problem Formulation

The following two chapters present a methodology for multiobjective optimization of supportability strategies for deep space missions, designed to address the domain challenges and research gaps identified in Chapters 1 and 2. This chapter summarizes the methodology in general and formally defines the supportability strategy optimization problem, based on the research question defined in Section 2.6. The problem formulation presented here includes an explicit mathematical definition of the inputs, decision variables, objectives, and constraints required to represent supportability strategy optimization as a multiobjective Constraint Optimization Problem (COP). This supportability strategy optimization problem formulation is a key methodological contribution of this research, since it fills the gap related to lack of strategy optimization capability in existing models and provides the means to answer the research question defined in Chapter 2. Specifically, the definition of decision variables capturing a range of supportability strategies, along with a constraint set defining feasible assignments and definitions of inputs, objective values, and relationships between all of these elements enables the identification of Pareto-optimal supportability strategies for a given mission and system. In addition, the definition of crew action rates as separate from failure rates is a novel contribution of this methodology, since it separates spares and crew time demands and enables more accurate crew time forecasting. Chapter 4 then presents the model used to evaluate the mass and crew time associated with a particular assignment to the supportability strategy decision variables, as well as heuristics for evaluating bounds on objective values for partial assignments to the decision variables, and the search strategy used to solve the multiobjective COP.

3.1 Modeling Objectives and Overview

The objective of the supportability strategy optimization problem is to identify and characterize the set of Pareto-optimal supportability strategies that minimize the amount of mass and crew time required for maintenance for a given mission and system. The supportability strategy optimization problem is formulated as a multi-objective COP, using decision variables encoding five types of supportability strategy decisions: level of maintenance, commonality, ODM, distributed functionality, and redundancy. An assignment to the supportability strategy decision variables is used along with a mission and system description to define an ORU list for that mission, where an ORU is defined as a component or set of components that have been encapsulated into a single unit in order to facilitate maintenance. This ORU list is then assessed using a supportability model that determines the mass and crew time required to achieve a desired level of POS using that supportability strategy. A flowchart depicting the overall analysis approach is presented in Figure 3.1. The methodology described here is implemented in an analysis tool called MCROSS.

This analysis is designed to occur after system architecture selection and function-to-form mapping, since characteristics of the physical system itself are required to assess mass and crew time requirements. A purely functional description of a system has no supportability characteristics of its own. The supportability characteristics of a system are related to the characteristics of the physical system itself (and its associated subsystems, assemblies, and components) rather than its abstract func-

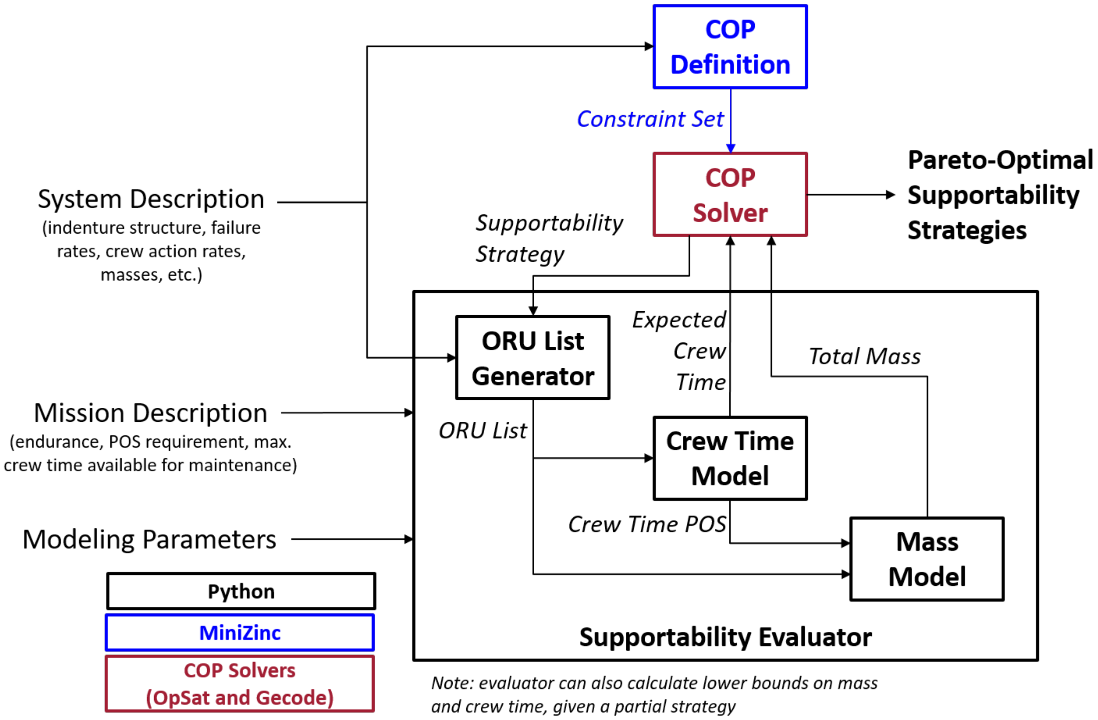
COP Definition
Constraint Set
COP Solver
Pareto-Optimal Supportability Strategies
System Description
(indenture structure, failure rates, crew action rates, masses, etc.)
Supportability Strategy
Expected Crew Time
Total Mass
ORU List Generator
Mission Description
(endurance, POS requirement, max. crew time available for maintenance)
ORU List
Crew Time Model
Crew Time POS
Mass Model
Modeling Parameters
Python
MiniZinc
COP Solvers (OpSat and Gecode)
Supportability Evaluator
Note: evaluator can also calculate lower bounds on mass and crew time, given a partial strategy

Figure 3.1: Flowchart of the analysis methodology. Boxes and text in black are portions associated with Python modules written for this research. The blue box and text indicate portions written in MiniZinc [117], with interfaces handled by the PyMzn module [118]. The red box indicates COP solvers (OpSat [119, 120] or Gecode [121], depending on whether optimization or full enumeration is performed) that are interfaced with the rest of the model. Given a full assignment to the supportability strategy variables, the evaluator returns the expected crew time and total mass associated with a given system and mission. Given a partial assignment, the evaluator returns a lower bound on both objective values.

tion. However, this does not mean that supportability analysis must come at the end of system development, once the physical system is fully defined. On the contrary, supportability analysis is most useful when it is performed early and often, informing system design decisions at each step of an iterative process. During early phases, when specific data about system characteristics such as mass and failure rates, these characteristics can be approximated based on comparison to existing systems or expert opinion. As system design matures and more information is gathered – particularly the results of test campaigns that can illuminate the true values of failure rates and crew time requirements – the supportability analysis can be repeated and refined. At each step of the way, sensitivity analysis can be used to explore a range of options for various design and operational decisions in order to provide results that can guide system development from a supportability perspective. Overall, supportability analysis is not an activity that is performed once in the system development lifecycle, like a box to be checked or a gate to be passed. Instead, it is a tool to be used throughout the process in order to inform decision-making and understand the characteristics of the system as it develops.

This chapter formulates the supportability strategy optimization problem itself, defining relevant inputs, decision variables, objectives, and constraints. The inputs, described in Section 3.2.1, define the context of the problem in terms of the form of the system being examined, the mission that the system will operate in, and various other parameters used by the model. Section 3.2.2 defines a set of decision variables that encode decisions related to level of maintenance, ODM, commonality, redundancy, and distributed functionality. An assignment to these variables defines a supportability strategy. Section 3.2.3 defines the objectives, specifically mass and crew time, which are to be minimized by the model. Section 3.2.4 defines constraints between the decision variables, as well as constraints on valid objective values. Taken together, these portions of Section 3.2 define the supportability strategy optimization problem as a multiobjective COP, summarized in equation 3.36 on page 148.

Chapter 4 presents the methodology for evaluating objective values based on an assignment to the supportability strategy decision variables. Section 4.1 describes how supportability strategy decisions are used to define the set of ORUs for a given system, mission, and supportability strategy. The ORU list defines the maintenance demands associated with that set of inputs and decision variables, and Section 4.2 describes the supportability model used to assess mass and crew time for those ORUs. Section 4.3 describes the multiobjective branch and bound strategy used to find the Pareto frontier, describes the methodology used to characterize the lower bound on partial assignments to the decision variables, and presents variable and domain ordering heuristics used to accelerate the search process. Finally, Section 4.4 presents a summary of the methodology define in Chapters 3 and 4, including a description of key research contributions and the implementation of this methodology in the MCROSS analysis tool.

3.2 Problem Definition

The objective of a supportability strategy optimization problem is to identify and characterize the set of Pareto-optimal supportability strategies that minimize total mass and expected crew time required for maintenance for a given system and mission. For the purposes of this research, a system is the physical spacecraft, systems, processors, and any other equipment that is to be included in the supportability assessment. A mission is the operating context of that system – how long it must operate without resupply, what risk threshold must be met, and the amount of crew time available for maintenance activities. A supportability strategy is the set of decisions relating to level of maintenance, commonality, ODM, distributed functionality, and redundancy that drive maintenance mass and crew time requirements and their relationship to risk. The supportability demand for a particular mission, measured for this research in terms of the amount of mass and crew time required, is driven by the system, mission, and particular supportability strategy taken.

This research formulates the supportability strategy optimization problem as a multi-objective COP, where the constraints describe feasible assignments to the supportability strategy variables. This section defines the supportability strategy optimization problem in general, describing its inputs, decision variables, objectives, and constraints. The supportability strategy optimization problem formulation described here is one of the primary contributions of this thesis, and is key to answering the research question described in Section 2.6 and addressing a key research gap. The majority of previous research into supportability modeling and evaluation, described in detail in Chapter 2, focuses on evaluation of a fixed system and supportability strategy. These previous approaches effectively start from a fully-defined ORU list and evaluate the mass and/or crew time required for maintenance. When supportability strategies are considered as part of system design optimization, models focus on one or two supportability characteristics at a time, such as redundancy or reliability, and do not evaluate the logistics requirements associated with long-term isolated operations. The formulation presented in this research, on the other hand, provides a framework to examine and and perform multiobjective optimization concurrently over a wide range of supportability strategies. A key innovation in particular is the definition of variables and constraints capturing level of maintenance decisions, enabling the definition of arbitrary assemblies (within the system hierarchy defined by the indenture structure) during optimization and much greater flexibility in exploring various options for ORU definition. By defining supportability strategy decision variables and a process for using assignments to them to define and characterize an ORU list, this research enables the application of formal optimization techniques to supportability decision-making.

3.2.1 Inputs

A supportability strategy optimization problem is defined by three inputs: a system description $\mathcal{S}$, a mission description $\mathcal{M}$, and a set of modeling parameters $\mathcal{P}$. Each of these consists of a set of elements providing all of the information required to define the problem – namely, analysis of the supportability requirements of a particular system in a particular mission context, under a particular set of assumptions and parameters. These inputs are described in the following sections, and a summary is provided in Table 3.1 on page 117.

Table 3.1: Summary of inputs to the supportability strategy optimization problem, with associated symbols and units where applicable. Indentation levels indicate nesting of input data. For example, the mission description consists of endurance, required POS, and maximum maintenance crew time. Each input characteristic below the item list is specified for each item i in the item list, or some subset of items, based on the given condition.

Name	Symbol		Units
System Description	$\mathcal{S}$		
Item List	ϕ		
Parent	p_i	$\forall\ i \in \phi$	
Component Indicator	θ_i	$\forall\ i \in \phi$	
Subsystem Indicator	ψ_i	$\forall\ i \in \phi$	
Maximum Functional Distribution	$\eta_{max,i}$	$\forall\ i \in \phi \mid \psi_i$	
Maintenance Crew Time	t_i	$\forall\ i \in \phi$	CM-h
Manufacturability	$\hat{\xi}_i$	$\forall\ i \in \phi$	
QPA	$\hat{q}_i$	$\forall\ i \in \phi$	
Baseline Unit Mass	m_i	$\forall\ i \in \phi \mid \theta_i$	kg
Life Limit	ℓ_i	$\forall\ i \in \phi \mid \theta_i$	h
Duty Cycle	d_i	$\forall\ i \in \phi \mid \theta_i$	
Mean Failure Rate	$\bar{\lambda}_i$	$\forall\ i \in \phi \mid \theta_i$	h^{-1}
Failure Rate Error Factor	ε_i	$\forall\ i \in \phi \mid \theta_i$	
Mean Crew Action Rate	$\bar{\lambda}_{Ci}$	$\forall\ i \in \phi \mid \theta_i$	h^{-1}
Crew Action Rate Error Factor	ε_{Ci}	$\forall\ i \in \phi \mid \theta_i$	
K-Factor	κ_i	$\forall\ i \in \phi \mid \theta_i$	
Mass Scaling Factor	γ_i	$\forall\ i \in \phi \mid \theta_i$	
Maximum Redundancy	$r_{max,i}$	$\forall\ i \in \phi \mid \theta_i$	
Potential Commonality Matrix	$\hat{C}$		
Mission Description	$\mathcal{M}$		
Endurance	τ		h
Required POS	POS_R		
Max. Maintenance Crew Time	t_{max}		CM-h
Modeling Parameters	$\mathcal{P}$		
Redundancy Mass Overhead Factor	γ_r		
Redundancy Effectiveness Factor	$\gamma_{\Lambda r}$		
ODM Mass Overhead Factor	γ_{odm}		
ODM Failure / Action Rate Factor	$\gamma_{\Lambda odm}$		
ODM System Mass	m_{odm}		kg
Crew Time Discretization	Δt		CM-h
ODM Feedstock Discretization	Δm		kg
Maximum Inversion Error	ϵ_{max}		

System Description

A system description $\mathcal{S} = \{\phi,\ \hat{C}\}$ describes the physical system that will be supported during the mission – that is, the systems and subsystems that ORUs will be defined for, that spares and maintenance items will be provided for, and that maintenance crew time will be spent on. The two elements of the system description are the item list ϕ and potential commonality matrix $\hat{C}$.

The item list ϕ defines and describes the structure and characteristics of the physical system itself. Each item $i \in \phi$ represents a system, subsystem, assembly, or component that could potentially be a level at which maintenance is implemented, if that item is selected as a replaceable item. By convention, items are numbered using sequential integers starting at 1. These items are arranged in a tree, called an indenture structure [12], that describes physical hierarchical relationships from the entire system (i.e. the root node of the tree) down to the lowest feasible level of maintenance, represented by the components (i.e. the leaf nodes of the tree). For example, the root node could represent an aircraft, with child nodes representing the fuselage, wings, and engines, each of which has children representing its constituent assemblies and items. The engine may be further decomposed into a compressor, turbine, and fuel system, each of which may be further decomposed into their constituent elements, and so on, down to the lowest level considered in the analysis (which depends on the desired level of fidelity and computational complexity).

Each item in the indenture structure has a set of supportability characteristics associated with it, described below. The indenture structure – defined by the parent-child relationships between items – is used along with the supportability characteristics of the items within and assignments to the supportability strategy variables to define ORUs for the system and characterize supportability as a function of those decisions. It is also used to define constraints on level of maintenance decision variables, as described in Section 3.2.4. It is important to note that this indenture structure is a structural decomposition of the system, not a functional decomposition – that is, it describes physical hierarchical relationships between system elements, not functional

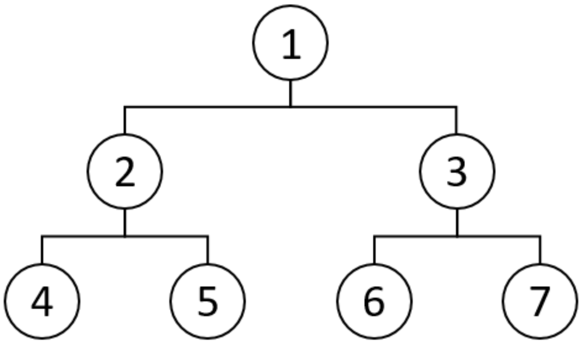

Figure 3.2: Example indenture structure

relationships.

The indenture structure itself is defined by specifying the parent p_i for each item in ϕ. Each item has exactly one parent, with the exception of the item representing the highest level of maintenance, which is the root node and by convention has $p_i = 0$ – an index pointing to no item in ϕ. This characteristic is used to define constraints related to to the root node in Section 3.2.4. The children j of a particular item i are defined as those items for which $p_j = i$. An item's ancestors are the set of items identified by recursively following parent relationships up the indenture structure, and an item's descendants are the set of items defined by recursively following child relationships down the indenture structure.

The component indicator θ_i is a binary variable indicating whether item i is a component – that is, whether it is an item at the lowest level of the indenture structure which has no children. Components could also be identified as the components that are not the parent for any other item:

$$\theta_i \iff \nexists\, j \in \phi \mid p_j = i \qquad \forall\, i \in \phi \tag{3.1}$$

However, these items are identified explicitly for convenience, since items that are components must have more characteristics defined for them than items that are not components.

An example indenture structure is shown in Figure 3.2. In this example, item 1 is the root node, with children 2 and 3. In terms of the item list, this means that

$p_1 = 0$ and $p_2 = p_3 = 1$. Item 2's children are 4 and 5, and item 3's children are 6 and 7. Since they are the lowest-level items in this indenture structure, items 4 through 7 are components. As noted earlier, the indenture structure indicates potential levels of maintenance within the system, and is therefore a physical hierarchy of the system itself. For example, items 4 and 5 (which are components), could combine into a higher-level assembly represented by item 2.

Components are the lowest-level unit; all other items are constructed via combinations of components, according to the parent-child relationships defined in the indenture structure. Critically, each level in the indenture structure represents a complete decomposition. That is, every item in the indenture structure must be fully decomposed into its children, and the aggregation of the characteristics of those children fully captures the characteristics of their parent item, with three exceptions, which are defined directly for non-component items (described below). For example, the mass of an item is equivalent to the sum of the masses of its children. This completeness of the decompositions in the indenture structure allows the supportability characteristics of higher-level items to be defined based on their children, which allows the definition of a wide range of levels of maintenance during the tradespace exploration and optimization process. Rules for combining lower-level characteristics into higher-level characteristics are used to define and characterize the ORUs used in supportability analysis, as described in Section 4.1.

The subsystem indicator ψ_i is a binary parameter indicating whether or not item i is a subsystem. In this context, a subsystem is defined as a node in the indenture structure that has relatively well-encapsulated functionality and represents a good level at which to apply distributed functionality, as identified by expert judgment. For example, in the context of spacecraft ECLSS, subsystems would be the groups of items responsible for water processing, carbon dioxide removal, and so on. Each subsystem (i.e. $i \in \phi \mid \psi_i = 1$) also has a maximum functional distribution parameter $\eta_{max,i}$ associated with it. This parameter indicates the maximum number of modules that the subsystem can be distributed across.

As noted earlier, there are three item characteristics that are defined for each item up front, rather than being generated by combining the characteristics of that item's children: the item's maintenance crew time t_i, manufacturability $\hat{\xi}_i$, and Quantity Per Application (QPA) $\hat{q}_i$. These characteristics depend upon more than simply the characteristics of the components within an item, and therefore they must be defined independently for each item rather than calculated based on the components that it contains.

Maintenance crew time t_i is the amount of time, in crewmember-hours (CM-h), required to perform a maintenance action on item i. A maintenance action is defined as any crew action related to that item, including inspection, trouble-shooting, and cleaning as well as R&R. This parameter is dependent upon a number of factors related to the maintainability of a specific assembly, and these factors are not fully captured by the children of that item.

Manufacturability $\hat{\xi}_i$ is a binary variable for each item indicating whether or not that item has the potential to be manufactured on demand; this acts as a constraint on decisions to use ODM to cover spares demands for particular ORUs (see Section 3.2.4). As with maintenance crew time, the ability to manufacture an item is dependent upon the characteristics of that item itself, and is not fully captured by the characteristics of its constituent components. For example, even if all of the children of a particular item could be manufactured on demand, it may be that the item itself (i.e. the assembly incorporating all of those children) is too large for the available ISM system. It is therefore true that all of the components of that item could be manufactured on demand, but in order to do so the item would have to be maintained at a lower level.

QPA $\hat{q}_i$ describes the number of instances of item i at that location in the indenture structure. This is the number of instances of this item within the particular assembly defined by the item's parent. QPA is separate from the quantity q_i, which represents the number of instances of this item within any given system or assembly, and is a function of the item's QPA and the QPA of all of its ancestors within that assembly. For example, if $\hat{q}_i = 2$, $p_i = j$, and $\hat{q}_j = 2$, then there exist two instances of item j, each of which contains two instances of item i, and therefore the quantity of item i is $q_i = \hat{q}_i\hat{q}_j = 4$. The calculation of quantities based on QPA is described in Section 4.1.

Note that in the indenture structure described by the parent-child relationships p, each item is considered unique. Potential commonality between items (encoded using the potential commonality matrix $\hat{C}$) is used to explore the potential for the same item to appear at multiple locations within the system.

All other characteristics, described below, need only be defined for components (i.e. $i \in \phi \mid \theta_i$). A component's baseline unit mass m_i is the mass of a single instance of that component, which is the mass of an installed unit as well as the mass of a spare. This is called the baseline mass because it may be scaled to account for a distribution of that component's functionality across multiple modules (described in Section 3.2.2). The life limit ℓ_i is the amount of time that a component can operate before a scheduled maintenance action is required. Duty cycle d_i indicates the portion of mission time that a component is operational – that is, the number of operational hours that have been accumulated by a given component is equal to the amount of time that has passed times the duty cycle for that component. Time-based supportability characteristics – that is, life limit ℓ_i, mean failure rate $\bar{\lambda}_i$, and mean crew action rate $\bar{\lambda}_{Ci}$ – are defined with respect to operational time, not clock time.

There are two stochastic processes associated with each component. The first process is driven by random failures, defined as instances where a component has failed and must be replaced. These are characterized by an uncertain failure rate Λ_i, which is lognormally-distributed and parameterized by the mean failure rate $\bar{\lambda}_i$ and error factor ε_i. These random failures drive corrective maintenance demand. The second process is crew actions, which are defined as instances where crew time is expended performing some maintenance-related action for an item, whether or not the item is replaced. That is, crew actions include ORU replacement after random failures as well as actions that do not include replacement of the ORU, such as inspection, troubleshooting, or cleaning. Crew actions are characterized by an uncertain crew action rate Λ_{Ci}, which is also lognormally-distributed and parameterized by a mean crew action rate $\bar{\lambda}_{Ci}$ and crew action rate error factor ε_{Ci}. Since crew actions drive maintenance crew time requirements, the associated rates are indicated using a subscript C. The maintenance demand model presented in Section 4.2.1 describes the process used to derive distributions of the number of random failures and crew actions required for each ORU as a function of their associated rates.

The use of a separate stochastic process to drive crew time demands is another contribution of this dissertation. Previous approaches to evaluating maintenance crew time use the same stochastic process to evaluate demand for both spares and crew time, thereby underestimating total crew time requirements by excluding crew time spent on tasks that do not involve ORU replacement. By including a separate crew action rate Λ_C, this research enables more complete modeling of the amount of crew time required for maintenance.

A component's K-factor κ_i is a multiplier on failure rates and crew action rates that accounts for "otherwise unplanned events in equipment maintenance," meaning induced failures, or events that are not accounted for in the failure or crew action rate itself [19, 122]. Note that the K-factor for a particular component may be different for different applications. For example, operations in a particular environment – such as an environment with higher radiation, or a dusty planetary surface environment – may result in a larger number of induced failures than a different environment with less strenuous environmental conditions. The appropriate K-factor for a particular environment is based on past experience, and for missions to new environments (such as interplanetary space or Mars), will likely be strongly dependent on expert opinion. For the purposes of this research, it is assumed that the K-factors provided in the item list ϕ are applicable to the mission being examined. In addition, it is assumed that the K-factors are applicable to both the failure rate and the crew action rate, since both processes would be affected by external factors.

The mass scaling factor γ_i is a number between 0 and 1 used to scale components in response to changes in functional distribution. This research follows the system scaling approach used by the ALSSAT, as described by Yeh et al. [50], in which the scaled mass m'_i of a particular item is equal to its initial mass m_i times a factor derived from its scaling factor and the change in the amount of functionality that it is responsible for,

$$m_i' = m_i \left(1 - \gamma_i + \frac{\zeta_i'}{\zeta_i}\gamma_i\right) \tag{3.2}$$

where ζ_i is the amount of functionality (e.g. processing rate) that the component is responsible for initially, and ζ_i' is the amount that it is responsible for after scaling. In this research, components are scaled in response to having their functionality distributed across multiple modules, and as a result the scaling factor $\frac{\zeta_i'}{\zeta_i}$ is a fraction based on the number of modules that the component's functionality is distributed across, as described in Section 4.1. This approach to component scaling is similar to that used in the chemical industry [50]. γ_i can be thought of as the proportional element of scaling, since it is a multiplier on the ratio of the final to initial functionality, while $1 - \gamma_i$ is a constant element of scaling that is not dependent upon the change in functionality. Therefore, a component whose mass is completely dependent upon the amount of functionality it is responsible for will have $\gamma_i = 1$, a component with some fixed mass and some mass that scales proportionally will have $0 < \gamma_i < 1$, and a component whose mass is not dependent upon functionality at all will have $\gamma_i = 0$. These cases correspond to scaling factor formulas "1," "2," and "3," respectively, as described by Yeh et al. [50].

The maximum redundancy $r_{max,i}$ for each component specifies the maximum number of redundant copies for component i. This parameter acts as a bound on the level of redundancy for each component. For the purposes of this research, these redundant copies are assumed to be offline and identical; that is, they have the same supportability characteristics as the installed component, and they do not activate until needed. In addition, redundancy is applied at the component level, within ORUs. The presence of redundant components within an ORU allows that ORU to remain in operation after a component within it fails, thus reducing the number of ORU replacements required at the expense of having a more massive ORU. This type of redundancy is similar to that used by the hydrogen sensor ORU within the ISS OGA, which contains three redundant copies of the sensor itself (though in the case of the hydrogen sensor ORU all three redundant sensors are active at all times to allow the ORU to remain in service after one of the sensors fails. This redundancy was utilized on orbit, for example, after one of the sensors failed in late 2017 but the ORU was able to remain operational using the remaining two sensors until mid-2018 [123].

Note that this is only one type of redundancy that could be implemented. Another approach would be to implement redundancy at a level above ORUs – that is, there could be redundant instances of a given ORU that continue functioning while a failed unit is being removed and replaced. This type of redundancy would impact system downtime and the time-criticality of maintenance activities, but it does not change the number of ORU replacements, since it does not affect failure rates. From a logistics perspective, redundancy above the ORU level is equivalent to providing additional spares for a given ORU or set of ORUs with additional constrains specifying that the same number of spares are provided for a given group of items (i.e. the items contained within the redundant system). Therefore, since the primary value of this type of redundancy is captured in metrics that are beyond the scope of the current research (i.e. system downtime and time-pressure on maintenance actions), redundancy above the ORU level is left to future work (described in Chapter 9).

For notation purposes, all of the information described above is included in the item list ϕ. Characteristics of specific items are identified via subscript; for example the mass of a particular item i in θ is denoted m_i. In addition, a vector representing a particular parameter for all items is represented by an unsubscripted variable; for example, the vector of mass of every item is given by m. Note that some characteristics will not be defined for items that are not components; the corresponding entry in the vector can be any value, since it will not be used in subsequent calculations. When a calculation or algorithm is passed ϕ, this is equivalent to passing the full set of characteristics $\{p,\ \theta,\ \psi,\ t,\ \hat{\xi},\ \hat{q},\ m,\ \ell,\ d,\ \bar{\lambda},\ \epsilon,\ \bar{\lambda}_C,\ \epsilon_C,\ \kappa,\ \gamma,\ r_{max}\}$.

The final component of the system description is the potential commonality matrix $\hat{C}$, which has binary entries $\hat{C}_{ij}$ indicating whether or not items i and j could potentially share common spares. Similar to item manufacturability, the entries in the potential commonality matrix serve as a constraint on commonality decision variables, as described in Section 3.2.4.

Mission Description

A mission description $\mathcal{M} = \{\tau,\ POS_R,\ t_{max}\}$ defines the context in which the system described by $\mathcal{S}$ will operate. The three elements of a mission description are the mission endurance τ, required POS POS_R, and maximum allowable maintenance crew time t_{max}.

Mission endurance τ is defined as the time between resupply opportunities [42], or the maximum amount of time that the system must support the crew without receiving additional supplies. For sortie missions, this is typically the time between launch and landing; for space stations or outposts, it is the time between resupply vehicle arrival. Mission endurance is the planning time horizon for supportability, and all risk and maintenance resource requirements will be evaluated with respect to this time period. (See Section 1.3 for a more detailed discussion of mission endurance and its relationship to supportability.)

POS_R is the minimum acceptable POS for the mission, or the minimum acceptable probability that the maintenance resources that are available – including spares and crew time – are sufficient to cover all maintenance demands for the period of time defined by the endurance τ [12, 13]. For crew time, the amount of time available is the maximum allowable maintenance crew time t_{max}, described below, and the probability of sufficient crew time is calculated with respect to this amount. From a mass perspective, spare parts are allocated to achieve this POS target, taking into account the probability of sufficient crew time, while minimizing spares mass.

Finally, the maximum allowable maintenance crew time t_{max} is an upper bound on the amount of crew time available for maintenance activities. This number is based on the amount of time left over after crew time liens for activities such as sleep, exercise, and other required operations are removed [7]. When POS is assessed for a given system, the crew time contribution to POS is assessed as the probability that the amount of maintenance crew time required does not exceed t_{max}.

Modeling Parameters

The system and mission description above are characteristics of the real-world system and mission that are required to define the supportability strategy optimization problem. A set of modeling parameters $\mathcal{P} = \{\gamma_r,\ \gamma_{\Lambda r},\ \gamma_{odm},\ \gamma_{\Lambda odm},\ m_{odm},\ \Delta t,\ \Delta m,\ \epsilon_{max}\}$ are also required for portions of the supportability model (described in Sections 4.1 4.2), both by providing adjustment factors to account for changes in component and system characteristics resulting from redundancy and/or ODM as well as providing parameters for numerical modeling methods used to assess crew time and ODM feedstock requirements.

A set of multiplicative factors are used to adjust the mass, failure rate, and crew action rate of items as a function of decisions relating to redundancy and ODM. The redundancy mass overhead factor γ_r indicates the amount of extra mass that must be added when a redundant component is added, as a fraction of the mass of the redundant component. Redundancy cannot simply be inserted into a system for free. Additional infrastructure (e.g. electrical connections, fittings, fluid lines, switches, etc.) is required to allow the redundant component to interface with the system and switch on when needed, and this infrastructure requires additional mass. For example, a redundancy mass overhead factor of 0.05 indicates that, for each redundant instance of a component added to an ORU, 5% of that component's mass must be added as overhead. This research assumes that $\gamma_r \geq 0$.

In addition, redundancy may not provide perfect failure rate reduction. In particular, the additional infrastructure described above is a new potential source of failure which must be accounted for. These impacts are captured in the redundancy effectiveness factor $\gamma_{\Lambda r}$, a parameter between 0 and 1 (inclusive) which indicates the portion of the component failure rate that is affected by each redundant instance. Perfect redundancy – i.e. the case where the presence of k redundant units effectively divides the failure rate by $1+k$ – is represented by a redundancy effectiveness factor of 1. A redundancy effectiveness factor of 0.95, for example, indicates that the presence of k redundant units divides the failure rate by $1+0.95k$ instead. This can be thought of as a 5% failure rate "overhead" associated with redundancy. This factor applies to both failure rates and crew action rates, and the overall approach to redundancy modeling used for this research is described in greater detail in Section 4.1.

The use of ODM to manufacture spares for certain items may impact the mass, failure rate, and crew action rate of those items, due to differences between manufacturing processes used in space and those used on Earth, as well as potential differences in item design resulting from the use of ODM. These impacts are captured in the ODM mass overhead factor γ_{odm} and ODM failure/action rate factor $\gamma_{\Lambda odm}$. The former indicates the portion of ORU mass that must be added when ODM is used for that ORU, while the latter indicates the portion of failure rate and crew action rate overhead. Both of these parameters are assumed to be greater than 0, meaning that ODM components are more massive and less reliable than they would be if they were manufactured using traditional processes. While it may be possible in the future to manufacture less massive components in the microgravity environment of space due to the removal of gravitational loading constraints, in the near term the quality of ODM components is likely limited by the manufacturing process itself. ODM also requires the presence of a manufacturing system, which incurs additional mass. The mass of this system, m_{odm}, is added to the overall system mass if any components use ODM in order to account for the mass cost of the infrastructure associated with ODM.

The ranges provided for the parameters listed above are primarily specified in order to ensure that the procedures used for identifying the bounds on mass and crew time for a partial assignment to the strategy variables, described in Section 4.3.2, are valid. The supportability model itself can be applied using values beyond those ranges. However, use of parameters outside the given range will require adjustment to the heuristics listed in Section 4.3.2 and may reduce the efficiency of the optimization process.

Evaluation of the distribution of maintenance crew time required, as well as of the amount of feedstock required for ODM (described in Sections 4.2.2 and 4.2.3, respectively), make use of an approximation that performs convolution using the characteristic functions of discretized probability distributions. The parameters Δt and Δm are the discretization step size used for crew time and mass, respectively, and the maximum inversion error ϵ_{max} is a parameter used to balance accuracy and computational cost in the numerical characteristic function inversion algorithm.

Derived Characteristics

The above inputs fully describe the data required for the supportability strategy optimization problem. Some of these inputs can be used to derive representations of system characteristics that are useful for later calculations. These derived characteristics depend only on the inputs described in this section, and can therefore be calculated prior to any optimization activities in order to reduce computational overhead during optimization. Specifically, the derived system characteristics used in this analysis are the adjacency matrix A, descendant matrix D, depth vector $\hat{d}$, total quantity vector q, and quantity matrix Q. Each of these are described below.

The adjacency matrix A is a matrix representation of the indenture structure as a directed graph, with directionality defined as flowing from parent to child, defined by the parent list p. Entries $A_{ij} = 1$ if j is a child of i (i.e. $p_j = i$), and 0 otherwise. For example, the adjacency matrix associated with the indenture structure presented in Figure 3.2 is:

$$A = \begin{bmatrix} 0 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \tag{3.3}$$

Note that this adjacency matrix is not a representation of any functional connections between items, and is not necessarily related to any other adjacency matrices that may be used in system architecting or design to indicate connections between subsystems.

The indenture structure is derived from the formal structure of the system, and is not a functional decomposition. Therefore, while formal relationships driven by functional adjacency may still appear in this adjacency matrix (e.g. assemblies which participate in similar functions may be grouped into subsystems), this adjacency matrix is a description of the form of the system, not its function. It is simply a convenient representation of the indenture structure in matrix form, encoding the parent-child relationships specified by the parent list p in a way that facilitates the use of matrix operations to define system supportability characteristics (see Section 4.1).

The descendant matrix D indicates which items are descendants of other items $D_{ij} = 1$ implies that j is a descendant of i, otherwise it is 0. D can be defined by iteratively "walking" down the indenture structure using the adjacency matrix, using equation 3.4.

$$D = \sum_{k=1}^{k_{max}} A^k \tag{3.4}$$

The entries in A^k (which is the adjacency matrix raised to the power k) give the number of walks of length k between i and j – that is, the number of ordered sequences of k nodes in the graph representation of the indenture structure that connect i to at j [124]. Iterative summation identifies nodes connected by all paths up to length k_{max}, which is the maximum path length in the graph. In this application, k_{max} is the depth of the indenture structure. (In practice, the summation is carried out as long as A^k is non-empty, meaning that there are still descendants to add.) Since the indenture structure is a tree, it is an acyclic, arborescent (i.e. tree-like) graph, meaning that there exists at most one path of length k from i to j for any k; therefore the entries in D are either 0 or 1, with the former meaning that j is not a descendant of i and the latter meaning that it is. The descendants matrix associated with the indenture structure in Figure 3.2 is:

$$D = \begin{bmatrix} 0 & 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \tag{3.5}$$

The depth vector $\hat{d}$ is calculated in a similar manner to the descendants matrix, and has entries $\hat{d}_i$ indicating the depth of item i in the indenture structure. Root nodes have depth 1, while their children have depth 2, and so on. The depth vector is calculated by initializing a row vector y with entries $y_i = 1$ if i is a root node, and $y_i = 0$ otherwise. $\hat{d}$ is then calculated as

$$\hat{d} = y + \sum_{k=1}^{k_{max}} yA^k \circ (k+1) \tag{3.6}$$

For the example in Figure 3.2, the depth vector is:

$$\hat{d} = \begin{bmatrix} 1 & 2 & 2 & 3 & 3 & 3 & 3 \end{bmatrix} \tag{3.7}$$

The quantity vector q and quantity matrix Q are two representations of the total number of a particular item in the system, based on the input QPAs $\hat{q}$. The entries q_i give the total quantity of item i in the system, while entries Q_{ij} give the quantity of item j within a given instance of item i. The total quantity of a given item is equal to its QPA multiplied by the QPA of all of its ancestors (i.e. the items above it in the indenture structure). Typical matrix operations represent linear combinations of the components of vectors; however, multiplication can also be accomplished by using the fact that a logarithm maps multiplication to addition for positive numbers. Specifically, since $\prod_i x_i = e^{\sum_i \ln x_i}$ for $x_i > 0$, and all QPAs are positive, the total quantity vector q can be calculated as

$$q = \exp\left[\left(I + D^T\right)\ln\left[\hat{q}\right]\right] \tag{3.8}$$

Here I is the identity matrix, D^T is the transpose of the descendant matrix (which indicates ancestors rather than descendants; entries D^T_{ij} are equal to 1 if j is an ancestor of i, and 0 otherwise), and $\hat{q}$ is the QPA vector. The functions $\exp[\cdot]$ and $\ln[\cdot]$ are the element-wise exponential and natural logarithm functions. For a given matrix M the entries of $\exp[M]$ are $e^{M_{ij}}$, and the entries of $\ln[M]$ are $\ln M_{ij}$.

The quantity of an item j within any other item i is equal to $\frac{q_j}{q_i}$ if j is a descendant of i or $j = i$ and 0 otherwise. The quantity matrix Q is therefore calculated as

$$Q = \left(q^{\circ -1}\right) q^T \circ (I + D) \tag{3.9}$$

where $^{\circ -1}$ is the element-wise inverse (i.e. $q^{\circ -1}$ is the vector with entries q_i^{-1}), ^{T} is the matrix transpose, and $\circ$ is the Hadamard matrix product (i.e. element-wise matrix multiplication; the Hadamard product is defined such that the entries of a matrix $A = B \circ C$ are $A_{ij} = B_{ij}C_{ij}$). The matrix formed by $\left(q^{\circ -1}\right) q^T$ gives the total number of item j divided by the total number of item j, regardless of whether i is a descendant of j or $i = j$. Element-wise multiplication by $I + D$ then sets any entries for which j is not a descendant of i to zero.

3.2.2 Decision Variables

The decision variables for the supportability strategy optimization problem capture decisions related to the five key supportability strategies described in Section 1.2: level of maintenance, commonality, ODM, distributed functionality, and redundancy. An additional set of auxiliary variables, called the maintenance coverage variables, are included to enforce constraints related to level of maintenance decisions Another auxiliary variable, ξ_s is used to indicate the presence or absence of an ODM system (for the purposes of computing system mass). Though part of the decision variable set,

Table 3.2: Summary of supportability strategy decision variables.

Name	Symbol	Set of Variables	Domain
Level of Maintenance	u	$u_i \; \forall \; i \in \phi$	$\{0,\ 1\}$
Maintenance Coverage	ν	$\nu_i \; \forall \; i \in \phi$	$\{0,\ 1\}$
On-Demand Manufacturing	ξ	$\xi_i \; \forall \; i \in \phi$	$\{0,\ 1\}$
On-Demand Manufacturing System	ξ_s		$\{0,\ 1\}$
Commonality	C	$C_{ij} \; \forall \; i,\ j \in \phi$	$\{0,\ 1\}$
Redundancy	r	$r_i \; \forall \; i \in \phi$	$\{0,\ \ldots,\ r_{max,i}\}$
Distributed Functionality	η	$\eta_i \; \forall \; i \in \phi$	$\{1,\ \ldots,\ \eta_{max,i}\}$

these variables are simply used as part of the COP model and are not a supportability strategy decision *per se*. All of these decision variables are described in the following sections, and a summary is presented in Table 3.2. For notation purposes, the set of supportability strategy decision variables is represented as $\mathcal{X} = \{u,\ \nu,\ \xi,\ \xi_s,\ C,\ r,\ \eta\}$.

Level of Maintenance

level of maintenance is encoded using a set of binary variables $u = \{u_i \in \{0,\ 1\} \; \forall \; i \in \phi\}$ indicating whether or not item i is a replaceable item.[a] In this context, the term "replaceable item" refers to an item that maintenance actions can be applied to. That is, a replaceable item represents a set of one or more components that have been combined into an assembly that can be removed and replaced for maintenance, following the system hierarchy laid out in the indenture structure. Note that a replaceable item is not necessarily the same as an ORU, since these items may be made common with other replaceable items to form a single ORUs. The rules for defining ORUs based on the system description and an assignment to the supportability strategy variables are described in Section 4.1. Effectively, the variables u_i are indicators of the level in the indenture structure at which maintenance actions are implemented.

In addition, a set of auxiliary variables $\nu = \{\nu_i \in \{0,\ 1\} \; \forall \; i \in \phi\}$, called maintenance coverage variables, are used to indicate whether the descendants of a particular item have been covered by some lower level of maintenance. These auxiliary variables

[a] This research follows the convention that 1 = True and 0 = False for binary variables.

are fully defined by an assignment to the level of maintenance variables u (see equations 3.14 and 3.15), and are used to enforce the constraint that an item is not a replaceable item if all of its children are already replaceable items or are covered by a lower level of maintenance.

On-Demand Manufacturing

ODM decisions are encoded as a set of binary variables $\xi = \{\xi_i \in \{0,\ 1\}\ \forall\ i \in \phi\}$ indicating whether or not spares for item i are manufactured on demand. These variables directly correspond to the manufacturability $\hat{\xi}_i$ of each item, which acts as a constraint on ODM decisions. If an ORU uses ODM, then the maintenance mass demands for that ORU, along with those of other ORUs that use ODM, are covered by a common pool of manufacturing raw materials, called feedstock. If an ORU does not use ODM, then it uses discrete spare parts that are specific to that ORU.

If ODM is used for any ORUs, then the spacecraft must include an ODM system and pay the associated mass penalty. A binary auxiliary variable ξ_s is used to indicate whether or not an ODM system is present. As with the maintenance coverage variables ν, ξ_s is fully defined by an assignment to ξ, and is used to as an input to system mass calculations rather than as a decision in and of itself.

Commonality

Commonality is encoded by a matrix of binary variables $C = \{C_{ij} \in \{0,\ 1\}\ \forall\ i,\ j \in \phi\}$ indicating whether or not items i and j share common spares. These variables directly correspond to the entries in the potential commonality matrix $\hat{C}$, which act as a constraint on commonality decisions. By convention, if an item is selected as a replaceable item then it is considered common with itself. Sets of replaceable items that are common with each other are combined to form ORU; if a replaceable item is not common with any other replaceable item, then it is its own ORUs.

Redundancy

Redundancy is applied at the component level in order to increase the number of component failures that would have to occur before an ORU requires replacement. This decision is represented by the set of non-negative integer variables $r = \{r_i \in \{0, \ldots, r_{max,i}\} \; \forall \; i \in \phi\}$ indicating the number of redundant instances provided. The domain is restricted from above by $r_{max,i}$, the maximum level of redundancy for that component. The level of redundancy provided for non-component items is constrained to 0 (as described in Section 3.2.4), since this research only considers commonality at the component level; however, these values are included in the decision variable vector in order to enable matrix-vector operations that define the characteristics of the ORU list as a function of decision variable assignments (described in Section 4.1). For the purposes of this research, redundant components are assumed to be offline, identical cold spares. That is, their design and supportability characteristics are identical to the component that they are providing redundancy for, and they do not activate and begin accumulating operating hours until they are needed.

Distributed Functionality

Distributed functionality applies to subsystems, and is encoded by the set of positive integer variables $\eta = \{\eta_i \in \{1, \ldots, \eta_{max,i}\} \; \forall \; i \in \phi\}$ indicating the number of modules that the subsystem's functionality is distributed across. As with the redundancy decision variables, these are constrained to 1 for items that are not subsystems; they are included in the decision variable vector to simplify the creation and characterization of the ORU list. For the purposes of this research, these modules are considered to be identical copies of the system, each of which has a mass scaled to handle a portion of the overall function using the technique described in equation 3.2. All other characteristics (failure rates, crew action rates, life limits, etc.) are assumed to be unaffected by scaling.

Table 3.3: Supportability strategy optimization problem objectives.

Name	Symbol	Units
Total Mass	m_{tot}	kg
Expected Maintenance Crew Time	$\mathrm{E}[T]$	CM-h

3.2.3 Objectives

The objectives of the supportability strategy optimization problem are to minimize both total mass and the expected maintenance crew time. These objectives are defined in the following sections and summarized in Table 3.3. Both of these objectives are evaluated by a supportability model that takes an assignment to the decision variables $\mathcal{X}$ along with a subsystem $\mathcal{S}$, mission $\mathcal{M}$, and set of parameters $\mathcal{P}$.

$$(m_{tot},\ \mathrm{E}[T]) = f\left(\mathcal{X};\ \mathcal{S},\ \mathcal{M},\ \mathcal{P}\right) \tag{3.10}$$

This evaluation occurs in two steps. First, an ORU list is defined based on the inputs described in Section 3.2.1 and an assignment to the decision variables described in Section 3.2.2. This ORU list is a compact representation of the relevant supportability characteristics of the system, as a function of the supportability strategy that is used. Then, the mass and crew time associated with this list are evaluated. The model is described in depth in Sections 4.1 and 4.2.

Mass

Total mass m_{tot} is defined as the sum of the installed system mass m_S, the mass of maintenance items and ODM feedstock required for preventative maintenance m_{PM}, and the mass of spares and ODM feedstock required for corrective maintenance m_{CM}.

$$m_{tot} = m_S + m_{PM} + m_{CM} \tag{3.11}$$

System mass m_S is defined as the mass of the installed system itself (i.e. not logistics mass), including any redundant modules. System mass is included in the analysis because some supportability decisions – particularly ODM, redundancy and distributed functionality – can potentially reduce supportability-related logistics at the expense of increased system mass. Preventative maintenance mass m_{PM} is the logistics mass associated with maintenance items, which are spares provided to cover scheduled maintenance activities. Corrective maintenance mass m_{CM} is the logistics mass associated with spare parts, which are provided to cover maintenance demands resulting from random failures.

Crew Time

$\mathrm{E}[T]$ is the expected total amount of crew time required for maintenance, including both preventative and corrective actions, and time spent removing and replacing failed ORUs and time spent on non-replacement activities such as inspection and troubleshooting. The total maintenance crew time T is defined as the sum of crew time spent on preventative maintenance activities t_{PM} and crew time spent on corrective maintenance T_{CM}. The former is deterministic, based on scheduled maintenance activities, while the latter is a random variable that depends on the crew action rates for all ORUs. Therefore, total expected crew time is the sum of time spent on preventative maintenance and the expected time spent on corrective maintenance.

$$T = t_{PM} + T_{CM} \tag{3.12}$$

$$\mathrm{E}[T] = t_{PM} + \mathrm{E}[T_{CM}] \tag{3.13}$$

Note that while expected crew time is selected as the particular metric used to assess crew time in this research, the methodology described here is not limited to the expected value. As shown in Section 4.2.2, the crew time model calculates the full distribution of T_{CM}. Therefore any descriptive statistic related to crew time – such as median, mode, or the amount of crew time associated with any percentile – could also be used as an objective value in the place of (or in addition to) the expected value.

3.2.4 Constraints

Supportability strategy optimization involves the consideration of a wide range of decisions that are coupled with each other and with the characteristics of the system being examined. Various system characteristics (e.g. indenture structure, maximum values for redundancy and distributed functionality) as well as decisions for specific supportability elements place constraints on valid and/or feasible assignments to other supportability decisions. As such, it is natural to encode the problem as a COP.

There are two types of constraints in the supportability strategy optimization problem: constraints on valid assignments to the supportability strategy variables, and constraints related to feasible values of the objective functions (i.e. mass and crew time). The former, are derived from the structure of the system itself and the relationship between the various supportability strategy decision variables. The latter, on the other hand, relate to acceptable values of the objective function.

The overall COP is summarized in Section 3.3. In total, there are 21 sets of constraints, all of which are listed in equation 3.36 on page 148. For this research, the COP defined here is encoded using MiniZinc, a solver-agnostic constraint programming modeling language [117, 125]. The PyMzn module [118] in Python is used to generate and interact with the MiniZinc model, which is passed to Gecode [121] and OpSat [119, 120] to enumerate candidate strategies and search for optimal solutions. Gecode is used to initialize a candidate Pareto frontier for the multiobjective branch and bound search carried out by OpSat. Candidate strategies provided by the solver (including full and partial assignments to the decision variables) are evaluated using a Python supportability evaluation module created as part of this research. The methodology used for evaluating supportability strategies is described in Sections 4.1 and 4.2.

Constraints on Valid Decision Variable Assignments

The first set of constraints deals with level of maintenance decisions u, and use the auxiliary maintenance coverage variables ν to ensure that the indenture structure is taken into account appropriately when selecting replaceable units. The constraints described in equations 3.14 and 3.15 define the value of ν_i for all items as a function of assignments to u. This ensures that these auxiliary variables can then be used to enforce the constraint shown in equation 3.16, which states that an item is not a replaceable item if it is already covered by a lower level of maintenance.

To begin, the value of the auxiliary maintenance coverage variables for components is fixed to 0, since those items have no children and therefore by definition cannot be covered by a lower level of maintenance.

$$\theta_i \implies \neg \nu_i \qquad \forall\ i \in \phi \tag{3.14}$$

Any item i that is not a component does have children, and ν_i is defined as a function of the values of ν_j and u_j for that item's children j. Specifically, an item is covered by a lower level of maintenance if and only if all children of that item are either replaceable items or are covered by a lower level of maintenance.

$$\bigwedge_{\substack{j \in \phi \\ p_j = i}} (u_j \vee \nu_j) \iff \nu_i \qquad \forall\ i \in \phi \mid \neg \theta_i \tag{3.15}$$

Constraints 3.14 and 3.15 fully define all maintenance coverage auxiliary variables ν as a function of the replaceable unit decisions u. The assignments to ν are used to ensure that if an item is already covered by a lower level of maintenance, it is not a replaceable unit.

$$\nu_i \implies \neg u_i \qquad \forall\ i \in \phi \tag{3.16}$$

Since ORUs are created by combining lower-level items into higher level items, this constraint prevents the case where a replaceable item is created where there are not lower-level items left to combine into it.

The purpose behind creating higher-level items is to simplify interfaces and (potentially) reduce maintenance crew time by aggregating components into assemblies.

Therefore, higher-level items should only be replaceable items if they contain at least two components. Any higher-level item that contains only one component is effectively only a different representation of that one component. Here it is important to expand on what it means for an item to "contain" a lower level item, once level of maintenance decisions are made. As described in Section 4.1.1, the set of components contained by a given replaceable item is defined as the set of components that are direct descendants of that replaceable item, meaning that there is no other replaceable item between the containing item and the components that it contains. Put another way, each component can be thought of as being contained in the nearest ancestor that is a replaceable item (or itself, if that component is a replaceable item).

Therefore, for a higher-level item i to be made a replaceable item, there must exist at least two components j and k ($j \neq k$) such that both j and k are descendants of i, neither j nor k are replaceable items, and there is no other item l that is a replaceable item, a descendant of i, and an ancestor of either j or k. The next constraint, one of the most complex ones in this model, ensures that this is the case.

$$u_i \implies \bigvee_{\substack{j,k \in \phi \\ \theta_j \wedge \theta_k \\ j \neq k}} \left(D_{ij} \wedge D_{ik} \wedge \neg \left(u_j \vee u_k \right) \right. \\ \left. \wedge \neg \bigvee_{\substack{l \in \phi \\ \neg \theta_l}} \left(u_l \wedge D_{il} \wedge \left(D_{lj} \vee D_{lk} \right) \right) \right) \qquad \forall\, i \in \phi \mid \neg \theta_i \qquad (3.17)$$

Note that in this constraint the entries D_{ij} of the descendants matrix D are treated as binary variables indicating whether or not j is a descendant of i.

To ensure that all items within the indenture structure are covered, equation 3.18 specifies that if an item has no parent and is not covered by a lower level of maintenance, then it must be a replaceable unit.

$$(p_i = 0) \wedge \neg \nu_i \implies u_i \qquad \forall\, i \in \phi \qquad (3.18)$$

This constraint uses the fact that items are assumed to be numbered using positive integers, and the condition $p_i = 0$ is only fulfilled by the root node of the indenture structure.

In many cases, one indenture structure is used to describe the entire system, and therefore there is only one root node. However, this constraint allows the methodology to be generalized to cases where there are multiple, independent indenture structures to be examined that are not derived from a single higher level of maintenance. This may be the case, for example, if a supportability analysis is to be carried out on the systems of both a habitat and a pressurized rover. Both elements would contain subsystems and assemblies that could be considered for various levels of maintenance, but it does not make sense from a maintenance perspective to consider the habitat and the pressurized rover to both be children of some higher level of maintenance, since they are two independent and separate objects. In general, an upper bound on the level of maintenance can be specified by setting $p_j = 0$ for any item j that is considered the highest level of maintenance.

Subsystems (i.e. $i \in \phi \mid \psi_i$) are by definition the only locations where distributed functionality is implemented. Therefore, η_i is constrained to 1 (i.e. all functionality contained in one module) for any item that is not a subsystem.

$$\eta_i = 1 \qquad \forall\ i \in \phi \mid \neg\psi_i \tag{3.19}$$

A particular subsystem may be an item covered by a lower level of maintenance (i.e. $\nu_i = 1$), it may be a replaceable item (i.e. $u_i = 1$), or it may be encapsulated within another replaceable item (i.e. $u_i = 0$ and $\nu_i = 0$). These replaceable items are later processed to create ORUs, using the procedure described in Section 4.1. In the first and second cases, the subsystem contains ORUs, and distributed functionality could be implemented to transform these monolithic ORUs or groups of ORUs into a larger number of smaller ORUs to reduce logistics mass, at the expense of increased crew time (see Section 1.2.4 for a more detailed description of this effect). However, it does not make sense to distribute the functionality of a subsystem that is entirely encapsulated within an ORU, since that would have only negative impacts. Specifically, the subsystem would be split into multiple modules, meaning that failures would occur more often, but since all modules are encapsulated within the same ORU there are no logistics benefits associated with scaling down the mass of any components within the ORU. When any failure occurs within the ORU, the entire ORU will be replaced, including all distributed modules within it. Therefore, equation 3.20 specifies that a subsystem's functionality is not distributed across more than one module if that subsystem is below the selected level of maintenance.

$$\neg\nu_i \wedge \neg u_i \implies \eta_i = 1 \qquad \forall\ i \in \phi \mid \psi_i \tag{3.20}$$

Put another way, for all items that are subsystems, if that item is not covered by a lower level of maintenance and is not itself a replaceable item, then it is encapsulated within an ORU, and therefore there should be no attempt to distribute functionality.

The next four constraints deal with commonality. First, the entries $\hat{C}_{ij}$ of the potential commonality matrix indicate whether it may be possible to make items i and j common with each other. These are direct constraints on the commonality decision variables. Specifically, if items i and j are not potentially common, then they cannot be common.

$$\neg\hat{C}_{ij} \implies \neg C_{ij} \qquad \forall\ i,\ j \in \phi \tag{3.21}$$

In addition, if either of items i and j (where $i \neq j$) are not a replaceable item, they cannot be made common, since they must both exist in order to be made common.

$$\neg u_i \vee \neg u_j \implies \neg C_{ij} \qquad \forall\ i,\ j \in \phi \mid i \neq j \tag{3.22}$$

Commonality is reflexive, symmetric, and transitive. This means that, by convention, any item that is a replaceable item is considered common with itself. Items that are not replaceable cannot be common with anything, and by convention (and for convenience in later calculations, see Section 4.1) their entry on the diagonal of the commonality matrix is set to 0. Also, if item i is common with item j, then item j is also common with item i. Similarly, if item i is common with item j and item j is common with item k, then item i is also common with item k. These three constraints are captured in equations 3.23 through 3.25.

$$C_{ii} \iff u_i \qquad \forall\, i \in \phi \tag{3.23}$$

$$C_{ij} \implies C_{ji} \qquad \forall\, i,\ j \in \phi \tag{3.24}$$

$$C_{ij} \wedge C_{jk} \implies C_{ik} \qquad \forall\, i,\ j,\ k \in \phi \tag{3.25}$$

The next five constraints deal with ODM decisions. First, if an item is not manufacturable or is not a replaceable item, then its spares cannot be manufactured on demand.

$$\neg\hat{\xi}_i \implies \neg\xi_i \qquad \forall\, i \in \phi \tag{3.26}$$

$$\neg u_i \implies \neg\xi_i \qquad \forall\, i \in \phi \tag{3.27}$$

These constraints are similar to those used for commonality. In addition, if items i and j are common (and $i \neq j$), then they do not use ODM. This is due to the fact that ODM is simply a less restricted form of commonality. When items use ODM, it no longer matters whether they were common with other items or not, since they are not included in the pool of items that use the common raw material feedstock as needed.

$$C_{ij} \implies \neg\xi_i \wedge \neg\xi_j \qquad \forall\, i,\ j \in \phi \mid i \neq j \tag{3.28}$$

ODM decisions also have the interesting property that at least two different items must use ODM in order for it to provide benefits. If only one item uses ODM, then there is no commonality of material between different items (since there are no other items to share common ODM feedstock), and as a result there is no logistics mass savings. Therefore, at least two different items must choose to use ODM.

$$\xi_i \implies \bigvee_{\substack{j \in \phi \\ j \neq i}} \xi_j \qquad \forall\ i \in \phi \tag{3.29}$$

Put another way, if an item uses ODM, then there must exist at least one other item that also uses ODM. In addition, if any item uses ODM, then an ODM system must be included; otherwise, no ODM system is included.

$$\xi_s \iff \bigvee_{j \in \phi} \xi_i \tag{3.30}$$

Any item that is not a component must have a redundancy of 0, since redundancy is implemented at the component level in this model.

$$r_i = 0 \qquad \forall\ i \in \phi \mid \neg\theta_i \tag{3.31}$$

In addition, items that are common with each other must have the same level of redundancy.

$$C_{ij} \implies r_i = r_j \qquad \forall\ i\ \in \phi \tag{3.32}$$

This only comes into effect when components are repairable items, and prevents the case where the benefits of redundancy within a given item are removed by making that item common with one that does not use redundancy.

Constraints on Objective Values

One driving constraint of the supportability strategy optimization problem is the requirement that the amount of maintenance resources available during the mission must be sufficient to cover all maintenance demands during the mission, with probability at least equal to POS_R. That is,

$$POS_S(n)POS_T(t_{max}) \geq POS_R \tag{3.33}$$

where $POS_S(n)$ is the probability that the spares allocation n is sufficient, and $POS_T(t_{max})$ is the probability that the maximum allowable crew time t_{max} is sufficient to perform all required maintenance actions.

However, the primary effect of this constraint is that it drives the selection of the required spares allocation n, which in turn drives the total mass m_{tot} associated with that supportability strategy. Once the ORU list is defined, the structure of the spares allocation problem can be exploited to enable fast calculation of the optimal spares allocation that achieves a POS requirement while minimizing mass. Therefore, in the formulation used for this research, the spares allocation n is not a decision variable in the COP and is instead considered part of the mass model, described in Section 4.2.3. Once a supportability strategy is defined, an ORU list is created (see Section 4.1), the optimal spares allocation for that ORU list is determined using the methodology described in Section 4.2.3, and the mass of that allocation is used as the spares contribution to the total mass objective.

The inclusion of the POS associated with crew time in the calculation of spares requirements is another contribution of this thesis. Previous spares allocation models use POS to exclusively refer to to the probability that sufficient spares are available for maintenance. However, crew time is also a necessary and limited resource, and the exclusion of crew time POS from supportability risk evaluations results in an underestimate of the actual risk associated with the availability of maintenance resources. By including $POS_T(t_{max})$ as a term in the overall POS, this methodology ensures that the amount of resources provided – both spares and crew time – do achieve the POS requirement POS_R set out in the mission definition, and therefore the calculated POS provides a more complete picture of supportability-related risk.

Beyond the calculation of spares mass, however, equation 3.33 implies an additional constraint on the probability of sufficient crew time. Specifically, that constraint cannot be met unless

$$POS_T(t_{max}) > POS_R \tag{3.34}$$

since all POS values are between 0 and 1. That is, if the probability of having sufficient crew time is below the required POS, then there is no spares allocation that will satisfy the risk requirement in equation 3.33. $POS_T(t_{max})$ is a direct function of the supportability strategy variables because it is unrelated to the spares allocation, and therefore this equation constrains feasible assignments to those variables in the COP.

In addition, the expected amount of maintenance crew time must be less than or equal to the maximum allowable crew time for maintenance.

$$\mathrm{E}[T] \leq t_{max} \tag{3.35}$$

This constraint stems from the fact that t_{max} is defined as the total amount of crew time in the mission not taken up by other crew time liens, where a crew time lien is defined as the time required for all mission activities other than utilization and maintenance, including work tasks such as routine operations, inspections, and training, as well as non-work tasks such as exercise, meals, and sleeping [7]. The time available for utilization will be equal to t_{max} minus the amount of time spent on maintenance activities, T. If $\mathrm{E}[T] > t_{max}$, then the expected time available for utilization will be negative; this is not considered an acceptable outcome.

3.3 Summary

The overall supportability strategy optimization problem is given by equation 3.36 on page 148. The net result is a discrete, finite-domain, nonlinear, multiobjective COP. The objective is to find Pareto-optimal assignments to the supportability strategy decision variables $\mathcal{X}$ that minimize total mass m_{tot} and expected crew time required for maintenance $\mathrm{E}[T]$ for a particular system $\mathcal{S}$, mission $\mathcal{M}$, and set of modeling parameters $\mathcal{P}$. The evaluation of these objectives is performed by the supportability model described in Section 4.2. Constraints 3.36b through 3.36t specify feasible assignments of supportability strategy decision variables, based on the structure of the system. Constraints 3.36u and 3.36v specify feasible amounts of crew time that can be spent on maintenance, both in terms of the POS associated with the maximum amount available and in terms of the expected amount, both of which are evaluated by the crew time model described in Section 4.2.2.

This formulation is a key novel contribution of this research. The level of maintenance encoding is particularly important, since it enables examination of a much wider range of options than existing models, which require pre-defined LRUs and SRUs. In contrast, this formulation enables optimization of the definition of each ORU, based on combinations of different assemblies, subassemblies, and components within the system. The constraints between various decisions are also important for reducing the size of the tradespace in order to make the problem more tractable and accelerate optimization. In addition, the inputs are defined such that demand for spares (R&R events) and demand for crew time (R&R and non-R&R events) are represented by separate stochastic processes, which significantly improves crew time estimates, as shown in Chapter 5. Overall, The ability to mathematically represent a wide range of supportability strategies using discrete decision variables, along with constraints defining feasible assignments is critical to enabling systematic optimization of supportability strategies in order to answer the research question defined in Chapter 2.

$$\min_{\mathcal{X}} \quad (m_{tot},\ \mathrm{E}[T]) = f\,(\mathcal{X};\ \mathcal{S},\ \mathcal{M},\ \mathcal{P}) \tag{3.36a}$$

$$\text{s.t.} \quad \theta_i \implies \neg\nu_i \qquad \forall\ i \in \phi \tag{3.36b}$$

$$\bigwedge_{\substack{j\in\phi\\ p_j=i}} (u_j \vee \nu_j) \iff \nu_i \qquad \forall\ i \in \phi \mid \neg\theta_i \tag{3.36c}$$

$$\nu_i \implies \neg u_i \qquad \forall\ i \in \phi \tag{3.36d}$$

$$u_i \implies \bigvee_{\substack{j,k\in\phi\\ \theta_j\wedge\theta_k\\ j\neq k}} \left(D_{ij} \wedge D_{ik} \wedge \neg(u_j \vee u_k) \wedge \neg \bigvee_{\substack{l\in\phi\\ \neg\theta_l}} \big(u_l \wedge D_{il} \wedge \big(D_{lj} \vee D_{lk}\big)\big) \right) \qquad \forall\ i \in \phi \mid \neg\theta_i \tag{3.36e}$$

$$(p_i = 0) \wedge \neg\nu_i \implies u_i \qquad \forall\ i \in \phi \tag{3.36f}$$

$$\eta_i = 1 \qquad \forall\ i \in \phi \mid \neg\psi_i \tag{3.36g}$$

$$\neg\nu_i \wedge \neg u_i \implies \eta_i = 1 \qquad \forall\ i \in \phi \mid \psi_i \tag{3.36h}$$

$$\neg\hat{C}_{ij} \implies \neg C_{ij} \qquad \forall\ i,\ j \in \phi \tag{3.36i}$$

$$\neg u_i \vee \neg u_j \implies \neg C_{ij} \qquad \forall\ i,\ j \in \phi \mid i \neq j \tag{3.36j}$$

$$C_{ii} \iff u_i \qquad \forall\ i \in \phi \tag{3.36k}$$

$$C_{ij} \implies C_{ji} \qquad \forall\ i,\ j \in \phi \tag{3.36l}$$

$$C_{ij} \wedge C_{jk} \implies C_{ik} \qquad \forall\ i,\ j,\ k \in \phi \tag{3.36m}$$

$$\neg\hat{\xi}_i \implies \neg\xi_i \qquad \forall\ i \in \phi \tag{3.36n}$$

$$\neg u_i \implies \neg\xi_i \qquad \forall\ i \in \phi \tag{3.36o}$$

$$C_{ij} \implies \neg\xi_i \wedge \neg\xi_j \qquad \forall\ i,\ j \in \phi \mid i \neq j \tag{3.36p}$$

$$\xi_i \implies \bigvee_{\substack{j\in\phi\\ j\neq i}} (\xi_j \wedge \neg C_{ij}) \qquad \forall\ i \in \phi \tag{3.36q}$$

$$\xi_s \iff \bigvee_{i\in\phi} \xi_i \tag{3.36r}$$

$$r_i = 0 \qquad \forall\ i \in \phi \mid \neg\theta_i \tag{3.36s}$$

$$C_{ij} \implies r_i = r_j \qquad \forall\ i, j \in \phi \tag{3.36t}$$

$$POS_T(t_{max}) > POS_R \tag{3.36u}$$

$$\mathrm{E}[T] \leq t_{max} \tag{3.36v}$$

Chapter 4

Methodology II: Supportability Evaluation and Optimization

The previous chapter defined the supportability strategy optimization problem as a multiobjective Constraint Optimization Problem (COP); this chapter defines the model used to evaluate candidate supportability strategies and solve the COP. Section 4.1 describes the process for defining an ORU list based on a particular system and supportability strategy. This portion of the methodology, along with the COP formulation presented in Chapter 3, is a key novel contribution enabling the use of discrete decision variables to define a variety of ORU lists from a general system description. All other supportability modeling techniques start from an ORU list or something similar; in contrast, this approach enables the definition of the ORU list as part of the optimization process.

Section 4.2 then describes the model used to assess the mass and crew time associated with that ORU list. Some elements of this model are standard practice, such as the marginal analysis approach used to optimize spares allocations. However, the model itself incorporates several novel contributions. In particular, the crew time model includes an assessment of the probability of sufficient crew time, which is used during spares allocation optimization to ensure that the overall POS value includes risks associated with both spares and crew time. In addition, this model defines separate stochastic demand processes for spares and crew time, a novel contribution that improves the accuracy of crew time estimates. The mass model also presents a novel method for evaluating the relationship between POS and the amount of ODM feedstock provided, which is critical for including ODM as a supportability strategy option. While previous versions of this model have been presented in the literature as part of the development of this research [24, 25], this is the first version of this methodology that includes epistemic uncertainty and does not require Monte Carlo simulation. This last point is also a contribution of the evaluation methodology in general: no Monte Carlo simulation is used to evaluate mass, crew time, or risk. This enables very rapid evaluation and facilitates the use of this model in large-scale optimization.

Section 4.3 describes the multiobjective branch and bound strategy used to solve the COP and the process used to calculate lower bounds on mass and crew time for partial assignments to the decision variables. The key contribution for this portion of the methodology is the formulation of these lower bounds, which enable optimization. Finally, Section 4.4 presents an overall summary of the methodology, highlighting key contributions.

4.1 Creating the Orbital Replacement Unit List

The key input for supportability evaluation is the ORU list ω, which is created based on the system description $\mathcal{S}$ and an assignment to the supportability strategy variables $\mathcal{X}$. The ORU list is the list of all ORUs that will be considered for maintenance, along with their associated supportability characteristics. An ORU ω_i is a component or set of components that have been encapsulated into a single unit in order to facilitate maintenance. For maintenance purposes, each ORU is treated as a monolithic unit – that is, whenever any of the components within an ORU fail, that ORU is considered failed, and is replaced. When there are redundant instances of a component within an ORU, the ORU is replaced after all redundant instances have failed. Stowed spares, maintenance items, and the initially installed instances of each ORU within the system are all referred to as ORUs.

ORUs are the core unit of maintenance for supportability analysis. Spares and maintenance items are provided for ORUs, and maintenance crew time is spent on crew actions related to ORUs, such as inspection, cleaning, troubleshooting, and removal and replacement. Therefore, the supportability characteristics of a system are defined by the supportability characteristics of the ORU list for that system, given the particular supportability strategy used.

As mentioned in Section 3.2, this formal process for defining ORUs using a system description and assignments to supportability strategy variables is a key contribution of this thesis, since it enables multiobjective optimization of a wide range of supportability strategies.

4.1.1 Characterizing Replaceable Items

The first step in defining the ORU list is to characterize the set of replaceable items, i.e. items for which $u_i = 1$. Replaceable items are similar to ORUs in that they represent encapsulation of components into specific replaceable units for maintenance, as a function of the supportability strategy decisions, particularly those related to level of maintenance. Each replaceable item's supportability characteristics (e.g. mass, failure rate, etc.) are determined as a function of the components within each item. However, multiple replaceable items may be combined into a single ORU based on commonality decisions, and therefore represent an intermediate step in the creation of the ORU list. The process of applying commonality decisions to the set of replaceable items to generate and characterize ORUs is described in Section 4.1.2.

The key characteristics for each replaceable item i that are derived from the system description $\mathcal{S}$ and supportability strategy $\mathcal{X}$ are:

- Quantity $q_{\varrho i}$,

- Mass $m_{\varrho i}$,

- Failure rate $\Lambda_{\varrho i}$, characterized by expected value $\mathrm{E}[\Lambda_{\varrho i}]$ and variance $\mathrm{Var}[\Lambda_{\varrho i}]$

- Crew action rate rate $\Lambda_{C\varrho i}$, characterized by expected value $\mathrm{E}[\Lambda_{C\varrho i}]$ and variance $\mathrm{Var}[\Lambda_{C\varrho i}]$
- Life limit $\ell_{\varrho i}$
- Crew time required for maintenance $t_{\varrho i}$, and
- ODM manufacturing decision $\xi_{\varrho i}$.

Note that, in order to distinguish the characteristics of replaceable items from those of components, a subscript ϱ is added. Similarly, the characteristics of ORUs (described in Section 4.1.2) are denoted using a subscript ω. For example, the unit mass of a component is indicated by m_i, the unit mass of a replaceable item is denoted $m_{\varrho i}$, and the unit mass of an ORU is denoted $m_{\omega i}$. When these characteristics are represented in vector format, the subscript i indicating a specific replaceable item is dropped; for example, the mass of all replaceable items is represented by the vector m_ϱ, and the mass of all ORUs is represented by the vector m_ω.

In application, the characteristics of the replaceable items can be determined from the system description and supportability strategy variables through a series of matrix operations. However, for the purposes of explanation the various steps taken in calculation of replaceable item characteristics are also defined in terms of summation and multiplication related to a single item, since this is often more intuitive. The vectorized forms of these calculations – which are typically easier to implement in code and more computationally efficient – are also described alongside the single-item definitions, and are summarized at the end of this section.

Redundancy Matrix

The quantity matrix Q (defined by equation 3.9 on page 132) does not include redundant instances of any components, since these are not part of the operating quantity of that item. Therefore, once a supportability strategy decision is made the redundancy matrix R must be calculated, which has entries R_{ij} indicating the number of redundant instances of item j in item i.

$$R = (I + D)\,\mathrm{diag}(r) \tag{4.1}$$

Here r is the vector with entries r_i indicating the number of redundant instances of item i, and $\mathrm{diag}(\cdot)$ is the vector diagonalization function, which creates a square matrix in which the diagonal entries are the entries of the input vector, and all other entries are 0 (i.e. if $A = \mathrm{diag}(a)$, then $A_{ii} = a_i$ and $A_{ij} = 0$ for all $i \neq j$). Right-multiplication of $I + D$ multiplies the columns j of the matrix $I + D$ (which are 1 where the column j is a descendant of the row i or $j = i$) by the number of redundant instances of item j. Since $r_j = 0$ for items that are not components, the only nonzero entries in R indicate places where j is a descendant of i and j is a component with at least one redundant instance. These two values – quantity Q and redundant quantity R – are kept separate because they impact mass, failure rates, and crew action rates in different ways.

Total Number of Modules for Functional Distribution

The total number of modules that a given item is distributed across, $\hat{\eta}$, is equal to the product of the number of modules η_j for each subsystem j that i is a part of. Since η is constrained to be 1 for any item that is not a subsystem and all η are positive, $\hat{\eta}$ can be calculated using a similar manner to that used to calculate the quantity vector q in equation 3.8.

$$\hat{\eta} = \exp\left[(I + D^T)\ln[\eta]\right] \tag{4.2}$$

This number is used to scale component masses and adjust quantities in response to the distributed functionality decision.

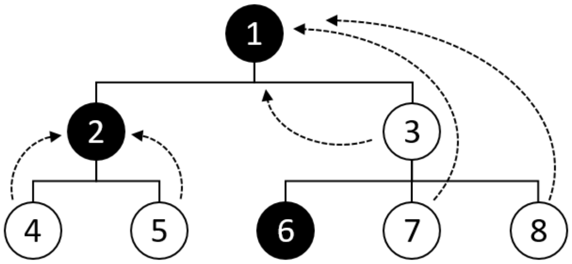

Figure 4.1: Example level of maintenance selection. Items 1, 2, and 6 (shaded black) are designated as replaceable items. Items 4 and 5 are considered to be a part of replaceable item 2, while 3, 7, and 8 are a part of replaceable item 1, and replaceable item 6 contains only itself.

Assigning Components to Replaceable Items

The characteristics of these replaceable items are defined as a function of the items that are contained within it. Each item i is considered to be contained within the lowest replaceable item that either is i or is an ancestor of i. Replaceable items may be partially nested within each other, as is shown in Figure 4.1. In this example, item 1 has children 2 and 3. Item 2 has children 4 and 5, and item 3 has children 6, 7, and 8. Items 1, 2, and 6 are selected as replaceable items. With this structure, replaceable item 6 contains only itself (since it is a component and has no children). Replaceable item 2 contains both of its children, 4 and 5. Replaceable item 1 contains 1, 3, 7, and 8 – itself and all of its descendants that are not covered by a replaceable item that is a descendant.

The replaceable item content matrix W has entries $W_{ij} = 1$ if item j is contained within i, and 0 otherwise. This matrix is generated by starting with a matrix with entries $i,\ j$ equal to 1 where i is a replaceable item and j is a descendant of i and 0 elsewhere, then iteratively taking the Hadamard product with respect to a matrix that with entries $i,\ j$ equal to 0 where i is a replaceable item and j is the k^{th} descendant of a replaceable item that is itself a descendant of i. The process is described in Algorithm 4.1. While this matrix could be defined by iterating over each item in the system, this algorithm generates W using only matrix operations, and as a result provides fast execution in code. In addition, the format of W enables matrix-based

Algorithm 4.1 Get-Content-Matrix returns the replaceable item content matrix W, based on the adjacency matrix A, descendants matrix D, and replaceable item vector u. The entries W_{ij} are equal to 1 if j is contained within replaceable item i, and 0 otherwise.

```
Get-Content-Matrix($A$, $D$, $u$)
$U \leftarrow \mathrm{diag}(u)$
$W \leftarrow U(I + D) \circ (1 - UDU)$
$k \leftarrow 1$
while $UDUA^k$ is not empty
      $W \leftarrow W \circ (1 - UDUA^K)$
      $k \leftarrow k + 1$
return $W$
```

calculations of all other replaceable item characteristics, which similarly accelerates the analysis process. As an example, the content matrix for the level of maintenance selection shown in figure 4.1 is

$$W = \begin{bmatrix} 0 & 0 & 1 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \tag{4.3}$$

The matrix U is the diagonalization of the vector of replaceable items u, and has entries $U_{ii} = 1$ where i is a replaceable item, and 0 elsewhere. Left-multiplication of another matrix by U therefore results in a matrix whose rows are empty (i.e. all 0s) when they correspond to a non-replaceable item and identical to the other matrix elsewhere. Similarly, right-multiplication of another matrix by U zeros out columns that correspond to non-replaceable items and leaves all other entries untouched.

As a result, $U(I + D)$ is a matrix with entries equal to 1 where i is a replaceable item and j is i or a descendant of i and 0 otherwise. UDU is the matrix indicating replaceable items that are themselves descendants of replaceable items; entries are equal to 1 when both i and j are replaceable items and j is a descendant of i, and 0 otherwise. Therefore $1 - UDU$ has entries equal to 0 where j is a descendant of i and both i and j are replaceable items, and 1 otherwise. Element-wise multiplication (using the Hadamard product $\circ$) of $U(I + D)$ by $1 - UDU$ sets nonzero entries in $U(I + D)$ that correspond to items that are replaceable items and descendants of replaceable items to zero, leaving all other entries untouched. This process effectively filters replaceable items out of the descendant list of each replaceable item.

To remove items that are contained within those descendant replaceable items, the process is iterated while right-multiplying UDU by increasing powers of the adjacency matrix A. Left-multiplying a matrix by the adjacency matrix can be thought of as "walking" down the indenture structure, since the rows of the resulting matrix will be equal to 1 where they correspond to the children of entries from that row that were 1 in the initial matrix. Left-multiplying by the adjacency matrix again will find the children of those children, and so on. The matrix $UDUA^k$ therefore has entries equal to 1 where i is a replaceable item and j is the k^{th} descendant of a replaceable item that is a descendant of i and 0 elsewhere. That is, $UDUA^0$ is simply the matrix indicating replaceable item descendants described above, $UDUA^1$ indicates the children of those replaceable item descendants, $UDUA^2$ indicates their children's children, and so on. Therefore, iterative element-wise multiplication of W by $1 - UDUA^k$ sets any entries of W that correspond to the k^{th} descendants of replaceable items that are descendants of replaceable items equal to zero. The process is repeated until $UDUA^k$ is an empty matrix, indicating that there are no further descendants to remove.

Calculating Replaceable Item Characteristics

A replaceable item's unit mass $m_{\varrho i}$ is the sum of the mass associated with each component contained within it, which is itself the product of the component unit mass, scaled according to the number of modules that functionality is distributed across, and the number of instances of that component in i, including both active and redundant instances. The redundancy mass overhead factor is applied to any redundant instances, and the mass is also scaled by the ODM mass overhead factor γ_{odm}, if ODM is used.

$$m_{\varrho i} = \left(\sum_{j \in \phi | W_{ij}} \left(Q_{ij} + (1 + \gamma_r) R_{ij} \right) \theta_j m_j \left(1 - \gamma_j + \frac{\gamma_j}{\hat{\eta}_j} \right) \right) (1 + \gamma_{odm} \xi_i) \tag{4.4}$$

A vectorized version of this calculation is given by equation 4.5. The replaceable item mass vector m_ϱ has entries equal to $m_{\varrho i}$ if i is a replaceable item and 0 if it is not, is defined as

$$m_\varrho = \left((Q + (1 + \gamma_r) R) \circ W \right) \left(\theta \circ m \circ \left(1 - \left(1 - \hat{\eta}^{\circ -1} \right) \circ \gamma \right) \right) \circ (1 + \gamma_{odm} \xi) \tag{4.5}$$

The matrix $(Q + (1 + \gamma_r)R) \circ W$ has entries i, j giving the effective number of item j within replaceable item i. This is called the effective number because it is not necessarily an integer; the redundancy mass overhead factor means that each redundant instance may, for mass calculation purposes, account for more than 1 times the mass of the component. This matrix is then multiplied by the scaled mass of each component, which is captured in the vector $\theta \circ m \circ (1 - (1 - \hat{\eta}^{\circ -1}) \circ \gamma)$. This series of element-wise multiplications applies the scaling function described in equation 3.2 to each item and ensures that the mass of any non-component item is 0. This latter condition is necessary because components are the only items in the system that have a mass in and of themselves, rather than a mass based on the set of components that they contain, and therefore there is no mass associated with non-component items.[a] The resulting vector has entries equal to 0 for non-replaceable items and equal to the sum of the scaled mass of all components contained within the item for replaceable items. If ODM is used, these values are then scaled again to account for the ODM mass overhead factor γ_{odm} by performing an element-wise multiplication with the

[a] Note that, based on the inputs described in Table 3.1, the vectors m and γ should only have values specified for components; however, θ is left in equation 4.5 in order to make this removal of non-component items explicit. This same reasoning, applied to all other system characteristics defined only for components, is used for all following equations involving θ.

vector $1 + \gamma_{odm}\xi$.

Similarly, the expected value of a replaceable item's failure rate Λ_{ϱ_i} is the sum of the expected values of the failure rates of the components contained within it, multiplied by the associated quantity, K-factor, and duty cycle and divided by one plus the number of redundant components multiplied by the redundancy effectiveness factor. If ODM is used, the ODM rate factor is also applied to the total.

$$\mathrm{E}[\Lambda_{\varrho i}] = \left(\sum_{j \in \phi | W_{ij}} \frac{\theta_j \bar{\lambda}_j \kappa_j d_j Q_{ij}}{1 + \gamma_{\Lambda r} r_j} \right) (1 + \gamma_{\Lambda odm} \xi_i) \tag{4.6}$$

When no redundancy is present, this is simply the sum of the failure rates of the constituent components, since the replaceable item would be replaced whenever any component within it fails.

However, if a component does have redundancy, then the replaceable item only needs to be replaced after the initially active component fails and all redundant instances fail – that is, after the $(1 + r)^{\text{th}}$ failure of that component. If redundancy is implemented without any impact on failure rates (i.e. the redundancy effectiveness factor $\gamma_{\Lambda r}$ is 1), then this impact can be approximated by dividing the expected value of the failure rate by $1+r$. However, as noted earlier, the additional infrastructure required to implement redundancy may incur a failure rate overhead, which is captured by a redundancy effectiveness factor less than 1. Therefore, the expected failure rate is divided by $1 + \gamma_{\Lambda r} r$ instead. This modeling approach is an approximation, since the overall failure rate of the assembly is no longer constant when there are standby redundant components available. Specifically, when redundancy is present the time to final failure (i.e. the one triggering replacement) follows an Erlang distribution rather than an exponential distribution. This violates the Constant Failure Rate (CFR) approximation used to assess spares requirements, described in Section 4.2.1, but it provides a simple model with the flexibility to rapidly examine a range of options, and provides sufficient accuracy in the range of interest (i.e. high POS values) for most supportability analyses. The accuracy of this redundancy approximation is examined as part of model verification and validation in Chapter 5.

To simplify the vectorized equations for this step, define a vector $\bar{\lambda}_\theta$ with entries $\bar{\lambda}_{\theta i}$ indicating the effective mean failure rate of item i if i is a component, and 0 otherwise:

$$\bar{\lambda}_\theta = \theta \circ \bar{\lambda} \circ \kappa \circ d \circ (1 + \gamma_{\Lambda r} r)^{\circ -1} \tag{4.7}$$

The effective mean failure rate is the mean failure rate multiplied by the K-factor and duty cycle and divided by one plus the level of redundancy (i.e. all elements of the summand of equation 4.6 except for the quantity Q_{ij}). The replaceable item expected failure rate vector is then

$$\mathrm{E}[\Lambda_\varrho] = (Q \circ W)\bar{\lambda}_\theta \circ (1 + \gamma_{\Lambda odm}\xi) \tag{4.8}$$

The variance of a replaceable item's failure rate is also derived from the variance of the failure rates of the components within it. This research uses the lognormal model for failure rate uncertainty (described in greater detail in Section 4.2.1), and therefore the variance in failure rate for a component with mean failure rate $\bar{\lambda}_i$ and error factor ε_i is [86]

$$\mathrm{Var}[\Lambda_i] = \bar{\lambda}_i^2 \left(e^{\left(\frac{\ln \varepsilon_i}{1.645}\right)^2} - 1 \right) \tag{4.9}$$

The variance of a linear combination of independent random variables $\mathrm{Var}[\sum_i k_i X_i]$ is equal to $\sum_i k_i^2 \mathrm{Var}[X_i]$ [86], and therefore the variance of the failure rate of a replaceable item is

$$\mathrm{Var}[\Lambda_{\varrho i}] = \left(\sum_{j \in \phi | W_{ij}} \left(\frac{\theta_j \bar{\lambda}_j \kappa_j d_j Q_{ij}}{1 + \gamma_{\Lambda r} r_j} \right)^2 \left(e^{\left(\frac{\ln \varepsilon_j}{1.645}\right)^2} - 1 \right) \right) (1 + \gamma_{\Lambda odm}\xi_i)^2 \tag{4.10}$$

which can be expressed in matrix-vector form as

$$\mathrm{Var}[\Lambda_\varrho] = (Q \circ W)^{\circ 2} \left(\bar{\lambda}_\theta^{\circ 2} \circ \left(\exp\left[\left(\frac{\ln[\varepsilon]}{1.645} \right)^{\circ 2} \right] - 1 \right) \right) \circ (1 + \gamma_{\Lambda odm}\xi)^{\circ 2} \tag{4.11}$$

where $^{\circ 2}$ is an element-wise square of the matrix, defined such that the entries of $A^{\circ 2}$ are A_{ij}^2.

The expected value and variance of the replaceable item crew action rate $\Lambda_{C\varrho}$ are calculated in a similar manner to those of the failure rate. Defining the component effective mean crew action rate vector $\bar{\lambda}_{C\theta}$ as

$$\bar{\lambda}_{C\theta} = \theta \circ \bar{\lambda}_C \circ \kappa \circ d \circ (1 + \gamma_{\Lambda r} r)^{\circ -1} \tag{4.12}$$

the expected value and variance of the replaceable item crew action rate are

$$\mathrm{E}[\Lambda_{C\varrho}] = (Q \circ W)\bar{\lambda}_{C\theta} \circ (1 + \gamma_{\Lambda odm}\xi) \tag{4.13}$$

$$\mathrm{Var}[\Lambda_{C\varrho}] = (Q \circ W)^{\circ 2} \left(\bar{\lambda}_{C\theta}^{\circ 2} \circ \left(\exp\left[\left(\frac{\ln[\varepsilon]}{1.645} \right)^{\circ 2} \right] - 1 \right) \right) \circ (1 + \gamma_{\Lambda odm}\xi)^{\circ 2} \tag{4.14}$$

The moment-matching technique described here is an approximation that is intended to enable the aggregation of component characteristics into single ORU characteristics to facilitate analysis without the computational overhead that would be required to compute exact distributions of replaceable item and ORU failure and crew action rates. Effectively, this approach uses the expected value and variance of the number of times that a given replaceable item would require replacement crew action due to failures/anomalies associated with components within it in order to characterize the failure or crew action rate of that replaceable item. However, this approach does introduce some error, since the sum of lognormal random variables (i.e. component failure rates Λ_i or crew action rates Λ_{Ci}) is not lognormal. The approach used to model redundancy (i.e. dividing the rate by $1 + \gamma_{\Lambda r} r$) is also an approximation, as discussed earlier. However, the flexibility provided by this assumption enables the analysis of a wide range of supportability strategies under the same general model, and is considered to be sufficiently accurate for this application. The accuracy of this approach is investigated in greater detail in Chapter 5.

A replaceable item's life limit is equal to the lowest life limit of any of its constituent components, adjusted to account for duty cycle and redundancy.

$$\ell_{\varrho i} = \min_{j \in \phi | W_{ij}} \frac{\ell_j(1 + r_j)}{d_j} \tag{4.15}$$

Components with lower duty cycles accumulate operating time at a slower rate than those with higher duty cycles, and therefore their effective life limit is lower. Each redundant instance of component j is a multiplier on the life limit, since standby redundancy is used in this application and the redundant instance can take over once the previous instance reaches the end of its lifetime. The value $\ell(1+r)$ can be thought of as the effective life limit of the component, given redundant instances. Note that the redundancy effectiveness factor $\gamma_{\Lambda r}$ is not used in the life limit calculation, since it only applies to failure rates and crew action rates.

The vector of replaceable item life limits ℓ_ϱ is generated by calculating a vector containing the inverse of the effective life limit (i.e. the duty cycle divided by life limit) for each component, diagonalizing it, left-multiplying by W to obtain a matrix with entries i, j containing the inverse of the effective life limit of component j, where j is contained within i, then taking the maximum and inverting again to obtain the replaceable item life limit.

$$\ell_\varrho = \left(\max_R \left(W \operatorname{diag}(\theta \circ (\ell \circ (1 + r))^{\circ -1} \circ d) \right) \right)^{\circ -1} \tag{4.16}$$

Here $\max\limits_R$ is the maximum along the rows of a matrix, and the life limit of any item that does not have a life limit specified in the inputs $\mathcal{S}$ is assumed to have a life limit of ∞ (and therefore $\ell^{-1} = 0$). The reason for inverting the life limit vector before diagonalizing is to transform the data into a format where items that do not contribute to the life limit have a value of 0 rather than ∞ in order to simplify calculations.[b]

Crew time required for maintenance and the ODM decision associated with each

[b] Note that the second element-wise inversion in equation 4.16 is not strictly necessary, since it is the inverse of the replaceable item life limit that is used for ORU life limit calculations (see equation 4.36). However, it is included here to maintain consistency in using life limit durations to parameterize scheduled maintenance.

replaceable item are given directly by the inputs and decision variables:

$$t_\varrho = t \circ u \tag{4.17}$$

$$\xi_\varrho = \xi \tag{4.18}$$

(Note that equation 4.18 is true due to the fact that only replaceable items can be selected to use ODM, and therefore ξ does not have to be filtered using u in the same way that t is.) Similarly, the total quantity of a given replaceable item in the system is equal to the total quantity of that item in the system multiplied by the number of modules that its functionality is divided across:

$$q_\varrho = q \circ \hat{\eta} \circ u \tag{4.19}$$

Summary

The end result of the above calculations is a set of vectors describing the replaceable items as a function of the system description and supportability strategy. For each vector, the i^{th} entry gives the value associated with replaceable item i, if i is a replaceable item, and is 0 if i is not a replaceable item. The vectorized calculations used to obtain these vectors are summarized in equations 4.20 through 4.30.

$$m_\varrho = ((Q + (1+\gamma_r)R) \circ W)\left(\theta \circ m \circ \left(1 - (1 - \hat{\eta}^{\circ -1}) \circ \gamma\right)\right) \circ (1 + \gamma_{odm}\xi) \tag{4.20}$$

$$\bar{\lambda}_\theta = \theta \circ \bar{\lambda} \circ \kappa \circ d \circ (1 + \gamma_{\Lambda r} r)^{\circ -1} \tag{4.21}$$

$$\mathrm{E}[\Lambda_\varrho] = (Q \circ W)\bar{\lambda}_\theta \circ (1 + \gamma_{\Lambda odm}\xi) \tag{4.22}$$

$$\mathrm{Var}[\Lambda_\varrho] = (Q \circ W)^{\circ 2}\left(\bar{\lambda}_\theta^{\circ 2} \circ \left(\exp\left[\left(\frac{\ln[\varepsilon]}{1.645}\right)^{\circ 2}\right] - 1\right)\right) \circ (1 + \gamma_{\Lambda odm}\xi)^{\circ 2} \tag{4.23}$$

$$\bar{\lambda}_{C\theta} = \theta \circ \bar{\lambda}_C \circ \kappa \circ d \circ (1 + \gamma_{\Lambda r} r)^{\circ -1} \tag{4.24}$$

$$\mathrm{E}[\Lambda_{C\varrho}] = (Q \circ W)\bar{\lambda}_{C\theta} \circ (1 + \gamma_{\Lambda odm}\xi) \tag{4.25}$$

$$\mathrm{Var}[\Lambda_{C\varrho}] = (Q \circ W)^{\circ 2}\left(\bar{\lambda}_{C\theta}^{\circ 2} \circ \left(\exp\left[\left(\frac{\ln[\varepsilon]}{1.645}\right)^{\circ 2}\right] - 1\right)\right) \circ (1 + \gamma_{\Lambda odm}\xi)^{\circ 2} \tag{4.26}$$

$$\ell_\varrho = \left(\max_R \left(W \operatorname{diag}(\theta \circ (\ell \circ (1+r))^{\circ -1} \circ d)\right)\right)^{\circ -1} \tag{4.27}$$

$$t_\varrho = t \circ u \tag{4.28}$$

$$\xi_\varrho = \xi \tag{4.29}$$

$$q_\varrho = q \circ \hat{\eta} \circ u \tag{4.30}$$

4.1.2 Applying Commonality to Create ORUs

Once the replaceable items are characterized, commonality decisions are applied to generate the ORU list ω. When two items are common, they can share the same ORU – that is, a spare for one instance of that item is equivalent to a spare for another instance. The entries of the commonality matrix C are 1 where i is common with j, including the case of $i = j$. As a result, each unique non-empty row of the commonality matrix represents an ORU, with the entries j equal to 1 if j is a replaceable item covered by that ORU and 0 if it is not.

Commonality between items typically requires some design adaptation and compromise to ensure that an ORU can fit the needs and constraints of multiple sites in the system. To model the impacts of this compromise, the characteristics of the common ORU are defined by taking the "worst" characteristic of the set of replaceable items that are common: the highest mass, highest failure rate expectation and variance, highest crew action rate expectation and variance, lowest life limit, and highest crew time required for maintenance. This is a conservative assumption, and is used to enable the selection of various potential common sets of items without having to pre-specify what the characteristics of a common ORU for those items would be. The quantity of any ORU in the system is the sum of the quantities of each replaceable item that it covers.

The ORU list ω consists of a set of vectors describing each ORU, analogous to the item list ϕ. Specifically, $\omega = \{m_\omega,\ \mathrm{E}[\Lambda_\omega],\ \mathrm{Var}[\Lambda_\omega],\ \mathrm{E}[\Lambda_{C\omega}],\ \mathrm{Var}[\Lambda_{C\omega}],\ \ell_\omega,\ t_\omega,\ \xi_\omega,\ q_\omega\}$ describes the mass, expected failure rate, variance in failure rate, expected crew action rate, variance in crew action rate, life limit, crew time required for maintenance, ODM decision, and quantity of each ORU using a set of vectors, in a similar fashion to that

Table 4.1: Summary of vectors characterizing ORUs. The i^{th} entry in each vector provides the value associated with ORU i.

Description	Symbol	Units
Mass	m_ω	kg
Expected failure rate	$\mathrm{E}[\Lambda_\omega]$	h^{-1}
Variance in failure rate	$\mathrm{Var}[\Lambda_\omega]$	h^{-2}
Expected crew action rate	$\mathrm{E}[\Lambda_{C\omega}]$	h^{-1}
Variance in crew action rate	$\mathrm{Var}[\Lambda_{C\omega}]$	h^{-2}
Life limit	ℓ_ω	h
Crew time required for maintenance	t_ω	h
ODM Decision	ξ_ω	
Quantity	q_ω	

used to describe replaceable items. These vectors and their associated units are summarized in Table 4.1.

Let C_U be the matrix consisting of the unique, nonempty rows of the commonality matrix C. That is, each row of C that is not empty (i.e. not full of 0s) appears in C_U only once. Each row of C_U represents a group of common replaceable items that will be combined into an ORU, with entries C_{Uij} is equal to 1 if replaceable item j is covered by ORU i, and 0 otherwise. The vectors characterizing the ORU list are

$$m_\omega = \max_R \left(C_U \operatorname{diag}(m_\varrho)\right) \tag{4.31}$$

$$\mathrm{E}[\Lambda_\omega] = \max_R \left(C_U \operatorname{diag}(\mathrm{E}[\Lambda_\varrho])\right) \tag{4.32}$$

$$\mathrm{Var}[\Lambda_\omega] = \max_R \left(C_U \operatorname{diag}(\mathrm{Var}[\Lambda_\varrho])\right) \tag{4.33}$$

$$\mathrm{E}[\Lambda_{C\omega}] = \max_R \left(C_U \operatorname{diag}(\mathrm{E}[\Lambda_{C\varrho}])\right) \tag{4.34}$$

$$\mathrm{Var}[\Lambda_{C\omega}] = \max_R \left(C_U \operatorname{diag}(\mathrm{Var}[\Lambda_{C\varrho}])\right) \tag{4.35}$$

$$\ell_\omega = \left(\max_R \left(C_U \operatorname{diag}(\ell_\varrho^{\circ -1})\right)\right)^{\circ -1} \tag{4.36}$$

$$t_\omega = \max_R \left(C_U \operatorname{diag}(t_\varrho)\right) \tag{4.37}$$

$$\xi_\omega = \max_R \left(C_U \operatorname{diag}(\xi_\varrho)\right) \tag{4.38}$$

$$q_\omega = C_U q_\varrho \tag{4.39}$$

Equations 4.31 through 4.38 identify the worst value for each characteristic from the set of common replaceable items by right-multiplying C_U by the diagonalized vector of replaceable item characteristics and taking the maximum along each row. To calculate the life limit (equation 4.36), the same approach used in equation 4.16 is used where the calculations are carried out on inverted replaceable item life limit values. The quantity of each ORU, which is simply the sum of the quantity $q_{\varrho i}$ for each replaceable item in the common set, is calculated in equation 4.39 by right-multiplying C_U by the replaceable item quantity vector q_{ϱ}. Equations 4.31 through 4.39 fully characterize the ORU list ω, which can then be passed to the model described in Section 4.2 to evaluate supportability and assess mass and crew time requirements.

As noted earlier, this ability to generate an ORU list is a key novel contribution of this research. Other analysis methodologies start from pre-defined ORU lists (or, in the case of METRIC-based approaches, an ORU list with specific lower level SRUs), limiting their ability to fully explore level of maintenance options. They also do not include the various other supportability strategy decisions and their impacts on ORU characteristics. In contrast, this methodology enables the combination of components into arbitrary replaceable items within the indenture structure, the calculation of the supportability characteristics of these items, and the application of commonality decisions to generate the final ORU list. This approach provides broad flexibility and is a critical enabler of supportability strategy optimization.

4.2 Evaluating Supportability

This section describes the supportability evaluation model, which determines the maintenance logistics mass and expected crew time required for maintenance associated with an ORU list ω in a given mission context. The evaluation consists of three parts. First, a maintenance demand model defines the number of preventative maintenance actions (i.e. scheduled replacements) n_{PMi} required for each ORU i, which is a deterministic number, as well as a random variable N_i characterizing the number of corrective maintenance actions that will be required. The distribution of N_i will be used to determine POS as a function of the number of spares n_i provided for each ORU and the amount of ODM feedstock provided, if ODM is used. Next, this demand model is used to assess maintenance crew time, including both the crew time required for preventative maintenance t_{PM} and the time required for corrective maintenance T_{CM} (a random variable). The model also calculates $POS_T(t_{max})$, the probability that there is not enough crew time available to complete required maintenance. Finally, the demand model is used to determine the number of spares and maintenance items required to achieve the desired POS level, applying discrete optimization to identify the spares allocation that achieves this POS requirement while minimizing mass. The number of spares and maintenance items, along with the installed system mass, are used to determine the total mass m_{tot}.

Some key contributions of this thesis in terms of supportability evaluation include the use of separate stochastic processes to describe spares and crew time demands and the inclusion of crew time POS in spares optimization, which have been mentioned previously. Separation of spares and crew time demand enables more accurate maintenance crew time assessment, as shown in Chapter 5. Inclusion of crew time POS results in a more complete risk assessment, since the result includes risks associated with both spares and crew time. Another key contribution of this research is the ODM model described in Section 4.2.3 (see page 183), which assesses the POS associated with the amount of feedstock provided for ODM manufacturing of spares and thereby enables the inclusion of ODM feedstock in spares allocation optimization. While some previous models (described in Section 2.4) have modeled ODM/ISM impacts by examining the ability to recycle failed parts or use ISRU for maintenance logistics, no previous model has provided a risk-based evaluation of feedstock levels or enable the examination of the impacts of commonality of material on spares logistics.

Finally, the supportability evaluation methodology presented here does not require any Monte Carlo simulation. Other models have also used closed-form solutions for spares POS or expected crew time requirements, but the contribution of this research is in the evaluation of the distribution of crew time – and therefore of crew time POS – via numerical inversion of characteristic functions rather than via Monte Carlo simulation (see Section 4.2.2). This same approach is used to evaluate the distribution of ODM feedstock requirements. The elimination of Monte Carlo simulation from the supportability evaluation process significantly reduces the amount of computational time required to evaluate supportability, which helps facilitate multiobjective optimization and tradespace exploration. A longer evaluation time per strategy would likely preclude large-scale supportability tradespace exploration and optimization.

4.2.1 Maintenance Demand Model

Maintenance demand is driven by two processes. Preventative (i.e. scheduled) maintenance is a deterministic process, based on the life limits of components. Corrective maintenance is stochastic, and includes events where a failed ORU is replaced as well as events where crew action is needed but no spare is used. Preventative maintenance consumes maintenance items and crew time deterministically, while corrective maintenance has an associated distribution of the number of events that are used to calculate the distribution of crew time and spares required. This research uses the term "crew action" to refer to to any instance in which crew time is required, including inspection, cleaning, troubleshooting, and R&R. Not all crew actions result in the changeout of an ORU, but all crew actions require crew time. An ORU R&R event, which occurs when an ORU fails, requires both crew time and a spare part. Thus, ORU replacement events are a subset of crew actions. Models for both processes are described below.

For the purposes of this research, all maintenance actions that result in the replacement of an ORU are assumed to be R&R of a failed ORU, inserting a fresh, identical spare in its place and discarding the failed item. If an ORU is manufactured on demand, then the new spare is manufactured from ODM feedstock when it is needed. Crew time associated with non-R&R events like troubleshooting, resetting, and cleaning activities are captured by the crew action rate, which is separate from the R&R demand process.

Preventative Maintenance

Preventative maintenance is defined for this research as the scheduled replacement of ORUs at regular intervals, defined by their life limit $\ell_{\omega i}$. This type of demand is deterministic, and therefore directly relates to the number of preventative maintenance actions and maintenance items that are required for each ORU. Specifically, for each ORU i, the number of preventative maintenance actions required n_{PMi} is

$$n_{PMi} = \left\lfloor \frac{\tau}{\ell_{\omega i}} \right\rfloor q_{\omega i} \tag{4.40}$$

where $\lfloor \cdot \rfloor$ is the floor operator (i.e. rounding down to the nearest integer), τ is the mission endurance, $\ell_{\omega i}$ is the ORU life limit, and $q_{\omega i}$ is the quantity of that ORU in the system. The vector providing the number of preventative maintenance actions for each ORU is

$$n_{PM} = \left\lfloor \tau \circ \ell_{\omega}^{\circ -1} \right\rfloor \circ q_{\omega} \tag{4.41}$$

where the floor operator is applied in an element-wise fashion. Each preventative maintenance action consumes both crew time and maintenance items, thereby impacting both crew time and mass.

Note that this model only accounts for deterministically scheduled maintenance. Another type of preventative maintenance that could be implemented would be one that responds to prognostics from an integrated vehicle health management system. However, assessment of this type of maintenance requires modeling of the sensing and diagnostic capability associated with a particular system architecture, which is beyond the scope of the current research. Therefore, analysis of this type of preventative maintenance is left to future work (described in Chapter 9).

Corrective Maintenance

Corrective maintenance is defined as the replacement of an ORU in response to a random failure, which consumes spares and crew time, or a crew action in response to some anomaly within an ORU that consumes crew time but does not result in a replacement of that ORU. Both of these processes include both internal ORU failures (i.e. ones captured by the failure and crew action rates of the components within the ORU) as well as induced failures, which are captured by the K-factors. In this research, K-factors have already been incorporated into the expected value and variance of the failure rate and crew action rate for each ORU via the process described in Section 4.1. Since these demands are stochastic, corrective maintenance demand is represented by the random variables N_i, representing the number of corrective maintenance replacements required for ORU i, and N_{Ci}, representing the number of crew time actions required. The POS associated with each ORU i when n_i spares are provided is given by the CDF $F_{N_i}(n_i)$. Given the distribution of N_i for each ORU i, spares are allocated to achieve POS requirements while minimizing mass. To determine the distribution of N_i, this research applies a version of the CFR model, modified to include with uncertain failure rates. The model is first described for deterministic failure rates, then expanded to account for epistemic uncertainty.

The CFR model is a common first-order model for random failures that is typically applied for spares planning purposes [126]. Under this model, each component is assumed to have a constant failure rate λ. The number of failures for an item with q instances that occurs in a time period τ follows a Poisson distribution with parameter equal to the expected number of failures, (i.e. $q\tau\lambda$).

$$N \sim \text{Poisson}(q\tau\lambda) \tag{4.42}$$

Thus, given a known failure rate, POS is given by the Poisson CDF evaluated at the provided number of spares [12, 86, 126].

$$POS(n) = \sum_{k=0}^{n} e^{-q\tau\lambda} \left(\frac{(q\tau\lambda)^k}{k!} \right) \tag{4.43}$$

When epistemic uncertainty is present, the deterministic failure rate λ is replaced by the lognormally-distributed random variable $\Lambda \sim \text{Logn}(\mu, \sigma)$. The lognormal distribution is a common model for failure rate uncertainty For example, it is used by the ISS Program to model ORU failure rates and perform Bayesian updates [13, 94, 96]. The Probability Distribution Function (PDF) representing failure rate uncertainty is

$$f_\Lambda(\lambda) = \frac{1}{\sqrt{2\pi}\sigma\lambda} e^{-\frac{(\ln\lambda-\mu)^2}{2\sigma^2}} \tag{4.44}$$

where the parameters $\mu \in \mathbb{R}$ and $\sigma > 0$ are known as the shape and scale parameters, respectively, and the domain is $\lambda \geq 0$ [13, 31, 86, 94, 96, 127].

An alternate representation, commonly used in reliability analysis, parametrizes the lognormal distribution using a mean $\bar{\lambda}$ and error factor ε rather than the shape and scale parameters. These are the parameters defined for each component in the system description $\mathcal{S}$. The error factor, a measure of the width of the distribution, is defined as the ratio of the 95th and 50th percentiles or, equivalently, the ratio of the 50th and 5th percentiles. Therefore, $\varepsilon \geq 1$ by definition. The mean and error factor are a function of the shape and scale parameters for a lognormal distribution [31, 96]:

$$\bar{\lambda} = e^{\mu+\frac{\sigma^2}{2}} \tag{4.45}$$

$$\varepsilon = e^{1.645\sigma} \tag{4.46}$$

Rearranging these equations enables calculation of μ and σ as a function of the mean and error factor.

$$\sigma = \frac{\ln\varepsilon}{1.645} \tag{4.47}$$

$$\mu = \ln\bar{\lambda} - \frac{\sigma^2}{2} \tag{4.48}$$

Therefore, the expected value and variance of the lognormal distribution $\Lambda \sim \text{Logn}(\mu,\ \sigma)$, as a function of either the shape and scale parameters or the mean and error factor, are [86]:

$$\mathrm{E}[\Lambda] = \qquad e^{\mu+\frac{\sigma^2}{2}} \qquad = \bar{\lambda} \tag{4.49}$$

$$\mathrm{Var}[\Lambda] = e^{2\mu+\sigma^2}\left(e^{\sigma^2} - 1\right) = \bar{\lambda}^2\left(e^{\left(\frac{\ln \varepsilon}{1.645}\right)^2} - 1\right) \tag{4.50}$$

When the failure rate is itself a random variable, the number of failures N follows a mixed Poisson distribution – that is, a Poisson distribution with a random variable parameter $q\tau\Lambda$ [86, 87]. When Λ is lognormal, this is called a Poisson-lognormal distribution.

$$N \sim \mathrm{Poisson}(q\tau\Lambda) \wedge \Lambda \sim \mathrm{Logn}(\mu,\ \sigma) \tag{4.51}$$

Note that a lognormal distribution Λ multiplied by a constant factor τ is still lognormal, $q\tau\Lambda \sim \mathrm{Logn}(\mu + \ln q\tau, \sigma)$.

Unfortunately, there is no closed-form representation of probabilities associated with a Poisson-lognormal distribution [128]. In addition, these distributions have no closed-form characteristic function. Since characteristic functions are a key element in the evaluation of ODM feedstock mass requirements and the distribution of corrective maintenance crew time (described in Sections 4.2.2 and 4.2.3), this is a significant limitation. Evaluation of POS using this model would require numerical approximation or Monte Carlo analysis, which can require significant amounts of computational time in order to achieve precise (and consistent) results.

However, lognormal failure rate distributions can be approximated approximated by gamma distributions to enable closed-form calculations of both probabilities and characteristic function values. The gamma distribution is also sometimes used to represent failure rate uncertainty. For example, it is the conjugate prior for Bayesian analysis of the parameter of a Poisson process. While Bayesian updates using the lognormal distribution require numerical evaluation, the gamma distribution enables direct calculation of the posterior distribution [96]. The accuracy of this gamma approximation, as compared to the lognormal model, is examined in greater detail in Chapter 5.

A gamma-distributed random variable $\Upsilon \sim \text{Gamma}(\alpha,\ \beta)$ has the PDF

$$f_{\Upsilon}(\upsilon) = \frac{\upsilon^{\alpha-1}\beta^{\alpha}e^{-\beta\upsilon}}{\Gamma(\alpha)} \tag{4.52}$$

where the parameters $\alpha \geq 0$ and $\beta \geq 0$ are the shape and scale parameters, respectively, and $\Gamma(\cdot)$ is the gamma function. When a gamma-distributed parameter is used in a mixed Poisson distribution, the result is a gamma-Poisson distribution, which is equivalent to a negative binomial whose parameters are derived from those of the gamma distribution [83, 87, 129]. That is,

$$N \sim \text{Poisson}(\Upsilon) \wedge \Upsilon \sim \text{Gamma}(\alpha,\ \beta) \implies N \sim \text{NB}\left(\alpha,\ \hat{p}\right) \tag{4.53}$$

where

$$\hat{p} = \frac{\beta}{1+\beta} \tag{4.54}$$

Therefore, POS for each ORU can be calculated using the negative binomial CDF, presented in equation 4.55, and the negative binomial characteristic function can be used to evaluate crew time and ODM feedstock distributions, described later [86]:

$$POS_i(n_i) = \sum_{k=0}^{n_i} \binom{k+\alpha_i-1}{k} \hat{p}^{\alpha_i}(1-\hat{p})^k \tag{4.55}$$

In this application, each ORU is characterized by quantity $q_{\omega i}$ and $\text{E}[\Lambda_{\omega i}]$ and $\text{Var}[\Lambda_{\omega i}]$, which are the expected value and variance of the ORU failure rate, respectively, taking into account the K-factors, duty cycles, and quantities of all of the components within the ORU. The parameters α_i and β_i of the gamma-distributed approximation Υ_i of the Poisson parameter $q_{\omega i}\tau\Lambda_{\omega i}$ for each ORU are determined by matching the expected value and variance of $q_{\omega i}\tau\Lambda_{\omega i}$. Noting that, for any random variable X and constant k, $\text{E}[kX] = k\,\text{E}[X]$ and $\text{Var}[kX] = k^2\,\text{Var}[X]$ [86], these values are:

$$\mathrm{E}[\Upsilon_i] = \frac{\alpha_i}{\beta_i} = q_{\omega i}\tau\,\mathrm{E}[\Lambda_{\omega i}] \tag{4.56}$$

$$\mathrm{Var}[\Upsilon_i] = \frac{\alpha_i}{\beta_i^2} = (q_{\omega i}\tau)^2\,\mathrm{Var}[\Lambda_{\omega i}] \tag{4.57}$$

Therefore,

$$\alpha_i = \frac{\mathrm{E}[\Lambda_{\omega i}]^2}{\mathrm{Var}[\Lambda_{\omega i}]} \tag{4.58}$$

$$\beta_i = \frac{\mathrm{E}[\Lambda_{\omega i}]}{q_{\omega i}\tau\,\mathrm{Var}[\Lambda_{\omega i}]} \tag{4.59}$$

and

$$\hat{p} = \left(\frac{\mathrm{E}[\Lambda_{\omega i}]}{q_{\omega i}\tau\,\mathrm{Var}[\Lambda_{\omega i}]}\right)\left(\frac{1}{1+\frac{\mathrm{E}[\Lambda_{\omega i}]}{q_{\omega i}\tau\,\mathrm{Var}[\Lambda_{\omega i}]}}\right) = \frac{\mathrm{E}[\Lambda_{\omega i}]}{q_{\omega i}\tau\,\mathrm{Var}[\Lambda_{\omega i}]+\mathrm{E}[\Lambda_{\omega i}]} \tag{4.60}$$

In vector form, α and $\hat{p}$ are calculated as

$$\alpha = \mathrm{E}[\Lambda_\omega]^{\circ 2} \circ \mathrm{Var}[\Lambda_\omega]^{\circ -1} \tag{4.61}$$

$$\hat{p} = \mathrm{E}[\Lambda_\omega] \circ (\tau \circ q_\omega \circ \mathrm{Var}[\Lambda_\omega] + \mathrm{E}[\Lambda_\omega])^{\circ -1} \tag{4.62}$$

The distribution of the number of crew actions required N_{Ci} is also modeled using the negative binomial approximation, but uses the crew action rate rather than failure rate. where the vectors giving the parameters α_C and $\hat{p}_C$ are

$$\alpha_C = \mathrm{E}[\Lambda_{C\omega}]^{\circ 2} \circ \mathrm{Var}[\Lambda_{C\omega}]^{\circ -1} \tag{4.63}$$

$$\hat{p}_C = \mathrm{E}[\Lambda_{C\omega}] \circ (\tau \circ q_\omega \circ \mathrm{Var}[\Lambda_{C\omega}] + \mathrm{E}[\Lambda_{C\omega}])^{\circ -1} \tag{4.64}$$

However, unlike spares, crew time is an undifferentiated resource, meaning it is not specialized to a particular type of failure. The distribution of interest is therefore not the number of crew actions on an ORU-by-ORU basis, but rather the total amount of time required, which is the sum of the crew time required for each ORU. Similarly, when ODM is used the value of interest is the sum of the amount of feedstock mass required for each ORU that is manufactured on demand. The distribution of total crew time or feedstock mass required is a convolution of the distributions associated with each individual source of demand, calculated using characteristic functions.

The characteristic function of a random variable X, defined as $\varphi_X(s) = \mathrm{E}[e^{\hat{\imath}sX}]$ (where $\hat{\imath} = \sqrt{-1}$ is the imaginary unit), is a particularly useful representation of random variables, since it is equivalent to the Fourier transform of the PDF $\hat{f}_X(s)$ with a change of variables [86, 130]:

$$\hat{f}_X(s) = \varphi_X(-2\pi s) \tag{4.65}$$

Therefore, characteristic functions map convolution to multiplication. Specifically, the characteristic function of a linear combination of random variables $Y = \sum_i k_i X_i$ is

$$\varphi_Y(s) = \prod_i \varphi_{X_i}(k_i s) \tag{4.66}$$

The characteristic function $\varphi_N(s)$ for a negative binomial random variable $N \sim \mathrm{NB}(\alpha,\ \hat{p})$ is [86]

$$\varphi_N(s) = \left(\frac{\hat{p}}{1 - (1 - \hat{p})e^{\hat{\imath}s}} \right)^{\alpha} \tag{4.67}$$

This is used to model both the number of ORU replacements N_i and the number of crew actions N_{Ci} for each ORU (in which case α and $\hat{p}$ are replaced by α_C and $\hat{p}_C$).

The Probability Mass Function (PMF) of a discrete random variable can be recovered from its characteristic function via numerical Fourier Transform inversion, using the technique described by Warr [130]. This approach is used to numerically assess both the distribution of the amount of crew time required for maintenance and the distribution of the amount of feedstock required for ODM, as described in Sections 4.2.2 and 4.2.3, respectively.

The above set of equations describe POS as a function of the number of spares provided for each ORU on an individual basis. The POS of the entire system is the product of the POS for each ORU.

$$POS(n) = \prod_{i \in \omega} POS_i(n_i) \tag{4.68}$$

Equations 4.55 and 4.68 therefore provide a model for the relationship between the spares allocation n and the associated POS, which is used in Section 4.2.3 to determine the minimum mass of spares required to achieve the POS target. Equation 4.67 provides a method for calculating POS for both ODM feedstock and crew time. These equations are used to calculate crew time and mass requirements using the approach described in the following sections.

4.2.2 Crew Time Model

This section describes the crew time model, which assesses the expected crew time required for maintenance E$[T]$ – including both time required for preventative maintenance t_{PM} and time required for corrective maintenance T_{CM} – as well as the probability that the maximum amount of crew time available will be sufficient for all required maintenance actions.

Expected Maintenance Crew Time

The total expected maintenance crew time is the sum of preventative maintenance crew time t_{PM} and the expected corrective maintenance crew time E$[T_{CM}]$. Crew time required for preventative maintenance t_{PM} is the sum of the preventative maintenance crew time associated with each ORU, which is simply the amount of crew time per maintenance action $t_{\omega i}$ multiplied by the number of preventative maintenance actions n_{PMi} for that ORU

$$t_{PM} = \sum_{i \in \omega} t_{\omega i} n_{PMi} = t_\omega^T n_{PM} \tag{4.69}$$

where n_{PMi} is given by equation 4.40.

Corrective maintenance crew time T_{CM} is a random variable driven by the number of crew actions N_{Ci} and the crew time associated with each ORU. Specifically, the total time required for corrective maintenance is equal to the sum of the corrective maintenance crew time required for each ORU.

$$T_{CM} = \sum_{i \in \omega} t_{\omega i} N_{Ci} \tag{4.70}$$

The full distribution is required to evaluate the probability of sufficient crew time, described in the following section. However, the expected value can be calculated in a relatively simple manner by examining the expected number of crew actions for each ORU:

$$\mathrm{E}[T_{CM}] = \sum_{i \in \omega} t_{\omega i} \, \mathrm{E}[N_{Ci}] \tag{4.71}$$

The expected number of crew actions is simply the expected value of the crew action rate $\mathrm{E}[\Lambda_{C\omega i}]$ multiplied by the quantity $q_{\omega i}$ and mission endurance τ. Therefore, expected corrective maintenance crew time is

$$\mathrm{E}[T_{CM}] = \sum_{i \in \omega} t_{\omega i} q_{\omega i} \tau \, \mathrm{E}[\Lambda_{C\omega i}] = t_\omega^T \left(q_\omega \circ \tau \, \mathrm{E}[\Lambda_{C\omega}] \right) \tag{4.72}$$

Combining the contributions of preventative and corrective maintenance, the total expected maintenance crew time is

$$\begin{aligned} \mathrm{E}[T] &= t_{PM} + \mathrm{E}[T_{CM}] && (4.73) \\ &= t_\omega^T \left(n_{PM} + q_\omega \circ \tau \, \mathrm{E}[\Lambda_{C\omega}] \right) && (4.74) \end{aligned}$$

Probability of Sufficient Crew Time

The probability of sufficient crew time $POS_T(t_{max})$ is the CDF of the distribution of total crew time required for maintenance T evaluated at the maximum crew time available for maintenance t_{max}, where T is the sum of the preventative and corrective maintenance crew time associated with each ORU.

$$T = t_{PM} + T_{CM} \tag{4.75}$$

Since t_{PM} is deterministic, POS_T can be calculated using the distribution of T_{CM}

$$POS_T(t_{max}) = POS_{T_{CM}}(t_{max} - t_{PM}) \tag{4.76}$$

where $POS_{T_{CM}}(t)$ is the POS for corrective maintenance crew time, which is equal to zero for any $t < 0$. $POS_{T_{CM}}$ is calculated using the CDF of T_{CM}, as defined in equation 4.70. This convolution is most effectively evaluated using characteristic functions, where the characteristic function for the number of crew actions N_{Ci} required for a given ORU is calculated using equation 4.67 with α and $\hat{p}$ replaced by α_{Ci} and $\hat{p}_{Ci}$.

Warr [130] presents a numerical approach for recovering the PMF of a discrete lattice distribution[c] from its characteristic function using an Inverse Discrete Fourier Transform (IDFT). The overall approach consists of sampling the characteristic function at a set of points $\hat{s}$, known as the sampling vector, and applying the IDFT to the resulting samples. For an integer random variable,[d] the sampling vector $\hat{s}$ (consisting of $n_{\hat{s}}$ samples) is defined as

$$\hat{s} = \left[\frac{-2\pi k}{n_{\hat{s}}} \;\middle|\; k = 0,\ 1,\ 2,\ ...,\ n_{\hat{s}} - 1 \right] \tag{4.77}$$

The estimate for the PMF, denoted $f_n \approx P(N = n)$ is calculated by applying the IDFT (denoted $\mathcal{F}^{-1}$) to the characteristic function sampled at the points indicated in equation 4.77 [130].

$$f_n = \mathcal{F}^{-1}\left(\varphi_N\left(\hat{s}\right)\right) \tag{4.78}$$

[c] A lattice distribution is one whose support is a series of evenly-spaced discrete steps, such as the negative binomial distributions examined here [130].

[d] Warr [130] describes a sampling approach that uses non-integer step sizes. However, for simplicity this research deals directly with the distribution of the number of discrete units for pre-discretized amounts, and therefore only requires assessment of integer random variables.

A cumulative sum of the PMF yields the CDF.

To use this approach to assess the probability of sufficient crew time, the crew time distribution is discretized onto a lattice of discrete time steps of equal size using the discretization factor Δt, which is defined as part of the modeling parameters $\mathcal{P}$. The distribution of the total number of time steps required is

$$N_{\Delta t} = \sum_{i \in \omega} \left\lceil \frac{t_{\omega i}}{\Delta t} \right\rceil N_{Ci} \tag{4.79}$$

where $\lceil \cdot \rceil$ is the ceiling operator (i.e. rounding up to the nearest integer). That is, $\left\lceil \frac{t_{\omega i}}{\Delta t} \right\rceil$ is the number of time steps associated with each crew action for ORU i, and N_{Ci} is the random variable representing the number of crew actions that occur during the mission. The characteristic function of $N_{\Delta t}$ is

$$\varphi_{N_{\Delta t}}(s) = \prod_{i \in \omega} \varphi_{N_{Ci}} \left(\left\lceil \frac{t_{\omega i}}{\Delta t} \right\rceil s \right) \tag{4.80}$$

where the characteristic function of the number of crew actions required for each ORU $\varphi_{N_{Ci}}(s)$ is given by equation 4.67.

The level of error in the PMF approximation is characterized by a bound on pointwise error, denoted ϵ_{pt} and defined as

$$\epsilon_{pt} = |f_n - P(N = n)| \tag{4.81}$$

Pointwise error is driven by of the number of samples $n_{\hat{s}}$, and can be bounded in a number of different ways, which are described below. Since the CDF is generated from a cumulative sum, the error in the evaluation of a given point k in the CDF is equal to $k\epsilon_{pt}$.

The quantity of interest, $POS_{T_{CM}}(t_{max} - t_{PM})$, is the probability that the maximum amount of crew time available for maintenance is sufficient, given preventative and corrective maintenance demands. This is approximated by the CDF of $N_{\Delta t}$ evaluated at the number of timesteps associated with $t_{max} - t_{PM}$:

$$POS_{T_{CM}}(t_{max} - t_{PM}) \approx F_{N_{\Delta t}}\left(\left\lceil \frac{t_{max} - t_{PM}}{\Delta t} \right\rceil\right) \tag{4.82}$$

Since the CDF is evaluated using a cumulative sum of the PMF (as described above), the error in this calculation is equal to $\epsilon_{pt}\left\lceil \frac{t_{max}-t_{PM}}{\Delta t} \right\rceil$. Given a maximum error in the overall calculation ϵ_{max} (specified in the modeling parameters $\mathcal{P}$), the required maximum pointwise error is

$$\epsilon_{pt,max} = \frac{\epsilon_{max}}{\left\lceil \frac{t_{max}-t_{PM}}{\Delta t} \right\rceil} \tag{4.83}$$

This maximum pointwise error is used to determine the required number of samples.

For nonnegative integer random variables, this pointwise error is bounded by the probability that $N \geq n_{\hat{s}}$ [130]:

$$\epsilon_{pt} \leq P(N \geq n_{\hat{s}}) \tag{4.84}$$

While the right side of equation 4.84 is not typically known *a priori*, it can be estimated based on the expected value and variance of N using Markov's inequality or Cantelli's inequality, which state, respectively, that

$$P(N \geq n) \leq \frac{\mathrm{E}[N]}{n} \tag{4.85}$$

$$P(N \geq k\sqrt{\mathrm{Var}[N]} + \mathrm{E}[N]) \leq \frac{1}{1+k^2} \qquad \text{for } k > 0 \tag{4.86}$$

These inequalities typically provide loose bounds at high probabilities, and therefore result in the selection of $n_{\hat{s}}$ that are much higher than would actually be required to achieve the desired error bounds [130]. In practice, these excessive sample sizes can result in significantly increased computational overhead and slower analysis.

However, a much smaller required sample size can be used when additional information is known about the distribution of N. Specifically, for distributions for which

$$P\left(N = a + \frac{kb}{2}\right) \geq P\left(N = a + \frac{(k+1)b}{2}\right) \tag{4.87}$$

for some nonnegative integers a and $\frac{b}{2}$, where $a < \frac{b}{2}$, and positive integer k, the use of the above technique with $n_{\hat{s}} = b$ will bound the pointwise error as

$$\epsilon_{pt} \leq f_{a+\frac{b}{2}} \tag{4.88}$$

Distributions that meet the condition described in equation 4.87 are called "eventually periodic nonincreasing" distributions, and are cases where, after some point a, the PMF is nonincreasing between steps of length $\frac{b}{2}$. Additional details and a proof are provided by Warr [130].

Inspection of representative PMF results for this research (calculated both using this method and via Monte Carlo simulation), indicate that crew time and ODM feedstock mass distributions are eventually periodic nonincreasing. This makes sense intuitively, since in this application the distributions of interest are the results of linear combinations of negative binomial distributions, which are unimodal and therefore are nonincreasing after a certain point. Based on these observations, the period of nonincreasing steps in the PMF is typically smaller than the point a at which the distribution becomes periodic nonincreasing, and therefore b is selected to be as small as possible while still fullfilling the condition $a < \frac{b}{2}$, namely $b = 2a + 2$. Therefore, given a number of samples $n_{\hat{s}}$, the pointwise error is bounded by $f_{n_{\hat{s}}-1}$ if the condition in equation 4.87 is met for $a = \frac{n_{\hat{s}}-2}{2}$ and $b = n_{\hat{s}}$.

To determine the required number of samples, this research follows an adapted version of the iterative procedure described by Warr [130]. Specifically, an initial guess for $n_{\hat{s}}$ is made, the inversion procedure is carried out, and the resulting error bound is checked using equation 4.88. If the error bound is not below the required value (i.e. if $f_{n_{\hat{s}}-1} > \epsilon_{pt,max}$), the number of samples is doubled and the calculation performed again. The process is repeated until the desired error bound is met (i.e. $f_{n_{\hat{s}}-1} \leq \epsilon_{pt,max}$). For this application, the initial guess is set to the number of time steps required to ensure that the 95$^{\text{th}}$ is reached, based on Cantelli's inequality. In practice, this approach has yielded a required number of samples significantly less than the one calculated using either Markov's inequality or Cantelli's inequality along with equation 4.84 – in some cases by multiple orders of magnitude – thereby significantly speeding up the analysis process.

It is important to note that the error bound described above only applies to the numerical inversion itself, and does not include discretization error or error related to machine precision. The overall accuracy of this approach is examined in Chapter 5, where results calculated using characteristic functions are compared to results from Monte Carlo simulations.

4.2.3 Mass Model

The total mass m_{tot} is the sum of the installed system mass m_S, preventative maintenance mass m_{PM}, and corrective maintenance mass m_{CM}, as described by equation 3.11. Each contribution is calculated independently, as described in the sections below.

Installed System Mass

Installed system mass m_S is the mass of all ORUs that must be installed for the system to function, as well as the mass of the ODM system if one is present. Put another way, this is the mass of the ORUs that are installed and operating at the beginning of the mission, rather than stored as logistics, and the systems required to support ODM if needed.

$$m_S = \xi_s m_{odm} + \sum_{i \in \omega} m_{\omega i} q_{\omega i} = \xi_s m_{odm} + m_\omega^T q_\omega \tag{4.89}$$

Preventative Maintenance Mass

Preventative maintenance mass m_{PM} is the total mass of all maintenance items and ODM feedstock allocated to cover preventative maintenance actions, meaning scheduled replacements of ORUs when they reach the end of their life limit.

$$m_{PM} = \sum_{i \in \omega} m_{\omega i} n_{PMi} = m_{\omega}^T n_{PM} \tag{4.90}$$

Here the number of scheduled replacements for each ORU n_{PM} is defined by equation 4.41. Note that preventative maintenance mass includes both spare parts and ODM feedstock. ODM does not impact the mass required for preventative maintenance (i.e. there is no commonality of material effect), the total mass can still simply be calculated using equation 4.90. The only impact of ODM here is to adjust the mass of each ORU using γ_{odm}, which was accounted for in the calculation of m_{ω}. The preventative mass associated with ODM feedstock and discrete spares can be identified separately by only including ORUs which do or do not use ODM, respectively.

Corrective Maintenance Mass

Corrective maintenance mass m_{CM} is the total mass of all spares and ODM feedstock allocated to cover random failures. Whereas installed system mass and preventative maintenance mass are both deterministic, demand for corrective maintenance mass is stochastic. Therefore, m_{CM} is a function not only of system and mission characteristics, but also of the desired level of of risk mitigation, captured by POS. Each allocation of spare parts and ODM feedstock results in a particular POS value. There are many allocations that will fulfill the POS requirement described in equation 3.33; however, only one will do so while also minimizing mass. This optimal spares allocation is found via discrete optimization, combining marginal analysis [12, 90] with a branch-and-bound search strategy [131].

A spares allocation n is defined as the allocation of a certain number of spares n_i to each ORU that does not use ODM and/or a number of ODM feedstock units n_{odm} to cover the set of spares that do use ODM (if any exist).

$$n = \begin{cases} \{n_i \; \forall \; i \in \omega\} & \text{if } \nexists \; \xi_{\omega i} = 1 \\ \{n_i \; \forall \; i \in \omega \mid \xi_{\omega i} = 0\} \cup \{n_{odm}\} & \text{else} \end{cases} \tag{4.91}$$

That is, if no ODM is used, n is simply an assignment of a number of spares n_i to each ORU. However, if ODM is used for any spares, then spares are assigned only to ORU that do not use ODM; the rest are covered by ODM feedstock, which is assigned in discrete units as a kind of spare. For brevity, the term "spare" is used to refer to spare ORUs or spare units of ODM feedstock interchangeably. Similarly, the enumeration of spares $i \in n$ includes the ODM feedstock allocation n_{odm}, if one exists.

Each spare has a mass associated with it. For ORU spares, this is simply $m_{\omega i}$, the mass of the associated ORU. For ODM feedstock, the mass is equal to the mass discretization factor Δm. The mass of any spare, whether an ORU or ODM feedstock, is denoted $\hat{m}_i$. Given an allocation n, the total corrective maintenance mass is

$$m_{CM} = \sum_{i \in n} \hat{m}_i n_i \tag{4.92}$$

Each spare also has a POS function, which describes the probability that a given allocation n_i provides a sufficient amount of that resource. For spare ORUs, POS is given by the negative binomial CDF, described by the maintenance demand model in Section 4.2.1 (equation 4.55). The POS calculation for ODM feedstock is more complex, and is described in the next section. The total POS of a given allocation n is the product of the POS for each spare in that allocation.

$$POS_S(n) = \prod_{i \in n} POS_i(n_i) \tag{4.93}$$

For notation purposes, the set of POS functions for each spare is denoted $POS = \{POS_i(n_i) \; \forall \; i \in n\}$.

Calculating the Probability of Sufficient Manufacturing Feedstock

The total amount of ODM feedstock required for corrective maintenance is a random variable, equal to the sum of the feedstock required for each ORU that is covered by ODM. The technique used to calculated probabilities associated with various levels of ODM feedstock is similar to that used to calculate the distribution of maintenance crew time, described in Section 4.2.2. The (random) number of units of feedstock required is

$$N_{\Delta m} = \sum_{i \in \omega | \xi_{\omega i}} \left\lceil \frac{m_{\omega i}}{\Delta m} \right\rceil N_i \tag{4.94}$$

and the associated characteristic function is

$$\varphi_{N_{\Delta m}}(s) = \prod_{i \in \omega | \xi_{\omega i}} \varphi_{N_i} \left(\left\lceil \frac{m_{\omega i}}{\Delta m} \right\rceil s \right) \tag{4.95}$$

where $\varphi_{N_i}(s)$ is given by equation 4.67.

The distribution of $N_{\Delta m}$ is estimated using the same IDFT technique as was applied to calculate crew time in Section 4.2.2, using the same maximum error ϵ_{max}. However, in this case the point at which the CDF will be evaluated is not known *a priori*. Therefore, this research bases the selection of the number of samples $n_{\hat{s}}$ on the amount of error in the calculation of $F_{N_{\Delta m}}(n_{POS_R})$, where n_{POS_R} is the minimum number of units of mass provided such that the POS for ODM feedstock is known to be greater than the requirement, based on all available information.

$$n_{POS_R} = \min\{n \mid P(N_{\Delta m} \leq n) \geq POS_R\} \tag{4.96}$$

As the CDF of $N_{\Delta m}$ is not known until it is calculated, the smaller of either Markov's inequality (equation 4.85) or Cantelli's inequality (equation 4.86) are used to set n_{POS_R} to the lowest known value, based only on the expected value and variance of $N_{\Delta m}$, for which $P(N_{\Delta m} \geq n_{POS_R}) \geq POS_R$:

$$n_{POS_R} = \min \left(\left\lceil \frac{\mathrm{E}[N_{\Delta m}]}{1 - POS_R} \right\rceil, \left\lceil \sqrt{\frac{\mathrm{Var}[N_{\Delta m}]}{1 - POS_R}} + \mathrm{E}[N_{\Delta m}] \right\rceil \right) \tag{4.97}$$

where

$$\mathrm{E}[N_{\Delta m}] = \sum_{i \in \omega | \xi_{\omega i}} \left\lceil \frac{m_{\omega i}}{\Delta m} \right\rceil q_{\omega i} \tau \, \mathrm{E}[\Lambda_{\omega i}] \tag{4.98}$$

$$\mathrm{Var}[N_{\Delta m}] = \sum_{i \in \omega | \xi_{\omega i}} \left(\left\lceil \frac{m_{\omega i}}{\Delta m} \right\rceil q_{\omega i} \tau \right)^2 \mathrm{Var}[\Lambda_{\omega i}] \tag{4.99}$$

Note that these summations include only the ORUs for which ODM is used. Once the evaluation point n_{POS_R} is determined, the required number of samples $n_{\hat{s}}$ is determined using the same iterative approach used in the crew time analysis (see Section 4.2.2).

The number of units of ODM feedstock that will be required may be greater than n_{POS_R}, and therefore this approach is not guaranteed to maintain the desired error bounds in all applications. However, error in the CDF grows proportionally with ϵ_{pt} (which is typically much smaller than ϵ_{max}) as values above n_{POS_R} are examined. In addition, since Markov's inequality and Cantelli's inequality typically produce very loose bounds, the value of n_{POS_R} determined using 4.96 is a conservative estimate. Finally, since the achieved bound on ϵ_{pt} is known after inversion (via equation 4.88), the achieved error bound on the final result can be checked after the number of ODM feedstock units in the spares allocation is determined.

Spares Allocation Optimization

The optimal spares allocation n^* is the solution to the spares allocation problem defined by equation 4.100. Specifically, the optimal allocation provides spares for each ORU and/or units of ODM feedstock such that the overall POS (accounting for both spares and crew time) is above the required POS and total spares mass is minimized.

$$n^* = \min_{n} \quad \sum_{i \in n} \hat{m}_i n_i \tag{4.100a}$$

$$\text{s.t.} \quad POS_S(n) POS_T(t_{max}) \geq POS_R \tag{4.100b}$$

This discrete optimization problem is solved using a combination of marginal analysis [12, 90] and a branch and bound strategy [131].

Marginal analysis is an algorithm for identifying Pareto-optimal solutions to discrete optimization problems that maximize concave, strictly increasing, separable objective functions and minimize linear cost functions. Starting from any Pareto-optimal solution (typically a corner point, such as the minimum allocation), the algorithm iteratively adds the most valuable item – that is, the item that has the greatest increase in the objective function per increase in cost function – until some stopping condition is reached [12, 90].

In this application, marginal analysis is used to identify the Pareto frontier of spares allocations minimizing mass and maximizing POS. In order to enable separability, a log-transform is applied to the POS constraint described in equation 4.100b, and the terms are rearranged to isolate the POS associated with the spares allocation.

$$\ln\left(POS_S(n)POS_T(t_{max})\right) \geq \ln POS_R \tag{4.101}$$

$$\ln POS_S(n) \geq \ln POS_R - \ln POS_T(t_{max}) \tag{4.102}$$

Inserting the definition of $POS_S(n)$ from equation 4.93,

$$\ln\left(\prod_{i\in n} POS_i(n_i)\right) \geq \ln POS_R - \ln POS_T(t_{max}) \tag{4.103}$$

$$\sum_{i\in n} \ln POS_i(n_i) \geq \ln POS_R - \ln POS_T(t_{max}) \tag{4.104}$$

The right side of equation 4.104 is a fixed value, based on the input parameter POS_R and the result of the crew time POS calculation described in Section 4.2.2. The left side is the POS objective to be maximized when marginal analysis is used to identify Pareto-optimal spares allocations; the right side is the stopping condition. That is, the algorithm stops once an allocation that satisfies equation 4.104 is found.

The logarithm of a function is maximized at the same point as the function itself, and therefore this log-transform does not impact the results of optimization [12]. Moreover, the negative binomial distribution is log-concave, meaning that the logarithm of its CDF is concave [132, 133]. In addition, since the POS function is a CDF, it is strictly increasing. Therefore, marginal analysis is guaranteed to identify only Pareto-optimal solutions to the spares allocation problem when only discrete spares are used.

When ODM is present, the POS associated with a given number of units of feedstock is determined using the numerical characteristic function inversion technique described in the previous section. The CDF resulting from this procedure is not necessarily log-concave, and as a result $\ln POS(n)$ for ODM may not be concave. However, the marginal analysis procedure can be expanded to non-concave objective functions by only considering concave points of that function, or points that make up what Fox calls the "least concave majorant" of the objective function [80, 83, 90]. These points could alternatively be considered to be the corners of the upper edge of a concave hull surrounding all of the points. In this application, when feedstock is added to the allocation, it is not necessarily added one unit at a time; instead, a number of units are added such that the resulting point is a corner of the least concave majorant of $\ln POS(n)$, as represented in Figure 4.2. These points can be identified in a variety of ways using any algorithm that can rapidly identify the convex hull surrounding a set of points, such as the Graham scan algorithm [134], or Andrew's monotone chain algorithm [135, 136], which takes advantage of the fact that $\ln POS(n)$ is an ordered, nondecreasing sequence. When only these points are used, marginal analysis is also guaranteed to identify only Pareto-optimal solutions.

For this application, an additional set of constraints are added to the marginal analysis problem, bounding the minimum and maximum spares allocations to n^{LB} and n^{UB}, respectively. That is, each entry n_i^{LB} is the minimum spares allocation for entry i, and n_i^{UB} is the maximum allocation. The problem of finding Pareto-optimal spares allocations within these bounds, called the bounded marginal analysis problem, is presented in equation 4.105.

$$\min_{n} \quad \left(-\sum_{i \in n} \ln POS_i(n_i),\ \sum_{i \in n} \hat{m}_i n_i \right) \tag{4.105a}$$

$$\text{s.t.} \quad n_i \geq n_i^{LB} \qquad \forall\ i \in n \tag{4.105b}$$

$$n_i \leq n_i^{UB} \qquad \forall\ i \in n \tag{4.105c}$$

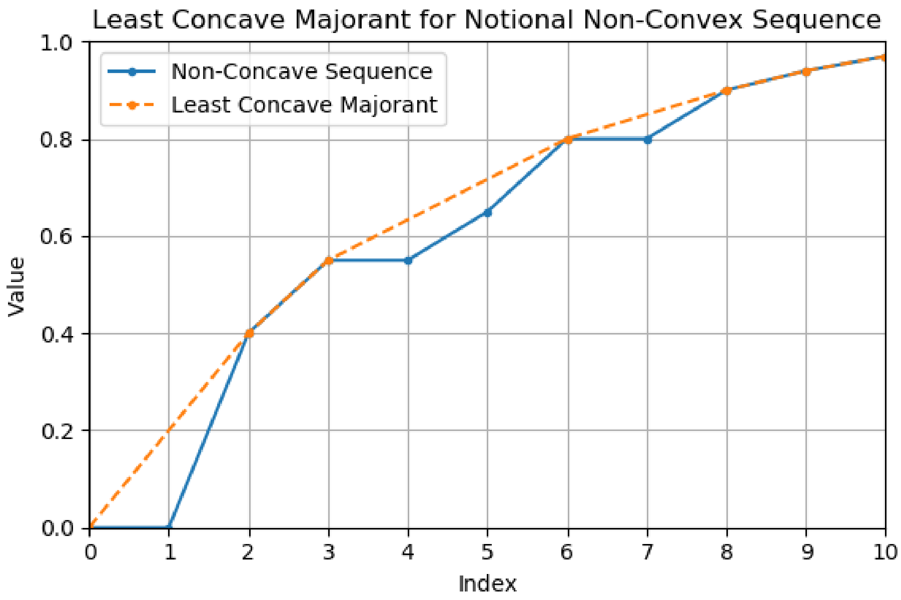

Figure 4.2: Example least concave majorant for a notional non-concave sequence. The original sequence is shown by the blue solid line. The concave sequence, which consists of only the subset of points in the sequence that are also in the least concave majorant, is shown by the orange dotted line. In this case, the concave indices are [0, 2, 3, 6, 8, 9, 10].

The bounded marginal analysis algorithm BMA, shown in algorithm 4.2, uses marginal analysis to identify the first Pareto-optimal solution to 4.105 that is above the POS requirement (i.e. the first solution that satisfies equation 4.104), as well as the infimum mass, which is a lower bound on the spares mass required to achieve the POS requirement under the given constraints (i.e. n_i^{LB} and n_i^{UB}). The algorithm also returns the mass of the allocation and the index i^+ of the last item added, which is used by the branch and bound algorithm. Note that, for simplicity, in this formulation the input POS requirement is assumed to represent the spares POS requirement, which has already been accounted for the POS associated with crew time.

The bounded marginal analysis process starts by initializing the spares allocation at the lower bound and calculating the objective value $\ln POS_S$ and mass m associated with that allocation. Then the marginal analysis procedure begins, and is repeated until the POS requirement is met. The next allocation value for each spare $n_{i,next}$

Algorithm 4.2 BMA solves the bounded marginal analysis problem to identify the first Pareto-optimal spares allocation n above the given POS requirement, its mass m, the infimum mass m_{inf} (which is the lower bound on the spares mass required under the given constraints), and the last spare that was added to the allocation i^+. The inputs are a set of spares masses $\hat{m}$, POS functions $POS = \{POS_i(n_i)\ \forall i \in n\}$, a required POS POS_R, and an allocation lower bound n^{LB} and upper bound n^{UB}. For simplicity, the required POS input to this algorithm is assumed to have already accounted for $POS_T(t_{max})$.

```
BMA($\hat{m}$, $POS$, $POS_R$, $n^{LB}$, $n^{UB}$)
initialize spares allocation $n \leftarrow n^{LB}$
initialize objective value $\ln POS_S \leftarrow \sum_{i \in n} POS_i(n_i)$
initialize mass $m \leftarrow \sum_{i \in n} \hat{m}_i n_i$
while $\ln POS_S < \ln POS_R$
    for each spare $i$
        determine the next allocation value $n_{i,next}$
        if $n_{i,next} > n_i^{UB}$
            $v_i \leftarrow -\infty$
        else
            calculate marginal value $v_i$ using eqn. 4.106
    identify most valuable spare $i^+ \leftarrow \arg\max_i v_i$
    increment most valuable spare $n_i \leftarrow n_{i,next}$
    save objective value $\ln POS_{S,prev} \leftarrow \ln POS_S$
    save mass $m_{prev} \leftarrow m$
    calculate new objective value $\ln POS_S \leftarrow \sum_{i \in n} \ln POS_i(n_i)$
    calculate new mass $m \leftarrow \sum_{i \in n} m_i n_i$
calculate infimum mass using eqn. 4.108
return $n$, $m$, $m_{inf}$, $i^+$
```

that results in a step to a point in least concave majorant for the objective function is determined. For ORU spares, this is simply $n_i + 1$. For ODM units, this is the next entry in the set of concave indices. If $n_{i,next} > n_i^{UB}$ for any spare, then that item cannot move to the next value, since it would violate the upper bound. The marginal value for those items is set to $-\infty$. This ensures that the spare is not selected as the next item to be added, and therefore does not exceed its upper bound. For any spare that would not violate its upper bound, the marginal value v_i is calculated using equation 4.106:

$$v_i(n_i,\ n_{i,next}) = \frac{\ln POS_i(n_{i,next}) - \ln POS_i(n_i)}{\hat{m}_i(n_{i,next} - n_i)} \tag{4.106}$$

Next, the most valuable spare i^+ is identified by finding the spare with the largest marginal value – that is, the spare that provides the largest increase in $\ln POS_S$ per unit increase in mass.

$$i^+ = \arg\max_i v_i(n_i) \tag{4.107}$$

The previous objective value and mass are saved to facilitate calculation of the infimum mass later, and the objective value and mass of the new allocation is calculated.

Once the allocation satisfies the POS constraint and the while loop is completed, the infimum mass m_{inf} is calculated as

$$m_{inf} = m_{prev} + \left(\frac{m - m_{prev}}{\ln POS_S - \ln POS_{S,prev}}\right)(\ln POS_R - \ln POS_{S,prev}) \tag{4.108}$$

where m and POS_S are the mass and POS of the solution allocation, and m_{prev} and $POS_{S,prev}$ are the mass and POS of the previous allocation. The infimum mass marks the point where the piecewise linear frontier defined by the Pareto-optimal solutions found via marginal analysis crosses the required POS level, represented graphically in figure 4.3.

As noted earlier, marginal analysis is guaranteed to identify Pareto-optimal solutions, but it does not necessarily identify all of them. The process of finding the "most valuable spare" can be thought of as sweeping a line clockwise from vertical and selecting the first spares allocation that it intersects, which is the one that increases POS the most for the least increase in mass. As a result, the solutions found via marginal analysis (indicated in figure 4.3 by blue stars) effectively form the corners of a concave surface surrounding the Pareto frontier. Additional solutions, such as the notional one indicated in figure 4.3 by a red X, could exist that are above the required POS value but have a lower mass than the first Pareto-optimal solution found via marginal analysis. However, none of these solutions will have a mass lower than the infimum mass since they would have been the next allocation selected via marginal analysis if they did.

Thus the infimum mass is the largest lower bound on the set of spares allocations that satisfy the POS constraint; that is, it is the largest mass known for which all solutions in the feasible space have a value at least as large. Note that this is distinct from the concept of a minimum, since the infimum does not necessarily correspond to a value associated with a feasible solution [77]. The infimum values are used in a dynamic programming approach (branch-and-bound) to iteratively tighten bounds on the number of spares provided for each item and identify the optimal allocation [131]. Specifically, branching is performed on the value of the last item added to the allocation i^{+}. In a linear relaxation of the spares allocation problem, the optimal solution would be to add some fraction of this item – enough to satisfy the POS requirement, but no more.

The algorithm for identifying the optimal spares allocation is described in algorithm 4.3. First, a lower bound on the spares allocation is initialized by determining the minimum number of spares required for each item to achieve the POS constraint individually, since the overall allocation is guaranteed to not meet this constraint if any individual item fails to do so on its own. The upper bound for each spare is initialized to ∞. A subproblem queue is created, where a subproblem consists of the upper and lower bounds for the spares allocation $(n^{LB},\ n^{UB})$. The incumbent solution n^{*} and incumbent mass m^{*} are both initialized to ∞ (that is, an allocation of infinite spares). After this setup, the following process is repeated as long as there are subproblems in the queue:

1. The last-added subproblem $(n^{LB},\ n^{UB})$ is extracted and removed from the queue.

2. Bounded marginal analysis is used to determine the first Pareto-optimal spares allocation above the POS threshold n, called the candidate solution, along with its mass m, the infimum mass m_{inf} for that subproblem, and the last-added

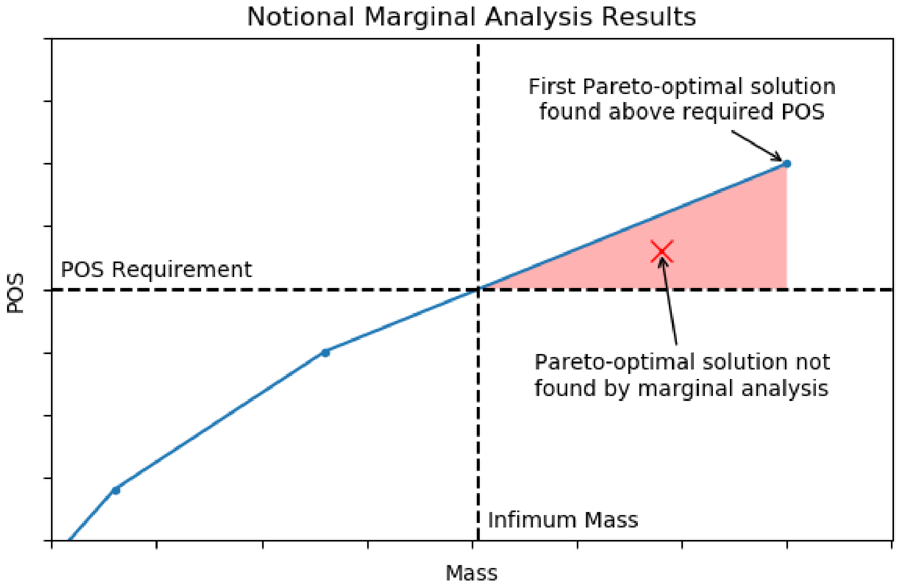

Figure 4.3: Notional POS-mass curve resulting from marginal analysis. Points along the blue line are Pareto-optimal spares allocations, and the line segments connecting them form the piecewise-linear Pareto frontier. The last point (top right) is the first Pareto-optimal above the POS requirement (indicated by the horizontal dotted line) found via marginal analysis. The infimum mass, indicated by the vertical dotted line, is the value at which the Pareto frontier crosses the POS threshold. There may be additional Pareto-optimal solutions, such as the one indicated by a red X, that fulfill the POS requirement for a lower mass than the first solution found via marginal analysis. All of these potential solutions would exist in the red shaded triangle, and have a mass higher than the infimum mass.

Algorithm 4.3 Allocate-Spares identifies and returns the optimal spares allocation n^* that achieves a desired POS while minimizing total spares mass, given an ORU list ω, analysis parameters $\mathcal{P}$ (including maximum Fourier Transform inversion error ϵ_{max}, ODM feedstock discretization Δm, and crew time discretization Δt), and a target POS value POS_R. Note that POS_R is assumed to have already taken $POS_T(t_{max})$ into account – that is, it is the required POS for spares specifically.

```
Allocate-Spares($\omega$, $\mathcal{P}$, $POS_R$)
for $i \in \omega$
    initialize $n_i^{LB} \leftarrow \min\{n_i \mid POS_i(n_i) \geq POS_R\}$
    initialize $n_i^{UB} \leftarrow \infty$
    initialize set of spares masses $\hat{m}$
    initialize set of spares POS functions $POS$ (using parameters
    in $\mathcal{P}$ as needed)
initialize queue $Q \leftarrow \{(n^{LB},\ n^{UB})\}$
initialize incumbent solution $n^* \leftarrow \infty$
initialize incumbent solution mass $m^* \leftarrow \infty$
while $Q \neq \emptyset$
    remove last-added subproblem $(n^{LB},\ n^{UB})$ from $Q$
    solve bounded marginal analysis for this subproblem
    $n,\ m,\ m_{inf},\ i^+ \leftarrow \text{BMA}(\hat{m},\ POS,\ POS_R,\ n^{LB},\ n^{UB})$
    if $m < m^*$
        update incumbent solution $n^* \leftarrow n$
        update incumbent mass $m^* \leftarrow m$
    if $m_{inf} < m^*$
        create first new subproblem $(n^{LB1},\ n^{UB1}) \leftarrow (n^{LB},\ n^{UB})$
        adjust lower bound for last-added item $n_{i^+}^{LB1} \leftarrow n_{i^+}$
        append $(n^{LB1},\ n^{UB1})$ to $Q$
        create second new subproblem $(n^{LB2},\ n^{UB2}) \leftarrow (n^{LB},\ n^{UB})$
        adjust upper bound for last-added item $n_{i^+}^{UB2} \leftarrow n_{i^+} - 1$
        adjust lower bounds for other items $n_i^{LB2} \leftarrow n_i\ \forall\ i \neq i^+$
        append $(n^{LB2},\ n^{UB2})$ to $Q$
return $n^*$
```

spare i^+.

3. If the candidate solution mass m is less than the incumbent mass m^*, then the incumbent solution and mass are replaced by the candidate solution and mass.

4. If the infimum mass m_{inf} is less than the mass of the incumbent solution m^*, then there may be better solutions that are subproblems of this subproblem. The subproblem is branched on the value allocated to the last-added spare, creating two new subproblems:

   (a) For the first subproblem, the lower bound for i^+ is set to its current value (i.e. $n_{i^+}^{LB} \leftarrow n_{i^+}$), and all other lower bounds and all upper bounds remain unchanged.

   (b) For the second subproblem, upper bound for i^+ is set to one less than its current value (i.e. $n_{i^+}^{UB} \leftarrow n_{i^+} - 1$), and all other upper bounds remain unchanged. Since in this subproblem the only change is in the upper bound, the marginal analysis results will be identical to the previous subproblem until that upper bound is reached, or until the algorithm attempts to allocate i^+ to arrive at the same solution as before, but cannot because of the upper bound. At this point, all other spares will have the same allocations that they had in the previous subproblem. As a result, the optimization process can be accelerated by setting the lower bound of all items other than i^+ to their values in the candidate solution n.

5. The process is repeated until the queue is empty, at which point there are no remaining subproblems to explore that may contain better solutions. The incumbent solution at this point is the optimal solution.

4.3 Optimization Strategy

MCROSS is formulated to interface with a range of optimal COP solvers by structuring the problem constraints in MiniZinc [117] and using the PyMzn Python library

[118] to call the resulting external programs and receive outputs. The specifics of the algorithms used by particular solvers to find the Pareto frontier for the supportability strategy optimization problem are beyond the scope of this dissertation, which is focused on formulating constraints and defining the model used to evaluate the mass and crew time associated with solutions and partial assignments. However, the general optimization strategy used for this research is described in this section. Section 4.3.1 describes the multiobjective branch and bound algorithm used to prune the search space and efficiently identify Pareto-optimal solutions. Section 4.3.2 presents the models used to calculate lower bounds on the amount of mass and crew time required for a partial assignment – that is, an assignment to the supportability strategy decision variables in which some variables remain unassigned, and are represented by a domain of possible assignments rather than a single value. These lower bounds are used for pruning during search. Finally, Section 4.3.3 describes the heuristics used for variable and domain ordering during search.

4.3.1 Multiobjective Branch and Bound

Branch and bound [131] is a search strategy in which bounds on objective values are used to identify regions of the decision space that cannot contain solutions more optimal than the ones already found, and therefore do not need to be searched. This approach is used in the single-objective sense as part of the mass model (Section 4.2.3 in order to find the minimum-mass spares allocation that meets the POS requirement. In that application, the infimum mass for a particular set of upper and lower bounds on the number of spares provided for each ORU is used to prune a branch of the search tree if it is larger than the mass of a known allocation.

Multiobjective branch and bound expands this approach to multiple dimensions using the concept of Pareto dominance. Specifically, a particular point x in multiobjective space is considered dominated (in terms of minimization) by another point y if $y_i \leq x_i$ for all dimensions and $y_i < x_i$ in at least one dimension. A decision vector whose set of objective values is dominated by those of another decision vector is considered dominated, and the set of decisions that are not dominated by any other decisions are Pareto-optimal [137]. Where single-objective branch and bound prunes portions of the search tree if the bound on the best objective value in that portion of the tree is greater than a known solution, multiobjective branch and bound prunes when the point represented by the lower bound on all objective values is dominated by a known solution [138].

To implement multiobjective branch and bound on the supportability strategy optimization problems, the search assigns each decision variable in turn, following the variable and domain ordering heuristics described in Section 4.3.3. Once an assignment is made, constraints are propagated to all other decision variables using AC-3 [139] (or any other arc consistency algorithm; as noted earlier, the specifics of implementation are left to the third-party COP solvers and are not within the scope of this dissertation) and the resulting decision vector is evaluated. If it is a complete assignment – that is, one in which all decision variables are assigned to specific values – it is evaluated using the model described in Sections 4.1 and 4.2. If it is a partial assignment, meaning that one or more decision variables still have several potential values that they could take, the lower bounds on mass and crew time that could be achieved given that partial assignment are evaluated using the method described in Section 4.3.2.

The resulting objective values are compared to a candidate Pareto frontier made up of known non-dominated solutions. If a full assignment is dominated by any point on this candidate frontier, then it is not part of the final Pareto frontier and is discarded. If, however, that assignment is not dominated, it is added to the candidate Pareto frontier and any assignments that it dominates are removed. When a partial assignment is examined, it is similarly discarded if the lower bounds on mass and crew time are dominated by any full assignment on that frontier, since any completion of that partial assignment would also be dominated. If a partial assignment is not dominated, however, then it is added to a queue of partial assignments to be expanded in the search. When no feasible completion of a partial assignment exists, the search backtracks to try a different assignment. This process is repeated until there are no more unexpanded partial assignments, at which point the candidate Pareto frontier is the solution to the multiobjective COP [138].

In its simplest formulation, the candidate Pareto frontier used to prune partial assignments is gradually built up during search as full assignments are identified and characterized. However, the process can be accelerated by providing an initial candidate Pareto frontier. For this research, the candidate Pareto frontier is initialized by performing a full enumeration on a much more tightly constrained version of the supportability strategy optimization problem which adds the following constraints:

$$\bigvee_{\substack{j,k\in\phi\\ j\neq k\\ p_j=p_k=i}} (u_j \wedge u_i) \implies \nu_i \qquad \forall\, i \in \phi \tag{4.109}$$

$$\neg\xi_s \tag{4.110}$$

$$\eta_i = 1 \qquad \forall\, i \in \phi \tag{4.111}$$

$$\hat{C}_{ij} \wedge u_i \wedge u_j \implies C_{ij} \qquad \forall\, i,j \in \phi \tag{4.112}$$

These constraints ensure that (respectively):

- If any two children of a given item i are replaceable items, then i is considered covered by a lower level of maintenance.
- ODM is not used.
- Distributed functionality is not used.
- Commonality is used whenever it is available.

These additional constraints effectively remove commonality, ODM, and distributed functionality decisions and greatly simplify level of maintenance decisions, resulting in a tradespace that is significantly smaller than the original problem. For the cases examined in this dissertation, these additional constraints are sufficient to reduce the size of the problem to the point where it can be enumerated and evaluated very quickly, and the resulting Pareto frontier extracted via filtering. However, for larger problems multiobjective branch and bound or other, more sophisticated optimization methods could be applied. Each feasible solution to the more constrained problem will also be a feasible solution to the full problem, and therefore this approach is a valid method to populating an initial Pareto frontier.

4.3.2 Calculating Bounds for Partial Assignments

This section describes the process used to calculate the lower bound on mass and expected crew time, given a partial assignment to the supportability strategy decision variables. These bounds are constructed by decomposing each objective into its constituent elements and finding a lower bound associated with each one. Specifically, based on the definition of total expected crew time given in equation 3.13, the lower bound on total expected crew time is the sum of the lower bound on preventative maintenance crew time and the lower bound on expected corrective maintenance crew time.

$$\mathrm{E}[T]^{LB} = t_{PM}^{LB} + \mathrm{E}[T_{CM}]^{LB} \tag{4.113}$$

Similarly, based on the definition of total mass in equation 3.11, the lower bound on total mass is the sum of the lower bounds associated with system mass, preventative maintenance mass, and corrective maintenance mass.

$$m_{tot}^{LB} = m_S^{LB} + m_{PM}^{LB} + m_{CM}^{LB} \tag{4.114}$$

In each of these equations, a superscript ^{LB} is used to indicate a lower bound. The approach used to calculate these bounds is described in each of the sections below.

Each of the following sections makes use of manipulations of partial assignments. In order to simplify these descriptions, the following notation and terminology is used:

- A partial assignment to a decision variable is denoted x^{p}. For example, a partial assignment to the level of maintenance decision variables u might be $u^{\mathrm{p}} = [0,\ 1,\ [0,\ 1]\,,\ 1]$, in which items 1, 2, and 4 have assigned values, but item 3 is unassigned and therefore has an entry equal to its available domain, namely $[0,\ 1]$.

- A maximized assignment, denoted $x^{\uparrow}$, is one in which all unassigned variables are set to the largest remaining value in their domain, based on a given partial assignment. For example, the maximized version of u^{p} above is $u^{\uparrow} = [0,\ 1,\ 1,\ 1]$.
- A minimized assignment, denoted $x^{\downarrow}$, is one in which all unassigned variables are set to the smallest remaining value in their domain, based on a given partial assignment. For example, the minimized version of u^{p} above is $u^{\downarrow} = [0,\ 1,\ 0,\ 1]$.

This terminology is applied normally to all decision variables, with only one special case. Since the diagonal of the commonality matrix C must correspond to the selected level of maintenance, the maximized and minimized assignments $C^{\uparrow}$ and $C^{\downarrow}$ are assumed to have an empty diagonal that is filled in via summation with a diagonalized vector as needed. Note that for assignment maximization and minimization purposes, the convention that False = 0 and True = 1 is applied to binary variables.

The calculations and decision assignments used to find lower bounds to not necessarily result in a feasible solution, and there is unlikely to be a single assignment to the decision variables that simultaneously minimizes both objectives in the remaining search space. However, the calculations described below do provide an infimum on both mass and crew time. The resulting values are known to be at least as good as the objective values associated with any completion of a given partial assignment, and therefore are valid values to use for pruning.

Lower Bound on Crew Time

Four of the five supportability strategy decisions – ODM, commonality, redundancy, and distributed functionality – have relatively straightforward impacts on crew time. Specifically, preventative and corrective maintenance crew time is minimized when redundancy is maximized and ODM, commonality, and distributed functionality are minimized. Redundancy's primary impact is to increase the number of instances of a particular component installed in the system and therefore reduce the number of maintenance actions – and crew time – required. ODM's primary impact on crew time is to apply the ODM rate overhead factor $\gamma_{\Lambda odm}$ to crew action rates. Since this research assumes that components manufactured in space will, in the near future at least, be of lower quality than similar components manufactured on the ground, $\gamma_{\Lambda odm}$ is assumed to be greater than 0 and application of ODM can only increase an item's crew action rate and the resulting amount of crew time required. Similarly, commonality aggregates ORU characteristics by taking the worst characteristics of all repairable items in the common set, and can therefore only result in items with higher crew action rates, lower life limits, and longer maintenance crew times. Finally, since distributed functionality increases the number of instances of a particular item in a system that are operational at a given time and does not scale maintenance crew time per item, each module that a system is distributed across is effectively a multiplier on the amount of crew time required to maintain the system. Therefore, the relevant completions of r^{p}, ξ^{p}, c^{p}, and η^{p} for a given partial assignment are $r^{\uparrow}$, $\xi^{\downarrow}$, $C^{\downarrow}$, and $\eta^{\downarrow}$. This final parameter, the distributed functionality assignment, is used along with equation 4.2 to calculate the total number of modules for each item in this assignment, denoted $\hat{\eta}^{\downarrow}$.

The variables described above, along with the system description, determine the number of maintenance actions that will occur. Level of maintenance decisions u determine the amount of crew time associated with each decision. Since the expected number of corrective maintenance actions for a particular replaceable item that combines several components is equal to the sum of the expected number of actions for each component individually, corrective maintenance crew time is minimized when each component is assigned to its lowest crew time option.

The set of options available to each component consists of the any direct ancestors (i.e. ancestors that do not have a replaceable item between them and the component in the indenture structure) that have not already been specified as non-replaceable items (i.e. $u_i = 0$), as well as the component itself if it has not been specified as non-replaceable. These options can be identified using a matrix W', called the direct descendants matrix. The entries W'_{ij} are equal to 1 if j is a direct descendant of i, meaning there are no repairable items between i and j in the indenture structure and i is an item that j could be assigned to for level of maintenance decisions. If these

Algorithm 4.4 Get-Direct-Descendants-Matrix returns the direct descendants matrix W', which has entries equal to 1 where item j is a direct descendant of item i, meaning there are no repairable items between i and j in the indenture structure. This is calculated as a function of a partial level of maintenance assignment u^{p} (which defines $u^{\uparrow}$ and $u^{\downarrow}$), adjacency matrix A, and descendants matrix D

```
Get-Direct-Descendants-Matrix($u^{\mathrm{p}}$, $A$, $D$)
1  $U \leftarrow \mathrm{diag}(u^{\downarrow})$
2  $W' \leftarrow (I + D) \circ (1 - UDU)$
3  $k \leftarrow 1$
4  while $UDUA^k$ is not empty
5        $W' \leftarrow W' \circ (1 - UDUA^k)$
6        $k \leftarrow k + 1$
7  $W' \leftarrow \mathrm{diag}(u^{\uparrow})W'$
8  return $W'$
```

conditions are not met by a particular pair $i,\ j$, then entry W'_{ij} is equal to 0.

The direct descendants matrix is generated using algorithm 4.4, which follows a similar approach to that used to generate the content matrix W described in Section 4.1. The key difference between the two algorithms appears in line 2. Where the content matrix is calculated by iteratively removing descendants of replaceable items that are themselves descendants of replaceable items from $U(I + D)$ (a matrix with entries equal to 1 where i is a replaceable item and j is i or a descendant of i), the direct descendants matrix is calculated by removing entries from $I + D$. As a result, it includes all direct ancestors for each item j, regardless of whether they are also replaceable items. In addition, the final calculation in line 7 zeros out rows corresponding to items that have been designated non-repairable items (i.e. any row i for which $u_i^{\uparrow} = 0$), since these cannot be direct ancestors of any item.

Next, adjusted parameter vectors are calculated for characteristics that may be changed based on commonality assignments. The adjusted crew time vector t', adjusted mean crew action rate $\bar{\lambda}'_C$, and adjusted life limit ℓ' are:

$$t' = \max_R \big((C^{\downarrow} + \mathrm{diag}(u^{\uparrow}))\, \mathrm{diag}(t) \big) \tag{4.115}$$

$$\bar{\lambda}'_C = \max_R \big((C^{\downarrow} + \mathrm{diag}(u^{\uparrow}))\, \mathrm{diag}(\bar{\lambda}_C \circ d) \big) \tag{4.116}$$

$$\ell' = \left(\max_{R}\left((C^{\downarrow} + \operatorname{diag}(u^{\uparrow}))\operatorname{diag}((\ell \circ d)^{\circ -1})\right)\right)^{\circ -1} \tag{4.117}$$

These calculations implement the same approach used to apply commonality in equations 4.31 through 4.39, except that they use the entire commonality matrix instead of simply the unique nonzero rows, and therefore result in adjusted values for each item in the system. Duty cycle is included in the calculations for adjusted mean crew action rate and life limit in order to ensure that the resulting characteristics are based on the effective parameters (i.e. parameters that take into account duty cycle).

In addition, a vector t_{min} is defined which indicates the lowest maintenance crew time available to each item.

$$t_{min} = \left(\max_{C}\left(\operatorname{diag}(t'^{\circ -1} \circ u^{\uparrow})W'\right)\right)^{\circ -1} \tag{4.118}$$

The entries of $t'^{\circ -1} \circ u^{\uparrow}$ are equal to $t_i'^{-1}$ if i is a repairable item, and 0 if it is not, in order to prevent non-repairable items from being selected, using an approach similar to the one used for life limits in equation 4.16. Maximization along the columns of an inverted matrix is used to identify the minimum crew time option in each case. This minimum crew time vector is used along with the adjusted mean crew action rate and other parameters to calculate the lower bound on expected crew time as follows (keeping in mind that t_{min} is a row vector):

$$\mathrm{E}[T_{CM}]^{LB} = t_{min}\left(\tau \circ q \circ \hat{\eta}^{\downarrow} \circ \theta \circ \bar{\lambda}_C' \circ \kappa \circ (1 + \gamma_{\Lambda r} \circ r^{\uparrow}) \circ (1 + \gamma_{\Lambda odm}\xi^{\downarrow})\right) \tag{4.119}$$

This equation multiplies the minimum time vector by the expected number of crew actions for each component, using the adjusted failure rate to account for commonality and applying the adjustments associated with redundancy and ODM. The sum of the number of expected crew actions times the minimum available crew time for each component provides the minimum possible expected corrective maintenance crew time.

Note that, in order for component j to be included in a replaceable item i, it must be a direct descendant of i, there must be no replaceable items between i and j.

This assignment approach, which places each component in the item with the lowest maintenance crew time among the available direct ancestors, does not generate crew time estimates based on level of maintenance assignments that violate this constraint. Once assignment is complete, each component has a clear path to the item that it is assigned to. This is due to the fact that the indenture structure is arborescent. If item i is an ancestor of both item j and k, then all ancestors of i are also ancestors of j and k. If i is selected by one of its descendants, it is the minimum crew time option among any of its available ancestors. Any other descendants of i would either also select i or some other item at a lower level in the indenture structure. As a result, assigning components to their lowest-crew-time ancestors ensures that no component is cut off from a lower crew time option by the assignment of another component. However, since the specific level of maintenance assignment is not necessary for calculating the minimum crew time option, it is handled implicitly in the selection of the lowest crew time option for each component in equation 4.118.

The lower bound on preventative maintenance crew time is calculated using a similar approach. However, identification of the assignment that associates each life-limited component with the repairable item that results in minimum overall crew time is more complex due to interactions between components. If a particular repairable item contains more than one life-limited components, then the number of scheduled replacements for that item is driven by the component with the lowest life limit. Any scheduled replacements associated with components that have longer life limits are effectively handled by this shorter replacement interval. As a result, the most effective way to reduce preventative maintenance crew time is not to simply assign each component to the lowest-crew-time option available to it, but rather to combine life limited items into the same repairable item where possible.

The vector $n_{PM\theta}$, defined below, gives the number of times that a given instance of a component reaches its life limit.

$$n_{PM\theta} = \left\lfloor \tau \circ \left(\ell' \left(1 + r^{\uparrow} \right) \right)^{\circ -1} \right\rfloor \tag{4.120}$$

Algorithm 4.5 Get-PM-Crew-Time-LB returns the lower bound on the preventative maintenance crew time associated with a partial assignment to the decision variables, based upon the number of preventative maintenance actions per component $n_{PM\theta}$, the direct descendants matrix W', available item crew time vector t', quantity vector q, and number of modules vector $\hat{\eta}^{\downarrow}$.

```
Get-PM-Crew-Time-LB($n_{PM\theta}$, $W'$, $t'$, $q$, $\hat{\eta}^{\downarrow}$)
Initialize preventative maintenance crew time lower bound $t_{PM}^{LB} \leftarrow 0$
Create a queue of indices $i$ corresponding to nonzero $n_{PM\theta i}$
while the queue is not empty
    Remove and return queue entry $i$ corresponding to the highest $n_{PM\theta i}$
    $k \leftarrow \arg\max_k \left( W'_{ki} (t'_k q_k \hat{\eta}_k^{\downarrow})^{-1} \right)$
    $t_{PM}^{LB} \leftarrow t_{PM}^{LB} + t'_k q_k \hat{\eta}_k^{\downarrow} n_{PM\theta i}$
    $t'_k \leftarrow 0$
return $t_{PM}^{LB}$
```

Note that this is the number of preventative maintenance events for a single instance of that component, not the total number of preventative maintenance events. To get the total number, $n_{PM\theta}$ would be multiplied by the relevant quantity q and number of modules $\hat{\eta}$. However, those parameters depend on the characteristics of the replaceable item containing the component, not the component itself. Given this vector, the minimum preventative maintenance crew time allocation is identified by iteratively assigning components to available direct ancestors that will result in the lowest maintenance crew time, taking into account the quantity of candidate repairable items (i.e. $q\hat{\eta}^{\downarrow}$), following the process defined in Algorithm 4.5.

Instead of assigning all items at once, life-limited components are assigned in descending order in terms of the number of preventative maintenance actions, with the component with the largest $n_{PM\theta i}$ going first. Once a component is assigned to an item, further assignments of components with lower $n_{PM\theta i}$ will not affect the preventative maintenance crew time associated with that item, since preventative maintenance actions will be driven by the earlier component. If any component with lower $n_{PM\theta i}$ can select that item, it should, and in doing so will have no effect on the total preventative maintenance crew time. To enforce this, for subsequent assignments the crew time associated with any item that already has a life-limited component assigned to it is set to 0, thereby ensuring that subsequent life-limited components choose that item if possible. The process is repeated, cumulatively adding the amount of crew time associated with each life limited item until all life-limited items are assigned. In line 5, where the option with the lowest crew time is selected for the component in question, $\arg\max_k \left(W_{ki}(t'_k q_k \hat{\eta}_k)^{-1}\right)$ is used instead of $\arg\min_k \left(W_{ki} t'_k q_k \hat{\eta}_k\right)$ in order to avoid selecting an item simply because $W_{ki} = 0$, which would indicate that k is not a direct ancestor of i. The lower bound on total maintenance crew time is then the sum of the lower bounds associated with corrective maintenance and preventative maintenance.

Lower Bound on Mass

As with crew time, the lower bound on total mass is found by deriving lower bounds on its individual elements, specifically system mass, preventative maintenance mass, and corrective maintenance mass. System mass is primarily driven by redundancy r, distributed functionality η, and ODM decisions ξ and ξ_s. Commonality has a smaller impact, since all items in a common set take the highest mass within that set. Level of maintenance decisions u do not impact system mass, since the mass of the system is a function of the total mass and quantity of the components within it, and level of maintenance decisions do not impact those values. Therefore, u is ignored for the purposes of placing a lower bound on system mass.

Redundancy, ODM, and distributed functionality can only increase system mass, since they either increase the number of instances of a particular component in the system or increase the mass of an individual component via the overhead factor γ_{odm}. ODM also has a strong impact on system mass via the manufacturing system mass m_{odm}, which is only required if $\xi_s = 1$. Therefore, system mass is minimized when ODM is used as little as possible. For distributed functionality, the best-case outcome occurs when the mass scaling factor for a particular item γ_i is 1, meaning that the mass of that item scales fully as it is distributed. However, since the number of instances of that item also increases, the net effect is that total system mass is unchanged. Any component with $\gamma_i < 1$ will experience increased mass from distributed functionality. Therefore the relevant completions for minimizing system mass are $r^{\downarrow}$, $\xi^{\downarrow}$, $\xi_s^{\downarrow}$, and $\eta^{\downarrow}$ (which is again used to calculate $\hat{\eta}^{\downarrow}$).

The potential mass increases associated with commonality are applied by generating a commonality matrix C_S that only applies commonality decisions to components.

$$C_S = \operatorname{diag}(\theta)\left(C^{\downarrow} + I\right) \tag{4.121}$$

While other, higher-level items may be specified as common in the partial assignment C^{p}, it is not necessarily true that all components descended from those items will be included in that higher-level item, and therefore application of commonality rules to all of those items may overestimate system mass. However, any components that have commonality applied to them are known to be their own repairable items, and therefore commonality rules can be applied directly in order to adjust their characteristics.

System mass is then calculated using the adjusted mass and quantity vectors defined below.

$$m' = \max_R \left(C_S \operatorname{diag}\left(m \circ \left(1 - \left(1 - \hat{\eta}^{\downarrow \circ -1}\right) \circ \gamma\right) \circ \left(1 + \gamma_{odm} \circ \xi^{\downarrow} \circ \theta\right)\right)\right) \tag{4.122}$$

$$q' = \left(q + (1 + \gamma_r) \circ r^{\downarrow}\right) \circ \hat{\eta}^{\downarrow} \tag{4.123}$$

These vectors capture adjustments related to redundancy, commonality, ODM, and distributed functionality. Note that, as with commonality, the mass overhead associated with ODM is only applied to components that use ODM. System mass is then

$$m_S^{LB} = q'^{T} m' + m_{odm} \xi_s^{\downarrow} \tag{4.124}$$

Preventative maintenance mass is minimized when the lowest available level of maintenance is used, since this separates items as much as possible and minimizes the amount of mass replaced each time a component reaches its life limit. This lowest level of maintenance, denoted u_{LL}, is calculated as follows. First, a row vector x is generated indicating the greatest depth in the indenture structure available to each item

$$x = \arg\max_{C} \left(\operatorname{diag}(\hat{d} \circ u^{\uparrow}))W' \right) \tag{4.125}$$

where $\arg\max_{C}$ returns the index of the maximum value within each column and $\hat{d}$ is the depth vector calculated in equation 3.6. The lowest level of maintenance is then determined by setting the greatest depth option for each component to be a repairable item. That is,

$$u_{LLi} = \begin{cases} 1 & \text{if } (i \in x) \wedge (\theta_i = 1) \\ 0 & \text{else} \end{cases} \tag{4.126}$$

Note that, while u_{LL} has entries equal to 0 where the partial assignment designates an item as non-replaceable, and the level of maintenance assignment described by u_{LL} does not allow components to be contained within items at a higher level than any items which have already been assigned as replaceable, in some cases u_{LL} may select a lower level of maintenance than an item with $u_i^{\mathrm{p}} = 1$ in the partial assignment. That is, there may be items for which $u_i^{\mathrm{p}} = 1$ but $u_{LLi} = 0$. In all of these cases, however, this will be because a lower level of maintenance has been selected. Since the lower level of maintenance results in lower mass, this is a valid relaxation of the level of maintenance problem for the purposes of finding a lower bound on mass.

Commonality and ODM provide no benefits in terms of preventative maintenance mass. Specifically, since ORUs covering a set of common items take the worst characteristics of those items, applying commonality can only increase mass and/or decrease life limits, or, at best, leave them the same. Preventative maintenance demands are deterministic, and therefore commonality does nothing to reduce the number of maintenance items required. Similarly, ODM can only increase the mass of a particular item via the overhead γ_{odm}, and the benefits of commonality of material have no impact on deterministic maintenance demands. Distributed functionality also provides no benefit in terms of preventative maintenance mass. If a life-limited item is split into multiple, smaller items, all of which operate concurrently, all of those items are still life-limited. The only change is that the number of replacements required is multiplied by the number of modules, and the mass associated with each replacement is scaled down according to equation 3.2. At best, distribution of functionality can only result in a system with the same preventative maintenance mass requirements as before.

Redundancy has two conflicting impacts on preventative maintenance mass. Increased redundancy extends the lifetime of a given component, resulting in fewer replacements. However, it also increases the mass of that component, and as a result each replacement incurs a larger mass cost. If a life-limited component is the only component in a given ORU, increasing redundancy results in fewer preventative maintenance actions and more mass per action. If the total number of instances (i.e. $1+r$) is a whole factor of the total number of end-of-life events for that component, then the level of redundancy is efficient, which means the number of redundant instances provided is the minimum amount required to reduce the number of replacements required. If $1+r$ is not a factor of the number of end-of-life events, however, then some of the redundant instances are effectively "wasted," since the resulting number of required replacements could have been achieved at a lower level of redundancy. Assuming efficient allocations of redundancy and ignoring the mass overhead γ_r, in the single component case the preventative maintenance mass saved by reducing the number of replacements required will exactly match the preventative maintenance mass increase resulting from increasing the mass for each replacement. Thus, any value of γ_r greater than 0 will result redundancy being a more costly solution (from a mass perspective) than using spares when components are their own repairable items, and the minimum-mass option will be to not use redundancy on those items unless it has already been explicitly assigned in r^{p}.

When a component is combined with other, non-life-limited components or components with longer life limits, efficient redundant allocations result in lower total preventative maintenance mass if γ_r is ignored, since the mass of all of those other components is not increased but the number of replacements required is decreased. The redundancy allocation that results in the lowest preventative maintenance mass is the one in which each item has the highest available efficient redundancy allocation. Therefore, to identify the lower bound on preventative maintenance mass the efficient redundancy r'_i for each unassigned r_i calculated as

$$r'_i = \max_r \left\{ r \in r_i^{\mathrm{p}} \;\middle|\; \left((r+1) \bmod \left\lfloor \frac{\tau d_i}{\ell_i} \right\rfloor = 0 \right) \vee (r = \min r_i^{\mathrm{p}}) \right\} \tag{4.127}$$

where r_i^{p} is the available domain for r_i based on the given partial assignment, d_i and ℓ_i are the duty cycle and life limit for that component, mod is the modulo operator (i.e. the remainder after division), and τ is the mission endurance.

The preventative maintenance mass lower bound is associated with the case where efficient redundancy is used and the redundancy mass overhead is ignored for items with unassigned redundancy. Therefore, instead of setting unassigned r_i to r'_i, a modified life limit vector ℓ' (distinct from the one used for preventative maintenance crew time) is defined as

$$\ell'_i = \ell \circ (1 + r') \tag{4.128}$$

Using this modified life limit in the calculation below avoids incurring the mass overhead associated with redundancy for items with unassigned r_i.

Therefore, the lower bound on preventative maintenance mass is calculated by creating and evaluating an ORU list using the same technique described in Sections 4.1 and 4.2, using the adjusted life limit ℓ' calculated in equation 4.128 and the following supportability strategy:

- Level of maintenance is set to u_{LL},
- ODM decisions are set to $\xi^{\downarrow} \circ \theta$ in order to apply the mass overhead of ODM to only the components that use it,
- Commonality decisions are set to $\mathrm{diag}(\theta)C^{\downarrow} + \mathrm{diag}(u_{LL})$ in order to apply commonality adjustments only to components that use it,

- Redundancy decisions are set to $r^{\downarrow}$, and
- Distributed functionality decisions are set to $\eta^{\downarrow}$.

Corrective maintenance mass is also minimized when the lowest available level of maintenance is selected, for similar reasons as those described for preventative maintenance mass above. Corrective maintenance mass is determined by the spares allocation that provides the required level of POS for minimum mass, as determined by the spares allocation optimization technique described in Section 4.2.3. Each spare provided for a given ORU is equivalent to adding a spare for each of the components within that ORU. Therefore, the aggregation of two or more components into a single ORU is effectively the same as adding a constraint to spares allocation optimization stating that the number of spares provided for those components must be equivalent. This constraint can at best result in a spares allocation with the same mass as the optimal one found when the constraint is not present. Put another way, a lower level of maintenance provides a relaxation of the spares allocation problem compared to a higher level of maintenance. Therefore, evaluation of corrective maintenance mass lower bounds uses the same lowest-level allocation u_{LL} as was applied to find the preventative maintenance mass lower bound.

Commonality and ODM both have the potential to significantly reduce corrective maintenance mass, but they are subject to a set of constraints that couple those decisions as well as decisions related to level of repair. In order to relax these constraints and identify the commonality and ODM assignments that minimize corrective maintenance crew time, the matrix C_{CM} and vector ξ_{CM} are defined by applying the following rules:

1. Any items that could use commonality or ODM in the relevant partial assignments do so, except where barred from the constraints listed below.

2. Commonality and ODM are both considered to be inherited from above for direct descendants. That is, if i is common with j, any direct descendants of i or j are considered common with each other. Similarly, if i uses ODM, then any direct descendants of i are also assumed to use ODM.

Algorithm 4.6 Get-CM-C-and-ODM returns the commonality matrix C_{CM} and ODM vector ξ_{CM} used to calculate the lower bound on corrective maintenance mass.

```
Get-CM-C-and-ODM($C^{\uparrow}$, $W'$, $\xi^{\uparrow}$, $u_{LL}$)
 1  $C_{CM} \leftarrow C^{\uparrow} + \text{diag}(\max_R C^{\uparrow})$
 2  $n_{nz} \leftarrow$ number of nonzero entries in $C_{CM}$
 3  $C_{CM} \leftarrow C_{CM} W'$
 4  $(C^T_{CM})_{ij} \leftarrow C_{CMij} \ \forall \ i, \ j \mid j \geq i$
 5  while number of nonzero entries in $C_{CM} > n_{nz}$
 6      $n_{nz} \leftarrow$ number of nonzero entries in $C_{CM}$
 7      $C_{CM} \leftarrow C_{CM} W'$
 8      $(C^T_{CM})_{ij} \leftarrow C_{CMij} \ \forall \ i, \ j \mid j \geq i$
 9  $C_{CM} \leftarrow \min\left[C_{CM}, 1\right]$
10  $\xi_{CM} \leftarrow \min\left[W' \left(I + C_{CM}\right) \xi^{\uparrow}, 1\right] \circ u_{LL}$
11  $C_{CM} \leftarrow \text{diag}((1 - \xi_{CM}) \circ u_{LL}) C_{CM} \, \text{diag}((1 - \xi_{CM}) \circ u_{LL})$
12  $C_{CM} \leftarrow C_{CM} \circ (1 - I) + \text{diag}(u_{LL})$
13  return $C_{CM}$, $\xi_{CM}$
```

3. ODM is considered to be inherited across commonality. That is, if i is common with j and j uses ODM, then i does as well, regardless of any other constraints.

4. Items do not use both commonality and ODM. Where both are available, ODM is used instead of commonality, since ODM is effectively a relaxation of commonality and will result in lower corrective maintenance mass.

5. Only repairable items in u_{LL} use commonality or ODM.

This calculation is implemented in Algorithm 4.6.

First, C_{CM} is initialized to $C^{\uparrow}$, with a diagonal equal to 1 for items that do or could use commonality in the partial assignment. This matrix is then multiplied by W' to expand this commonality to direct descendants, and the portion of the matrix above the diagonal is copied across the diagonal to create a symmetric matrix (line 4). This process is repeated as long as the number of nonzero entries in C_{CM} increases, since that indicates that new items are being added. Once that process is complete, any nonzero entries of C_{CM} are set to 1 using the element-wise minimum $\min[\cdot]$ in line 9. The ODM vector ξ_{CM} is then defined by left-multiplying $\xi^{\uparrow}$ by this commonality matrix (with an identity matrix added to fill the diagonal) to expand ODM across common pairs, then left-multiplying again by W' to propagate to direct descendants. Nonzero entries are again set to 1, using the same technique described above, and element-wise multiplication with u_{LL} is used to leave only entries corresponding to repairable items in the lowest level of maintenance. Finally, entries in C_{CM} corresponding to items that either use ODM or are not in the lowest level of maintenance are zeroed out by multiplying by $\text{diag}((1-\xi_{CM}) \circ u_{LL})$ from both the left and the right, and the diagonal is zeroed out and replaced by u_{LL}.

Component masses, failure rates, and mass scaling factors are adjusted to account for these decisions, as well as redundancy and distributed functionality. ODM mass overhead is applied to components that use ODM and also appear in the lowest level of maintenance, following the same logic used for preventative maintenance mass.

$$m' = m \circ \left(1 + \gamma_{odm} \circ \xi^{\downarrow} \circ \theta \circ u_{LL}\right) \tag{4.129}$$

Commonality is a decision applied between specific items in the indenture structure, and therefore does not fully handle cases where descendants of a given item have unassigned level of maintenance decisions u. Those items may or may not be included in the higher level item, and therefore may or may not impact the characteristics of the resulting common ORU or be replaceable items of their own. However, a lower bound can be calculated by noting that ODM is a relaxation of commonality, since it allows different items to obtain the benefits of common materials without enforcing common design. Therefore, for the purposes of obtaining a lower bound on corrective maintenance mass each set of common items is modeled as a specific type of ODM feedstock, separate from any feedstock used for ODM items. That is, the ODM feedstock model described in Section 4.2.3 is applied to each collection of items to determine the distribution of the total amount of corrective maintenance mass that will be required for those items. Since commonality is assumed to be inherited from above for lower bounding purposes, this approach ensures that the potential benefits of commonality are provided to all items that could potentially use it, without forcing lower-level items to be combined upwards in order to gain those benefits.

Redundancy can have different effects on corrective maintenance mass. On one hand, increased redundancy can reduce the number of times that an ORU requires replacement due to random failures. However, it also increases the mass of each spare, and could reduce the efficiency of the optimal spares allocation. Since there is no simple way to determine the appropriate level of redundancy to minimize the overall corrective maintenance mass, for the purposes of lower bounding all components with unassigned redundancy are provided the benefits of redundancy without the costs. That is, the failure rates for those components are adjusted as if they were provided with the maximum amount of redundancy, but there is no mass increase. This adjustment is implemented as

$$\bar{\lambda}' = \bar{\lambda} \circ \left(1 + \gamma_{\Lambda odm} \xi^{\downarrow} \circ \theta \circ u_{LL}\right) \circ \left(1 + \gamma_{\Lambda r} \circ \left(r^{\uparrow} - r^{\downarrow}\right)\right)^{\circ -1} \tag{4.130}$$

where the vector $r^{\uparrow} - r^{\downarrow}$ is equal to the maximum additional redundancy that could be applied above the lowest allocation for each entry r^{p}; for entries corresponding to an assigned redundancy, this is equal to 0.

The motivation behind distributed functionality is to split a given system into a larger number of smaller modules in order to enable more efficient spares allocations, thereby reducing corrective maintenance mass. However, the effectiveness of this strategy depends strongly on the scaling factor γ_i associated with each component. As noted in Section 3.2.1, this scaling parameter effectively describes the portion of a given component's mass that changes as a function of the number of modules that it is split across. Higher values of γ_i indicate cases where the component's mass is more sensitive to distributed functionality, while lower values indicate that distributed functionality may not produce the desired results since the split component may not be that much smaller than the original. For the purposes of bounding corrective maintenance mass, the best possible outcome is to apply distributed functionality as much as possible with component mass scaling as much as possible. Therefore, $\eta^{\uparrow}$ is used for corrective maintenance mass bounds calculations, and γ_i for any components that are descended from subsystems with unassigned η_i are set to 1 for the purposes of mass scaling calculations.

$$\gamma' = \gamma + (1 - \gamma) \circ \min\left[\mathrm{diag}(\eta^{\uparrow} - \eta^{\downarrow})(I + D),\ 1\right] \tag{4.131}$$

The only interaction between η and the other decision variables is the constraint described in equation 3.36h, which requires $\eta_i = 1$ for any i below the specified level of maintenance. The lower bounding calculations presented here assume that constraints have been propagated for each partial assignment, meaning that the domain of each decision variable has been reduced as much as possible based on assigned values. Therefore, if η_i is unassigned for a particular subsystem, then there exists a completion of the current partial assignment to u that would result in that subsystem being above the specified level of maintenance; if that were not true, η_i would have been assigned 1. Since u_{LL} is the lowest possible level of maintenance, given the partial assignment, any unassigned η_i will be above the level of maintenance specified by u_{LL}, and therefore any assignment to these subsystems is consistent.

The lower bound on corrective maintenance mass is then calculated by creating and evaluating an ORU list using the technique described in Sections 4.1 and 4.2, with the following modifications:

- Mass, mean failure rate, and mass scaling factor are replaced by m', $\bar{\lambda}'$, and γ' as calculated in equations 4.129, 4.130, and 4.131, respectively.

- The overhead factors γ_{odm} and $\gamma_{\Lambda odm}$ are both set to 0.

- Commonality calculations are modified to treat each set of common items as a type of ODM, meaning that they use a shared resource rather than discrete, common spare parts that have characteristics derived from the common set.

- The supportability strategy uses u_{LL} for level of maintenance, ξ_{CM} for ODM, C_{CM} for commonality, $r^{\downarrow}$ for redundancy, and $\eta^{\downarrow}$ for distributed functionality.

4.3.3 Variable and Domain Ordering Heuristics

Different decision variables – both in terms of broad sets of variables such as u or r as well as decisions within each set related to specific components – have different magnitudes of impact on the final solution, as well as on the tightness of the bounds defined in Section 4.3.2. The efficiency of the multiobjective branch and bound approach described in Section 4.3.1 depends on the amount of pruning that occurs during the search, which itself depends in part on the order in which regions of the tradespace are explored. Ideally, Pareto-optimal or near-Pareto-optimal solutions would be identified early in the search, allowing them to be used to prune less optimal regions of the search tree. (When a candidate Pareto frontier is initialized, as described in Section 4.3.1, this may be achieved independently of variable ordering.) In addition, decisions that improve the tightness of bounds can increase the likelihood that those bounds can be used to prune subsequent assignments. This section describes the variable and domain ordering heuristics that are used in this research to guide the search process.

For this research, level of maintenance decisions are considered first, since they are primary drivers of both mass and crew time and an assignment to u imposes significant constraints on the remaining variables, potentially reducing the size of the remaining search tree. Distributed functionality and redundancy are considered next, since these can have a very significant impact on crew time requirements. ODM and commonality decisions are considered last, since these typically have smaller effects.

Within each set of decisions (i.e. u, η, r, ξ, and C), the variables associated with each item are assigned in order of the expected impact of that item's decisions on the overall metrics, approximated by the sum of the expected mass and expected crew time associated with that item, including both corrective and preventative demands. For components, these values can be calculated directly as a function of component characteristics. The value associated with higher-level items is the sum of the values associated with their descendants. Individually, the expected mass and crew time associated with each item are

$$\mathrm{E}[M_\phi] = (I + D)\left(m \circ q \circ \left(\tau \circ \bar{\lambda} \circ \kappa \circ d + \left\lfloor \frac{\tau}{\ell d} \right\rfloor\right) \circ \theta\right) \tag{4.132}$$

$$\mathrm{E}[T_\phi] = (I + D)\left(t \circ q \circ \left(\tau \circ \bar{\lambda}_C \circ \kappa \circ d + \left\lfloor \frac{\tau}{\ell d} \right\rfloor\right) \circ \theta\right) \tag{4.133}$$

The list of ordered indices giving the variable order is then

$$\vartheta = \underset{D}{\arg\operatorname{sort}}\left(\mathrm{E}[M_\phi] + \mathrm{E}[T_\phi]\right) \tag{4.134}$$

where $\arg\operatorname{sort}_D$ returns the list of indices that sort the input in descending order. These indices are used in order to assign η, r, u, and ξ. Since entries of C correspond to two items, they are assigned in order by rows, then by columns – that is, rows i are examined in the order indicated by ϑ, and within each row the columns j are also examined in that order.

There is no clear "best-first" domain ordering in this multiobjective problem, since most decisions have conflicting impacts on mass and crew time. Therefore, the following heuristics are used for each decision:

- Level of Maintenance: u_i are assigned from highest to lowest (i.e. True first, then False).

- Distributed functionality: η_i are assigned from lowest to highest values in their domain, since this approach minimizes crew time and could have a positive or negative effect on mass.

- Redundancy: r_i are assigned from highest to lowest in their domain, since this minimizes crew time and could have a positive or negative effect on mass.

- ODM: ξ_i are assigned from highest to lowest (i.e. True first, then False).

- Commonality: C_{ij} are assigned from lowest to highest (i.e. False first, then True).

4.4 Summary

Chapters 3 and 4 present the overall methodology used in this research, which takes in a description of a system and mission of interest and identifies and characterizes the Pareto-optimal set of supportability strategies that minimize mass and crew time requirements, where a supportability strategy is defined as a set of decisions that determine:

- The level at which maintenance is implemented in the system when a failure occurs (i.e. the definition of an ORU);
- Which items are common with each other, meaning they can share spare parts and are covered by a single ORU;
- Which ORUs use ODM to produce spares when needed, and which use the traditional approach of having stored spare parts on hand;
- Which subsystems have distributed functionality, meaning that they are split into multiple modules that share the load associated with particular system functions rather than being gathered into a single, monolithic system; and
- Which components have redundant instances that allow ORUs containing them to continue functioning after a failure occurs.

The encoding scheme presented in Section 3.2 describes the inputs, objectives, constraints, and decision variables used in the supportability strategy optimization problem, which is summarized in equation 3.36 on page 148. Section 4.1 presents the process for creating and characterizing the ORU list – that is, the list of items that cause demands for supportability resources like spares, maintenance items, and crew time, as well as their supportability characteristics – based on the system description and the selected supportability strategy. This ORU list is the primary input of the supportability evaluator described in Section 4.2, which assesses mass and crew time requirements for the specified mission. Section 4.3 presents the general optimization strategy used to solve the multiobjective COP, including a description of the multiobjective branch and bound algorithm, methods for calculating lower bounds on mass and crew time for a given partial assignment, and heuristics for variable and domain ordering. This formalization of supportability strategy optimization allows the exploration and optimization of a wide range of strategies using various COP solvers.

4.4.1 Contributions

The key contributions of this methodology, which have been discussed in the above text where relevant, are summarized here:

- A formal definition of the supportability strategy optimization problem, including

    - the definition of sets of decision variables that can be used to define a wide range of supportability strategies,
    - a mathematical method for using those variables along with a system definition to define an ORU list and evaluate its supportability characteristics, and
    - a set of heuristics that can be used to bound the supportability characteristics of a partial assignment to the decision variables.

  Taken together, all of these contributions enable the application of formal multiobjective optimization techniques to supportability strategy optimization.

- The inclusion of crew time demands as a separate stochastic process from random component failures, which enables a more complete evaluation of maintenance crew time that captures time spent on inspection and troubleshooting activities as well as on removal and replacement of failed ORUs.

- The inclusion of the probability of sufficient crew time in the evaluation of overall POS, and resulting adaptation of spares allocation optimization to account for crew time POS, which enables a more complete assessment of the risk associated with maintenance resource availability.

- A model for ODM feedstock requirements that evaluates the POS associated with a given amount of feedstock, captures the effect of commonality of material on maintenance logistics requirements, and enables the inclusion of ODM feedstock in spares allocation optimization, thereby enabling the consideration of ODM as a supportability strategy alongside more traditional approaches.
- A supportability evaluation model that assesses the distribution of required crew time, alongside all other required metrics, without the use of Monte Carlo simulation, thereby facilitating the exploration and optimization of large supportability strategy tradespaces.

4.4.2 Implementation

This methodology is implemented in a Python-based analysis tool called MCROSS, which utilizes a Python supportability strategy evaluator developed based on the methodology described in Section 4.2 with MiniZinc [117, 125], a solver-agnostic COP modeling language. The interface between Python and MiniZinc is facilitated by the PyMzn package [118]. For the purposes of this research, the feasibility constraints of the supportability strategy optimization problem (constraints 3.36b through 3.36t) are encoded in MiniZinc, which is then used to call OpSat [119, 120]. OpSat then executes the search, passing partial and full assignments to a Python evaluator and receiving bounds and objective values as needed. When pure enumeration is used, Gecode [121], an open-source COP solver is used to enumerate all feasible strategies, which are then evaluated using Python. The Pareto frontier is then extracted from the resulting tradespace.

One key enabler of this implementation is communication between OpSat and the Python supportability evaluation module developed as part of this research. Initial efforts to encode the objective function and bounding functions in MiniZinc were not effective, since that language does not currently include efficient representations of probability distributions, IDFT, or other elements of the evaluation methodology. However, these were readily encoded in Python. Python also allowed easy access to a range of libraries to enable rapid, efficient calculation. While it may be possible to encode the entire supportability strategy optimization problem and evaluation module in MiniZinc, for the purposes of this research it was much more effective to use Python, and as a result OpSat's ability to call external programs to evaluate objective values and bounds was critical for accelerating the development of MCROSS.

Chapter 5

Verification and Validation

This chapter presents a verification and validation of the MCROSS methodology described in Chapters 3 and 4. Section 5.1 compares model outputs and intermediate results (e.g. ORU-level POS assessments) to Monte Carlo simulation. Section 5.2 then validates the model in two ways by comparing model outputs to EMAT, one of the most capable space mission supportability analysis tools identified in the literature review (Chapter 2), as well as to real-world maintenance demands on the ISS. Finally, Section 5.3 summarizes the results of this chapter. In addition to the model validation presented in this chapter, the optimization strategy itself is also validated via comparison to full enumeration using the notional case study presented in Chapter 6.

5.1 Verification

This section verifies three key elements of the supportability model described in Section 4.2: the negative binomial approximation used to enable closed-form POS calculations, the expected value / variance-based technique for calculating ORU characteristics based on the components within them and determining the impacts of redundancy, and the numerical characteristic function inversion approach used to model crew time and ODM feedstock requirements. In each case, model outputs are compared to results from large-scale Monte Carlo simulation for example problems, and differences are characterized and discussed. A summary of these verification results is presented at the end of this section.

5.1.1 Negative Binomial Approximation

One of the key assumptions made in this research is that failure processes can be approximated using a negative binomial distribution, or a Poisson distribution with a gamma-distributed failure rate. Most input data characterizes failure rate uncertainty using a lognormal distribution, meaning that the distribution of failures would follow a Poisson-lognormal distribution. However, as discussed in Section 4.2.1, the Poisson-lognormal distribution does not have a closed-form CDF or characteristic function. This section examines the differences between lognormal and gamma-distributed failure rates sharing the same expected value and variance, as well as the differences in the distribution of the number of failures observed under each model.

Figure 5.1 shows a comparison between the lognormal and gamma failure rate distributions for a component with a mean failure rate of 1×10^{-5} and an error factor of 2. Both distributions have the same expected value and variance, since the parameters of the gamma distribution are defined via the moment-matching technique described in Section 4.2.1. The mean and error factor values for this example are selected to be representative of space systems. In particular, an error factor of 2 represents a medium amount of uncertainty, meaning that the 95th percentile failure rate is twice the median value. As Figure 5.1 shows, the gamma approximation tracks fairly closely with the lognormal CDF. Figure 5.2, which shows the same distributions but with the error factor set to 3 – a much higher level of uncertainty. In this case, the difference between the two CDFs is more apparent. Overall, error factor is the primary driver of the magnitude of the difference between the lognormal and gamma distributions, as shown by Figures 5.3 and 5.4.

Figure 5.3 shows the difference between the two CDFs – i.e. the value of the lognormal CDF minus that of the gamma CDF – for a range of error factors. In all cases, the mean failure rate is 1×10^{-5} and the difference between the CDFs follows the same pattern. The gamma approximation first overestimates, then underestimates it,

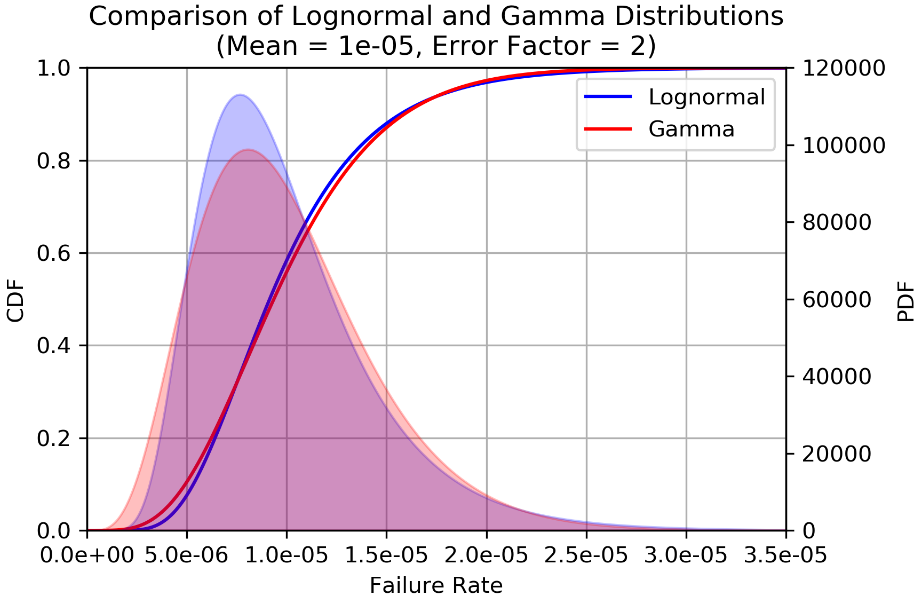

Figure 5.1: Comparison of lognormal (red) and gamma (blue) distributions with a mean of 1×10^{-5} and an error factor of 2. CDFs are plotted as solid lines, while the associated PDF is shown as a shaded region.

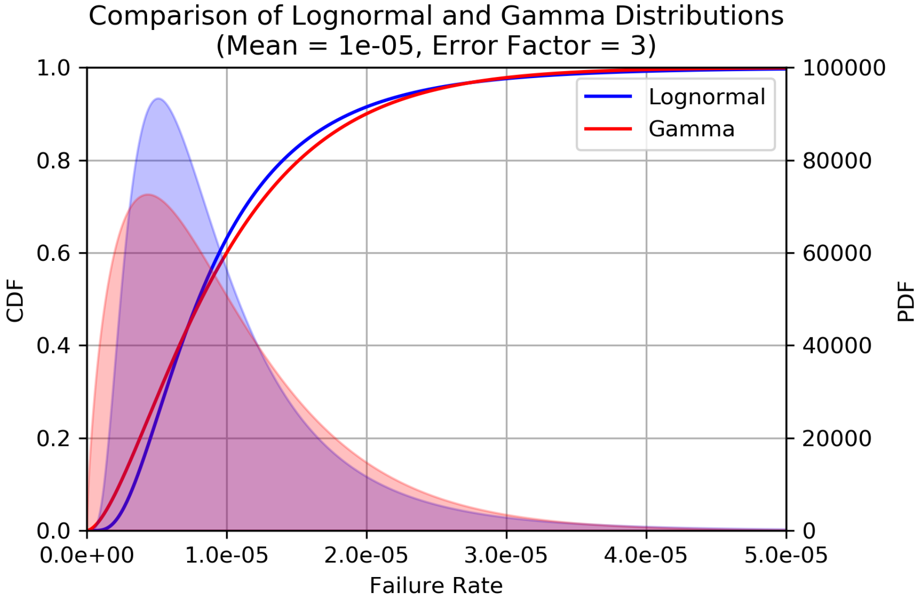

Figure 5.2: Comparison of lognormal (red) and gamma (blue) distributions with a mean of 1×10^{-5} and an error factor of 3. CDFs are plotted as solid lines, while the associated PDF is shown as a shaded region.

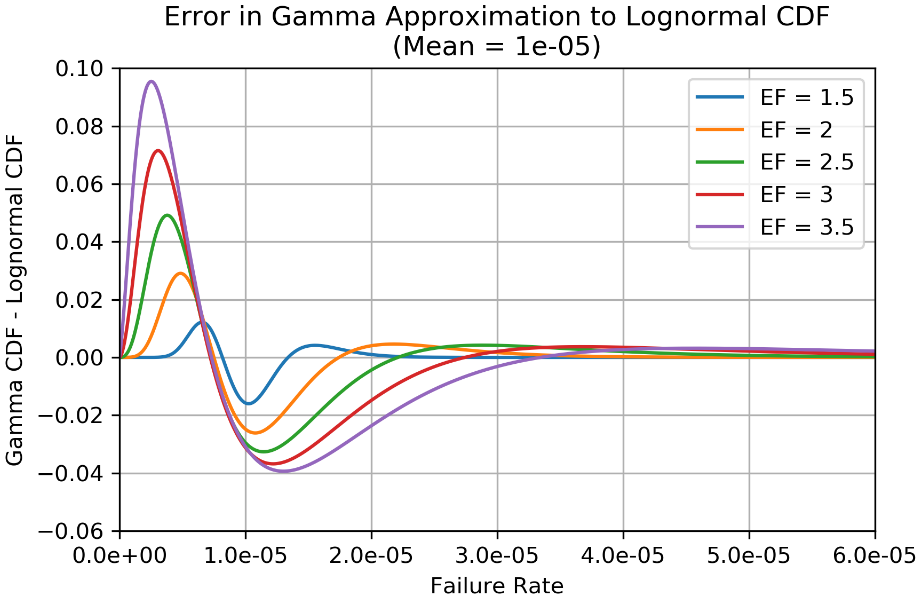

Figure 5.3: Error in the gamma approximation to the lognormal CDF for a range of error factors. The mean failure rate is 1×10^{-5} in all cases.

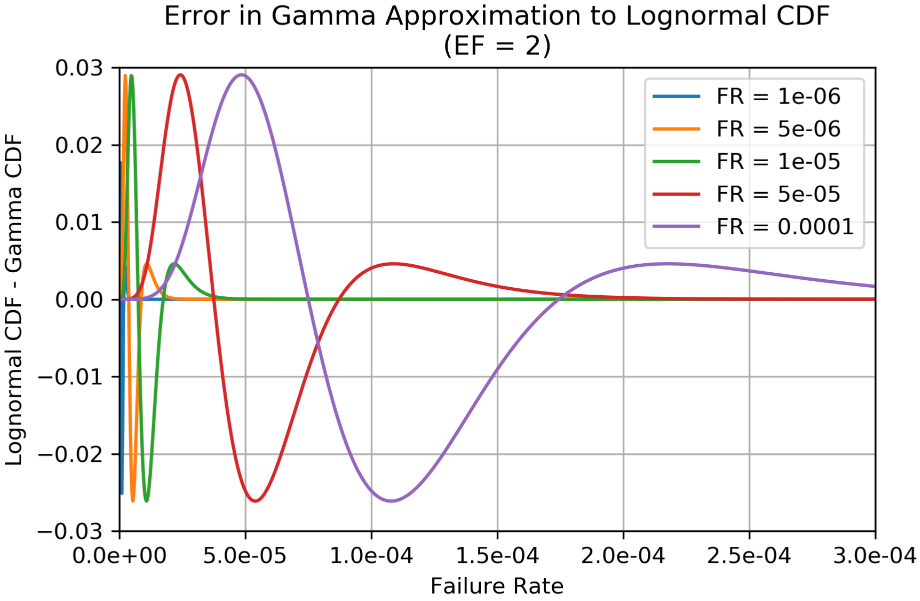

Figure 5.4: Error in the gamma approximation to the lognormal CDF for a range of mean failure rates. The error factor is 2 in all cases.

then overestimates the lognormal CDF. The overestimate peaks in the early portion, at values less than half of the mean, while the overestimate peaks just after the mean value. The magnitude of the difference decreases for higher values, and is proportional to the error factor. At an error factor of 3.5 – a high level of uncertainty – the largest difference between the two is approximately 0.095.

Figure 5.4 shows the same data while varying the mean failure rate instead while keeping the error factor fixed at 2. In this case, the pattern remains the same, and the magnitude of the largest difference between the two is approximately the same across the range of failure rates, between approximately 0.29 and -0.26. The main effect of varying mean failure rate is to stretch the distributions, shifting the location of peaks and valleys in the error along the x-axis.

The effect of this difference in failure rate distributions on the distribution of the number of failures observed is shown in Figures 5.5 through 5.7. Figure 5.5 shows the POS for a notional item as a function of the spares provided, calculated using both the Poisson-lognormal model (evaluated using 1,000,000 Monte Carlo simulations) and the negative binomial approximation used for this research. The mean failure rate is 1×10^{-5}, error factor is 3 (i.e. a relatively high amount of uncertainty), and the mission endurance is 1,200 days. The values are almost indistinguishable, indicating that the negative binomial provides a close approximation, at least for these values of mean, error factor, and endurance.

Figures 5.6 and 5.7 show the error in POS calculation (defined as the POS calculated by the negative binomial approximation minus that calculated via Poisson-lognormal Monte Carlo simulation) for a range of mean failure rate and error factor values. Since the number of spares required to achieve a given POS value can vary significantly in all of these cases, the error is plotted against the POS calculated by the negative binomial approximation. Each point along each curve has an x-value equal to the POS calculated by the negative binomial approximation and a y-value equal to the error in that approximation for 0, 1, 2, etc. spares, starting from the first point on the left. Error in the negative binomial approximation is proportional to both error factor and mean failure rate, but the magnitude of error remains below

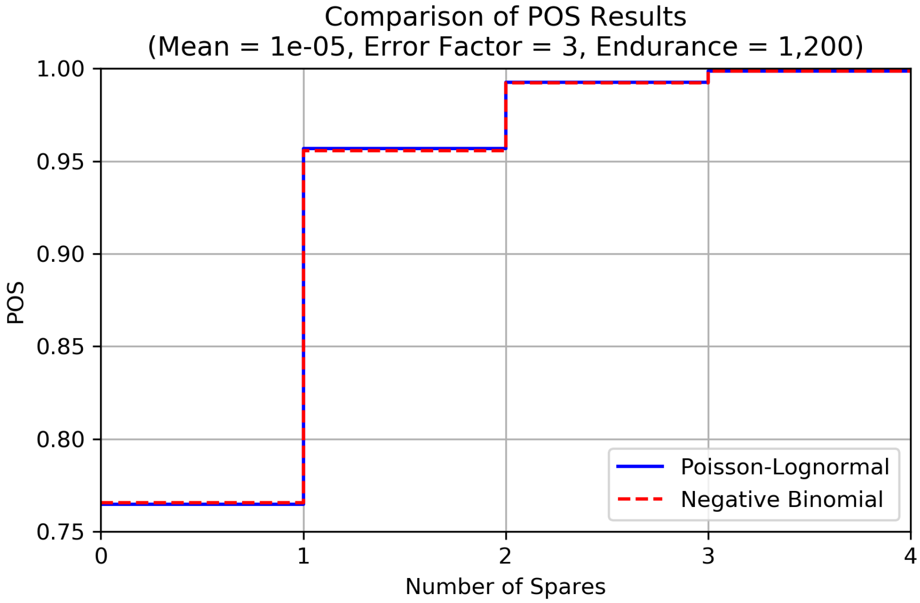

Figure 5.5: Comparison of POS values calculated using Poisson-lognormal and negative binomial models.

0.006 for all values examined here.

The difference between negative binomial and Poisson-lognormal probabilities shrinks as higher POS values are examined. Since most supportability analysis applications are interested in the evaluation of spares associated with high POS levels, and since the presence of multiple spares within a system requires each spare to have a POS much higher than the required value, this indicates that the negative binomial approach provides a close approximation for supportability applications.

Figures 5.6 and 5.7 do indicate, however, that the level of error present increases proportionally to both error factor and failure rate. As a result, analyses which include items with high failure rates or error factors are likely to see the greatest error. Error in the approximation follows a similar pattern, in general, to the one described above (for lognormal vs. gamma distributions) in that the negative binomial first overestimates, then underestimates, then overestimates Poisson-lognormal probabilities. The only exception in the cases examined here was the case with an error factor of 1.5 in Figure 5.6 (i.e. the lowest error factor), which had the lowest level of error but remained an underestimate across the entire range. At higher error factors,

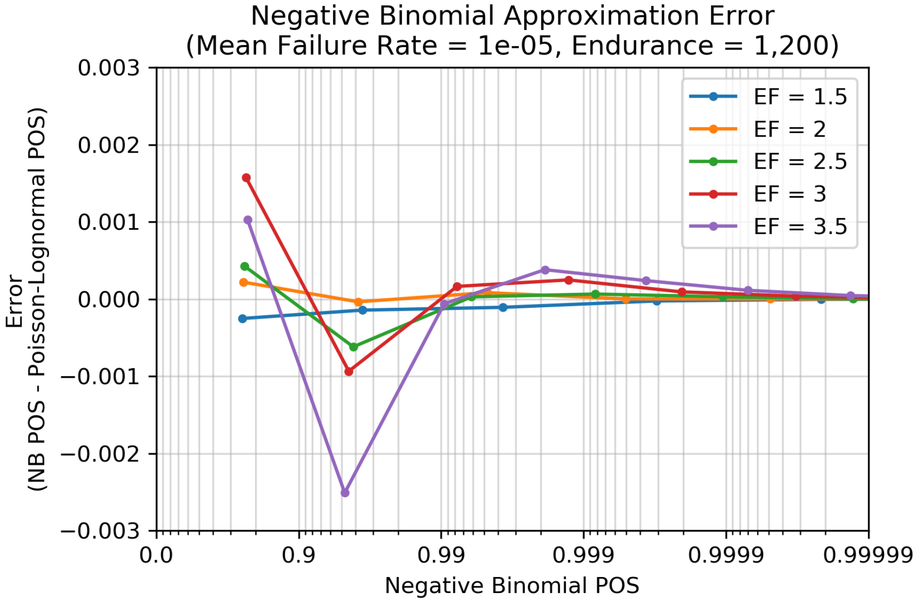

Figure 5.6: Error in negative binomial approximation to Poisson-lognormal probabilities for a range of error factors.

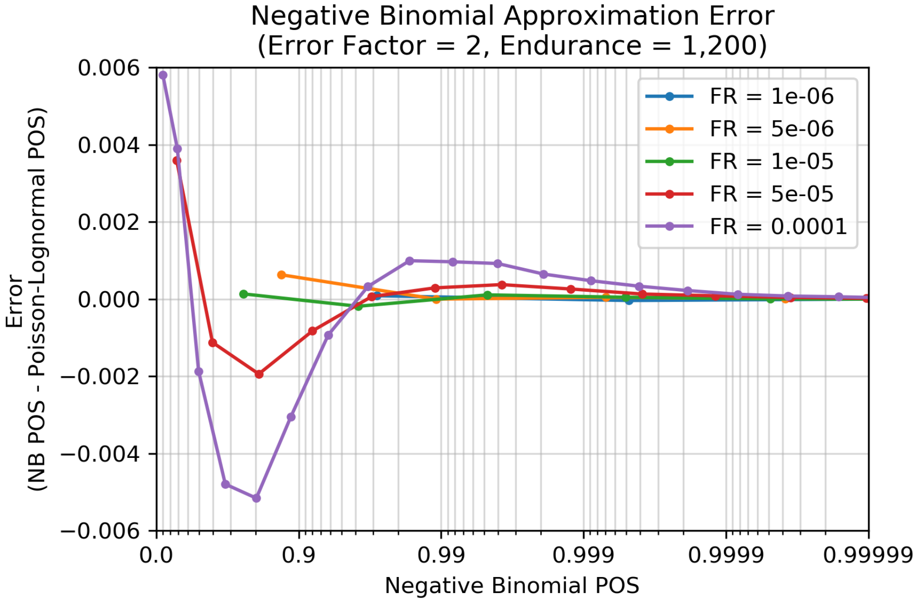

Figure 5.7: Error in negative binomial approximation to Poisson-lognormal probabilities for a range of failure rates.

however, these results indicate that, at the high end of the POS range, the negative binomial approximation slightly underestimates the risk associated with a particular application (since it overestimates POS); as a result, spares allocations identified using the negative binomial approach may not quite achieve the required POS when evaluated using the Poisson-lognormal model. However, the difference between the two is small (on the order of less than 0.1% at that range).

Overall, these results indicate that the negative binomial model provides a close approximation for Poisson-lognormal probabilities in this range. Since both gamma and lognormal models are used in practice to represent failure rate uncertainty, and there is a significant amount of other uncertainty present in system parameters during early supportability analysis, the low level of error identified in these validation cases is considered acceptable for this application.

5.1.2 Combining Components into ORUs, with Redundancy

This section examines the method used to combine components into ORUs and implement redundancy. As discussed in Section 4.1.1, the failure and crew action rate characteristics of a given replaceable item (and, therefore, an ORU) are defined by adding the failure and crew action rates of its constituent components using expected value and variance. To explore this process, a notional ORU is constructed from three notional components with mean failure rates of 5×10^{-5}, 1×10^{-5}, and 5×10^{-6} and error factors of 2, 2.5, and 3, respectively. The mission endurance is again assumed to be 1,200 days. The POS calculated using the method described in Section 4.2.1 is then compared to one calculated using Monte Carlo simulation of Poisson-lognormal probabilities, with failures occurring at the component level and aggregated to the ORU level. A sensitivity analysis is also performed on the level of redundancy provided for the first component. (For the purposes of this assessment, the redundancy effectiveness factor is assumed to be 1.)

Figure 5.8 shows the POS for the notional ORU described above as a function of the number of spares provided, evaluated using both Monte Carlo simulation at the component level and the model described in Section 4.2.1. 1,000,000 Monte Carlo

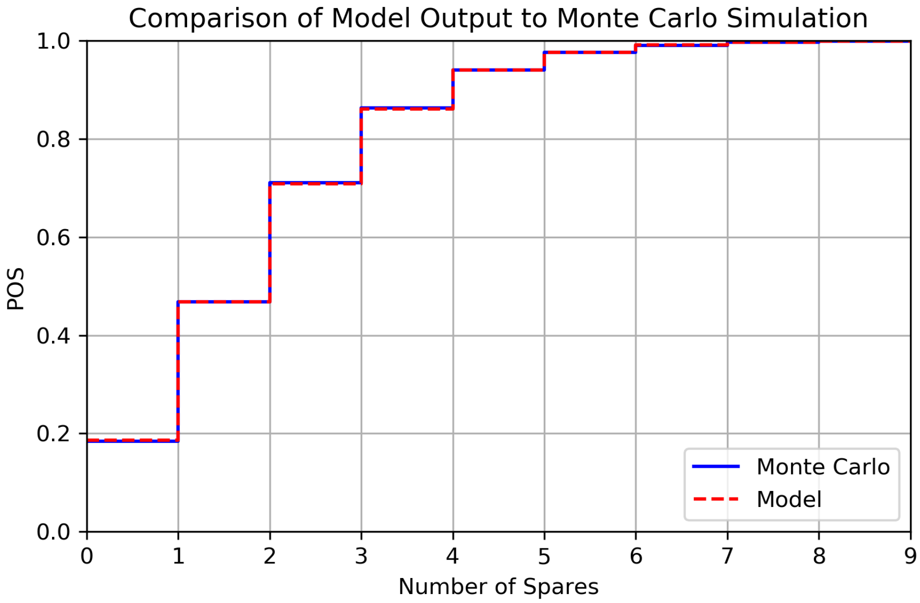

Figure 5.8: Comparison of POS values for combined components, calculated using the method described in Section 4.1.1.

simulations were evaluated to assess POS. As with Figure 5.5, the values calculated using the model are nearly indistinguishable from those derived from Monte Carlo simulation, indicating that the model provides a good representation of the case with no redundancy. The largest error in the case evaluated here is approximately -0.001, corresponding to 0 spares.

Figure 5.9 uses the same approach as Figures 5.6 and 5.7 to show the relationship between the level of redundancy provided to the first component (i.e. the one with a failure rate of 5×10^{-5} and an error factor of 2) and error in the POS approximation. These results indicate that redundancy can have a significant impact on the validity of POS approximations calculated using the approach described in Section 4.1.1, particularly at the lower ends of the POS range. Specifically, the model used here significantly overestimates POS when redundancy is present for POS values below 0.9. However, the amount of error in the approximation decreases rapidly as higher POS values are examined, and for the range of redundancy levels examined here error is less than 0.03 for POS values above 0.9 – and as noted before, this is typically the range of interest for supportability analyses.

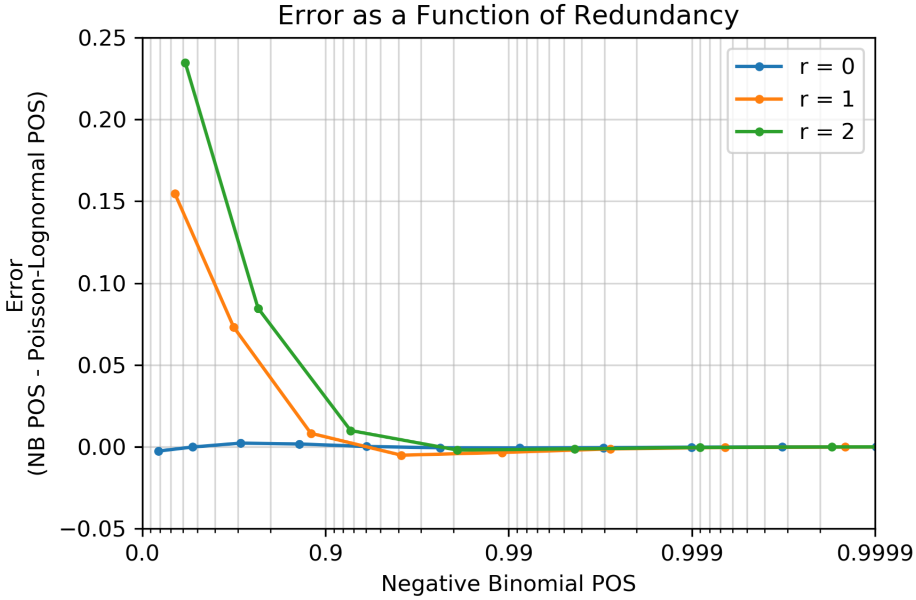

Figure 5.9: Error in POS calculation for a range of redundancy values.

5.1.3 ODM / Crew Time Model

This research uses the same approach to evaluate the distribution of crew time (Section 4.2.2) and the distribution of ODM feedstock (Section 4.2.3), namely numerical characteristic function inversion based on the approach presented by Warr [130]. Therefore, for the purposes of verification, the models are treated together. A notional set of five ORUs were examined, using the rates and amounts described in Table 5.1. The baseline step size (i.e. Δt or Δm) is set to 0.3, and the numerical inversion error parameter ϵ_{max} is set to 1×10^{-8}. The rates and amounts listed are meant to be generic and representative of typical values for space systems applications; for the ODM model they would represent failure rates and ORU masses, while for the crew time model they would represent crew action rates and maintenance crew time requirements. In addition, the amounts and step size are selected to simulate a challenging case for this discretized approximation, since the amounts are not multiples of the selected step size.

Figure 5.10 shows the PMF and CDF of the total amount required for the set described in Table 5.1, as calculated by the model described in Section 4.2 and by

Table 5.1: ODM / crew time model verification case parameters.

ORU	Rate Expected Value	Rate Variance	Amount
1	2.0×10^{-4}	4.5×10^{-10}	0.6
2	3.5×10^{-5}	1.0×10^{-9}	1.6
3	8.5×10^{-6}	8.0×10^{-10}	1.4
4	9.0×10^{-5}	1.7×10^{-9}	0.8
5	5.0×10^{-5}	7.0×10^{-10}	2.0

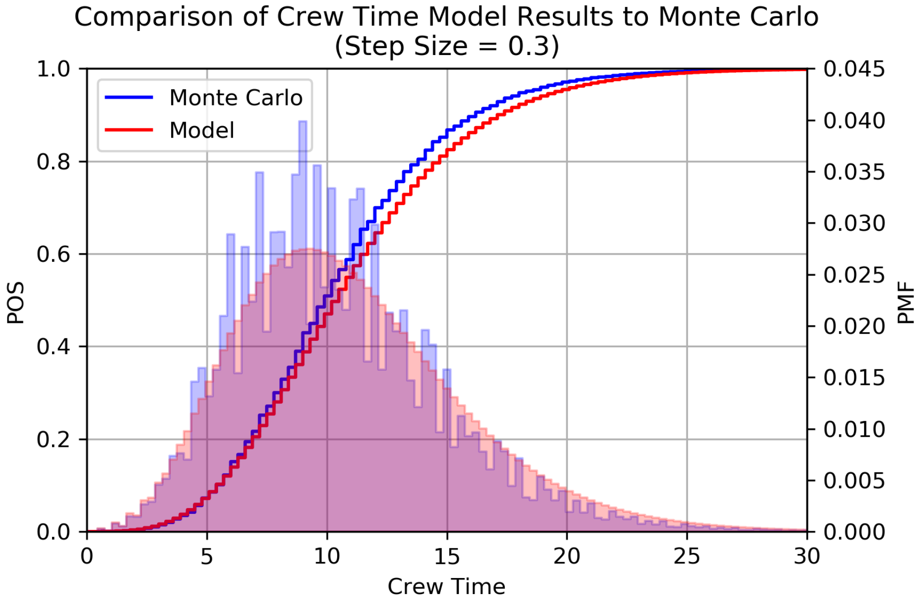

Figure 5.10: Comparison of ODM / crew time model output to Monte Carlo simulation, with a step size of 0.3.

Poisson-lognormal Monte Carlo simulation, using a population of 1,000,000 simulations, as before. These results show that the model tends to underestimate POS for a given amount of crew time or ODM feedstock. As mentioned earlier, the step size was deliberately selected to not be a multiple of the amounts under consideration. As a result, much of the error in this approximation results from discretization. The characteristic function inversion error ϵ_{max} only bounds error in the numerical characteristic function inversion process itself, not error resulting from discretization or error from the use of negative binomial random variables rather than Poisson-lognormal random variables.

The impact of various levels of discretization on model accuracy is shown in Figure

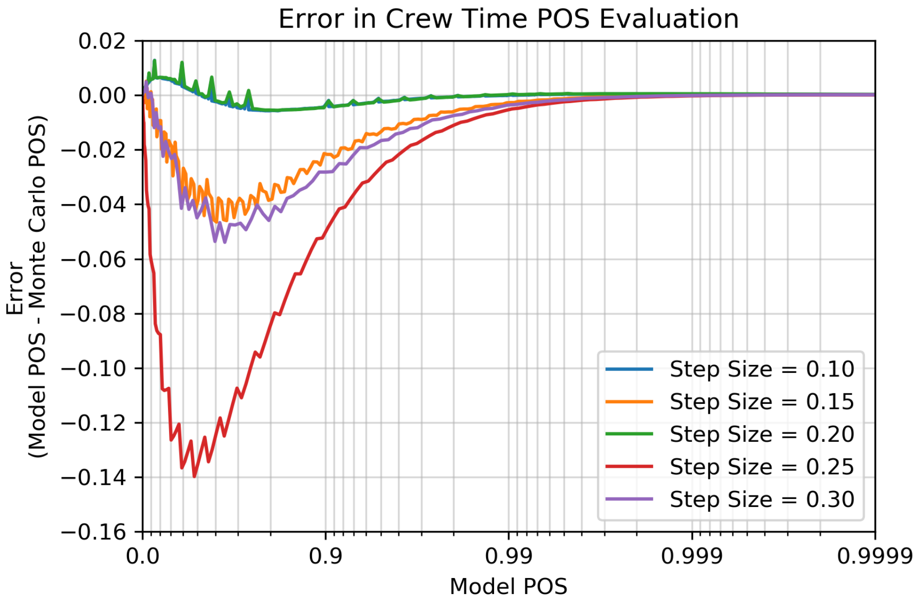

Figure 5.11: Error in ODM / crew time POS evaluation for a range of step sizes.

5.11, which plots the difference between model output and Monte Carlo results against the POS calculated by the model for a range of step sizes. The purple line corresponds to a step size of 0.3, the results of which are shown in Figure 5.10. In all cases, error decreases as higher POS values are investigated, with the magnitude of error dropping below 0.05 for POS greater than 0.9. In addition, nearly all cases show the model underestimating POS, with the exception of POS values below 0.6 at step sizes that have low error in general. This means that the model is conservative for the purposes of risk assessment and resource allocation. In the case of crew time, a slight underestimate of the probability of sufficient crew time will mean that solutions satisfying the total POS constraint specified in equation 3.33 on page 144 based on model output would also satisfy that constraint if a more detailed model were used. For the purposes of ODM feedstock allocation, this conservatism means that the model will allocate slightly more feedstock than is required, rather than less, depending on the level of error.

Interestingly, a smaller step size does not necessarily mean reduced error. In the cases examined here, the largest error comes from a step size of 0.25. Compared to a step size of 0.3, this smaller step size more than doubles the amount of error in

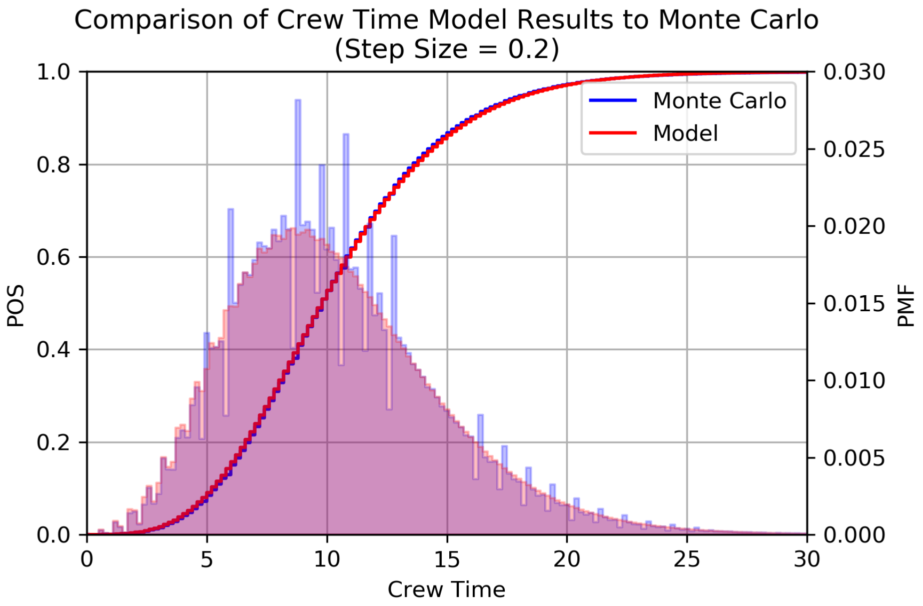

Figure 5.12: Comparison of ODM / crew time model output to Monte Carlo simulation, with a step size of 0.2.

the model output, to -0.14 at a POS of approximately 0.5. The reason for this is that while there is an ORU in this case for which the amount is a multiple of 0.3 (i.e. ORU 1), there are no ORUs with amounts that are multiples of 0.25. Overall, the most accurate assessments occur at step sizes of 0.1 and 0.2, since all amounts in this case are multiples of these values. In both of these cases, the magnitude of error remains below 0.125 across the entire POS range, and below 0.005 for POS greater than 0.9. The CDF and PMF resulting from a step size of 0.2 are shown in Figure 5.12 for comparison. These results show that discretization is a significant driver of error in ODM mass and crew time evaluation, and that it is important to select a discretization step size that is a common factor of as many ORU masses or maintenance crew times as possible.

The expected amount of resources required output by the model is 10.37, which matches the sum of the expected amounts associated with each of the ORUs in Table 5.1. Expected corrective maintenance crew time is calculated directly using a closed-form expression (equation 4.72) rather than a numerical evaluation, and as a result the outputs of the expected crew time model are not affected by discretization or any other parameters.

5.1.4 Summary

The above sections indicate that, in most circumstances, the outputs of the model used in this research closely match results obtained via Monte Carlo simulation. The negative binomial approximation produces POS results that closely match those from Poisson-lognormal Monte Carlo simulation, particularly in the high-POS range that is typically of interest in supportability evaluation. When aggregating components to define replaceable items and ORUs, summation of failure rates (or crew action rates) based on expected value and variance produces results that also closely match Monte Carlo simulation when no redundancy is present. When redundancy is present, model results and Monte Carlo results can diverge significantly (with POS error of 0.15 to 0.25) in the low-POS range (i.e. POS less than 0.9). However, as with all other sources of error in the model, the error introduced by the redundancy approximation declines significantly as higher POS values are evaluated, and falls below 0.02 for POS values above 0.9 in the case examined here. As a result, while improvement to the redundancy model is recommended as part of future work (described in Chapter 9), the current approximation is considered appropriate for supportability evaluation for POS greater than 0.9. Finally, the numerical characteristic function inversion approach used to model crew time and ODM feedstock requirements is shown to closely agree with Monte Carlo results if the selected step size is a common factor of the amounts being examined. If not, the model tends to underestimate POS, but the error in the approximation decreases as higher POS values are examined.

5.2 Validation

This presents a validation of the corrective maintenance crew time and mass model described in Section 4.2, implemented in MCROSS, via comparison to an existing National Aeronautics and Space Administration (NASA) supportability analysis tool and to actual flight maintenance data from the ISS. Specifically, approximately eight years of flight maintenance data for ISS water processing and oxygen generation systems were compiled from MDC, representing the entire fully-operational time period for those systems between November 16, 2008 and November 20, 2016. The first half of this dataset is used to define ORU failure rate and crew action rate parameters by performing a Bayesian update on initial estimates provided in MADS, using a similar approach to that used by ISS L&M. Other MADS data, such as ORU masses and duty cycles, are also used to fully define the validation case ORU list. The second half is then used as a test dataset. Crew time and spares requirements are evaluated using both MCROSS and EMAT, and the outputs of both models are compared to each other as well as to actual observed maintenance demands on the ISS. EMAT (described in greater detail in Section 2.1.5) is one of the most capable space mission supportability analysis tools currently in operation, and is regularly used by the SMAB at NASA Langley Research Center to assess supportability for mission concepts and inform technology investment, system development, and mission architecture decisions across the Agency. Statistical analysis is used to assess the validity of inputs (e.g. failure rates derived from the first four years of observed operations) as well as the validity of MCROSS outputs.

The validation presented here focuses on corrective maintenance activities, and does not include installed system mass or preventative preventative maintenance. This is due to the fact that these latter two factors are deterministic and typically well-understood for a given system; the most challenging aspect of supportability evaluation is the assessment of stochastic maintenance demands. In addition, the validation case examines a subset of ISS ECLSS (described in Section 5.2.1). The results shown here only include the crew time and mass associated with these systems, and should not be interpreted as representing the total maintenance crew time and mass associated with the ISS, or the ISS ECLSS. However, for the purposes of validation a comparison of results for a subset of the system is sufficient. Finally, while MCROSS is designed to examine a range of supportability strategies, this analysis examines only one supportability strategy – that is, the one resulting in the ORU set deployed to the ISS. This is due to the fact that EMAT is designed to examine a single ORU list, rather than a range of different supportability strategies, and the fact that the only operational data available correspond to the ORUs which were deployed to the ISS. That is, space station operational data only provides data for validation in the case of the supportability strategy used on the station, and that is the case examined here.

Section 5.2.1 describes the validation case examined here, along with the sources of raw input data used and the overall scope of the analysis, as well as the Bayesian approach used to define required parameters based on input data and the strategy used to compare model outputs to flight data. Section 5.2.2 examines the validity of the input parameters for MCROSS and EMAT – that is, the failure rates, crew action rates, and maintenance crew times for each ORU – on an ORU-by-ORU basis against observed behavior during the test period. Section 5.2.3 examines crew time results from MCROSS and EMAT in comparison to each other and to ISS actuals, and Section 5.2.4 does the same comparison for mass.

5.2.1 Data and Analysis Scope

The validation case examines a set of 30 ORUs from the ISS water recovery and oxygen generation systems, consisting of Water Processor Assembly (WPA), Urine Processor Assembly (UPA), and OGA ORUs for which the data required for analysis – specifically, an initial failure rate estimate (mean and error factor), duty cycle, mass, and estimated maintenance crew time – are listed in MADS. This includes all major ORUs from these three systems except for the WPA External Filter Assembly (EFA), UPA Advanced Recycle Filter Tank Assembly (ARFTA), and OGA ACTEX filter assembly, which were added to these systems after they were deployed to the ISS and are missing some or all of the required data in MADS. A description of each of these systems and their functions is presented by Carter et al. [140], Takada et al. [105], and Bagdigian et al. [58]. A full list of ORUs included in this validation is presented in Table 5.2.

It is important to note that these three subsystems (WPA, UPA, and OGA) do not encompass the entire ISS ECLSS, and the mass, crew time, and risk results presented

Table 5.2: List of 30 ORUs used in validation cases. These include all major ORUs from the ISS water recovery and oxygen generation systems except the WPA EFA, UPA ARFTA, and OGA ACTEX filter, which are not included due to a lack of data.

Subsystem	ORU
WPA	Wastewater ORU
	Pump/Separator ORU
	Separator Filter ORU
	Particulate Filter ORU
	Multifiltration Beds
	Sensor ORU
	Catalytic Reactor ORU
	Gas Separator ORU
	Reactor Health Sensor ORU
	Ion Exchange Bed ORU
	Water Storage ORU
	Water Delivery ORU
	Process Controller
	Oxygen Filter
	Microbial Check Valve
UPA	Wastewater Storage Tank Assembly (WSTA)
	Fluids Control and Pump Assembly (FCPA)
	Distillation Assembly (DA)
	Pressure Control and Pump Assembly (PCPA)
	Separator Plumbing Assembly (SPA)
	Firmware Controller Assembly (FCA)
OGA	Water ORU
	Inlet Deionizing Bed ORU
	Hydrogen ORU
	Oxygen Outlet ORU
	Hydrogen Sensor ORU
	Pump ORU
	Nitrogen Purge ORU
	Process Controller
	Power Supply Module

in this section only account for those three subsystems. The scope of this validation is restricted to these systems in order to maintain analysis scope at a reasonable level. While maintenance data are available on other ECLSS systems (as well as spacecraft systems beyond ECLSS), the three subsystems examined here represent the ECLSS subsystems for which maintenance data were well-organized and labeled, and therefore could be processed to generate the inputs required for this validation without requiring excessive data compilation, organization, and labeling. The data associated with these 30 ORUs is considered sufficient for the purposes of model validation, and more in-depth and comprehensive validation is proposed as part of future work (described in Chapter 9).

It is also important to note that the ORUs used here are based on current ISS systems, and do not necessarily represent the systems that would fly on future missions. In fact, the driving motivation for this research area is to inform the development of new systems, or next-generation versions of current systems, in order to optimize supportability to meet the challenges of future human spaceflight. The results shown in this section – as well as the results in all other case studies presented as part of this research – indicate what the supportability characteristics of the system *would likely be* if the given set of ORUs were used on notional future missions. These supportability characteristics will almost certainly change as systems mature, new designs or maintenance strategies are introduced, and problems are identified and addressed. The purpose of supportability analyses such as these is not to predict the future, but rather to forecast what the future might look like under various circumstances in order to inform the development of more effective systems.

Data Sources

The primary data sources used for this analysis are MADS and MDC. MADS lists key characteristics for each ORU, including mass, duty cycle, estimated maintenance crew time (captured as Crew Member Mean Time To Repair (CMMTTR)), and K-factor, as well as an initial failure rate estimate, characterized by MTBF (i.e. the inverse of the mean failure rate) and error factor. Crew action rates are characterized in MADS. As noted in Section 2.5, previous supportability models use failure rates to assess crew time rather than considering crew time demands as a separate process; the use of a separate crew action rate in supportability modeling is a contribution of this research. For the purposes of this validation, the initial crew action rate estimate is assumed to be the same as the initial failure rate estimate. MDC is a record of maintenance actions on the ISS, covering the time period from the start of crewed ISS operations in 1998 through November 20, 2016. Each entry describes the event, the associated items, any corrective action that was taken, the date of the event (e.g. failure, anomaly, etc.), the date of the maintenance action, and the amount of crew time used. Events are also categorized as corrective or preventative, and information is provided in the maintenance action description to distinguish between R&R events that involved the use of a spare and non-R&R events that only used crew time.

OGA was delivered to the ISS aboard STS-121 (ULF 1.1), which launched on July 4, 2006, and docked with the station on July 6. The system operated intermittently starting in July 2007, but operations were limited by water availability until the WPA and UPA were delivered aboard STS-126 (ULF 2), which launched on November 14, 2008 and docked on November 16 [58, 141–143]. Therefore, for the purposes of this analysis the start of full operations for these three subsystems is taken to be November 16, 2008.

To gather the data required for this validation, MADS data and MDC entries associated with each ORU in the WPA, UPA, and OGA for the time period between November 16, 2008 and November 20, 2016 were compiled and organized. Entries were cross-referenced against an ISS Reliability and Maintainability Working Group (RMWG) failures list (used for ISS program Bayesian updates) as well as descriptions in literature where available in order to separate random events from scheduled, preventative maintenance and wearout failures (i.e. failures occurring due to ORUs reaching the end of a limited lifetime). The RMWG dataset is not used directly for this analysis because it does not record the crew time associated with each maintenance action and therefore does not provide all information required for maintenance crew time assessment. For the purposes of this analysis, random events are considered to be any unscheduled maintenance action, whether driven by internal random failure or induced by external factors. Internal failures are characterized by the failure rate, while induced failures are characterized by the K-factor associated with an item, but the Bayesian update methodology applied here does not distinguish between the two since both are simply factors driving the overall effective failure rate.

As noted earlier, this validation focuses on corrective maintenance – i.e. random events – and does not consider preventative maintenance actions. Therefore, preventative maintenance events are removed from the dataset and the remaining events are characterized as either R&R or Non-R&R events, depending on whether a spare ORU was expended during maintenance. Finally, events were also associated with the relevant ORU, based on the MDC maintenance description. The end result of this data compilation was a set of supportability characteristics and a maintenance history for each ORU in Table 5.2.

In some cases, multiple MDC entries exist for the same event, each with different crew time values. These indicate instances where multiple astronauts were involved in a maintenance activity and/or the activity was spread across multiple sessions of different lengths. In these cases, multiple entries were combined into single events with a total crew time equal to the sum of the crew times associated with each entry, and a date corresponding to the date of the initiating event in order to link crew time or spares expenditures to the failure or crew time event that initiated the maintenance action.

The characteristics of the MDC data gathered for this validation case are summarized in Table 5.3. A total of 79 corrective maintenance actions associated with the ORUs listed in Table 5.2 were identified for the time period between November 16, 2008 and November 20, 2016, representing a total of 261.1 CM-h of crew time. 40 of the recorded actions, or 51%, correspond to R&R of failed ORUs, accounting for a total of 192.9 CM-h (74%) of all corrective maintenance crew time spent on these systems over this time period. 40 crew actions (49%) did not involve R&R of a failed ORU; these account for 68.2 CM-h, or 26% of maintenance crew time. These results indicate the importance of separating the parameters associated with the rate

Table 5.3: Summary of MDC data used for validation, consisting of all corrective maintenance actions between November 16, 2008 and November 20, 2016 for the ORUs listed in Table 5.2.

Category	Number of Events		Crew Time (CM-h)	
R&R	40	(51%)	192.9	(74%)
Non-R&R	39	(49%)	68.2	(26%)
Total	79		261.1	

of ORU spares demand from the rate of crew time demand; for the systems examined here, over a quarter of non-preventative maintenance crew time – that is, crew time spent in response to random, unscheduled events – used did not involve R&R of a failed ORU or use of a spare.

Validation Approach

The MDC data gathered for this research describe 2,927 days (just over eight years) of WPA, UPA, and OGA operations between November 16, 2008 and November 20, 2016. While MADS does include a current failure rate estimate for each ORU, that failure rate estimate was generated via Bayesian updating of the initial estimate based on observed system behavior on-orbit. As a result, the current MADS failure rate estimates cannot be validated via comparison to ISS maintenance history, since that history was used to define the rates. To do so would be akin to using the same dataset for both training and testing of a machine learning algorithm; the results of that approach would not be a true validation. In addition, MADS does not track crew action rates, and therefore those must be generated for this research.

Therefore, the full dataset is split into two sections, each consisting of approximately four years, or half of the timeline examined here. The first is used for Bayesian updating, while the second is used for testing, and is further subdivided into fifteen simulated missions in order to validate POS calculations, discussed below. The overall timeline is displayed in Figure 5.13, which shows the R&R and non-R&R actions associated with each ORU, as well as the division of the timeline into data used for Bayesian updating and data used for testing.

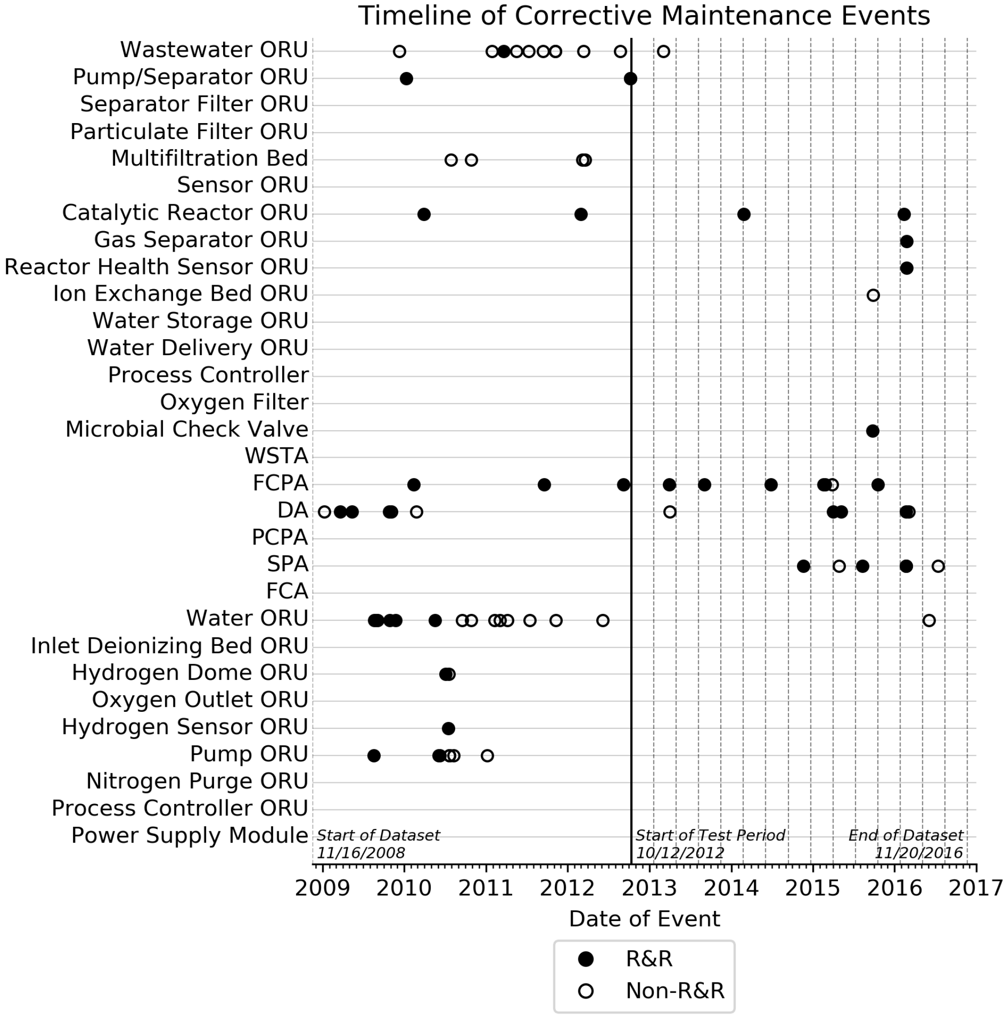

Figure 5.13: Timeline of corrective maintenance actions used for validation, showing R&R and non-R&R events for each ORU. A solid black vertical line marks the split between the data used for Bayesian updating (on the left), and the data used for testing (on the right). The test period is further subdivided into fifteen 100-day missions, indicated by dotted vertical lines.

The first portion of the timeline, consisting of the 1,427 days up to and including October 12, 2012, is used to perform a Bayesian update on failure rates and crew action rates based on the number of observed failures and crew actions during that time period. These updates are performed using the approach described by Dezfuli et al. [96, p. 38], with gamma-distributed failure rate distributions being used as the conjugate prior for the parameter of a Poisson distribution. This approach is similar to the Bayesian Reliability Average Weighted Likelihood Prior (BRAWL) model, which is one of two models used by ISS L&M to update failure rate estimates for MADS. However, unlike MADS updates, this validation do not distinguish between random and induced failures for the purposes of Bayesian updates, since both are contributors to the overall random processes described by Λ and Λ_C. The parameters α_{prior} and β_{prior} of the gamma-distributed effective failure rate prior distribution are derived by matching the expected value and variance of the lognormally-distributed failure rate described by the mean and error factor provided in MADS, multiplied by the associated K-factor. Given these prior parameters, an operating time t_o (which accounts for both duty cycle and the quantity of the item in operation), and an observed number of events n_o, the posterior gamma distribution representing effective failure rate has the parameters [96]

$$\alpha_{post} = \alpha_{prior} + n_o \tag{5.1}$$

$$\beta_{post} = \beta_{prior} + t_o \tag{5.2}$$

For failure rate updates, n_o is the number of R&R events; for crew action rate updates, n_o is the total number of events, including both R&R and non-R&R events. The posterior parameters are then used to define required inputs for MCROSS and EMAT, including failure / crew action rate expected value, variance, and error factor.

The second half of the data – characterizing 1,500 days of operations from October 13, 2012 to November 20, 2016 – is used for testing. First, the inputs themselves (failure rates, crew action rates, and maintenance crew times per action) are validated on an ORU-by-ORU basis by examining the p-value associated with each rate and observed number of failures – that is, the probability of a result equal to or more extreme than the observed value, given the model [144]. A low p-value indicates that the number of observed failures or crew actions during a test period are not in agreement with the hypothesized model, while a higher p-value indicates greater agreement between the model and the result. However, it is important to note that a p-value is not a measure of the probability that a hypothesis (in this case, a failure or crew action rate estimate) is true, and that conclusions should not be drawn from p-values alone – a point emphasized by the American Statistical Association (ASA) [144]. Therefore, this research uses p-values as one of many factors used to inform a discussion of the validation results, and does not apply a specific p-value threshold for strict hypothesis testing. The amount of crew time required for maintenance for each ORU is also compared to the CMMTTR estimate in MADS and discussed.

Model outputs are tested in two ways. First, MCROSS and EMAT are both used to assess the spares mass required as a function of POS_R for a range of values, and the results are compared. Then, to validate MCROSS spares allocations and their associated POS estimates, these allocations are compared to actual spares demands on the ISS. Spares allocations cannot be compared directly to actual spares demand during the test period, since the number of spares provided is a protection against a specified level of risk, not a prediction of the specific number of failures that will occur. Therefore, a binomial test is used to validate the POS associated with a particular spares allocation, as modeled by MCROSS. The test portion of the data is split into fifteen simulated missions of 100 days each. Given a spares allocation (in this case, the optimal allocation found by MCROSS for a particular POS), each of the fifteen simulated missions is designated a "success" or "failure" based on whether the provided allocation would have been sufficient to cover all corrective maintenance demands during that mission. The p-value associated with the modeled POS level and observed number of successes is then calculated and discussed, using a binomial model where the probability of success for each mission is equal to the POS.

5.2.2 Input Parameter Validation

System-level supportability analysis combines ORU-levels supportability characteristics such as failure rates, crew action rates, mass and maintenance crew times in order to determine overall risk, mass, and crew time. This section validates the input parameters associated with each ORU against ISS flight data in order to understand the validity of the data being input to the model and provide context for the validity of model outputs, which are examined in Sections 5.2.3 and 5.2.4, examining failure rates, crew action rates, and maintenance crew time. Mass estimates are not validated here because that is a system parameter that can be measured directly and there is little uncertainty associated with it.

Figure 5.14 shows the results of two-tailed hypothesis testing on the failure and crew action rates for each ORU, using results from the entire testing period. The top portion shows results for failure rates, while the bottom portion shows the results for crew action rates. In each case, the *p*-value – that is, the probability of observing a number of failures or crew actions of an equal or lesser likelihood as the one actually observed, based on the negative binomial distribution defined by the parameters generated via the Bayesian update described above – is shown on the vertical scale for each ORU. Lower *p*-values indicate that the observed number of failures are statistically different from what would be expected if the failure rate or crew action rate estimate were accurate.

In order to indicate the directionality of the disagreement between the model and the observed number of events, the vertical strip associated with each ORU is shaded red or blue to indicate whether the rate was underestimated or overestimated, respectively. In this context, a rate is considered a possible underestimate if the number of observed events is greater than the expected value and possible overestimate if the observed number is less than the expected value. The intensity of the shading is inversely proportional to the *p*-value so that darker strips indicate cases of greater statistical difference between observed and modeled numbers of events.

In terms of failure rates, these results indicate that for most ORUs the observed

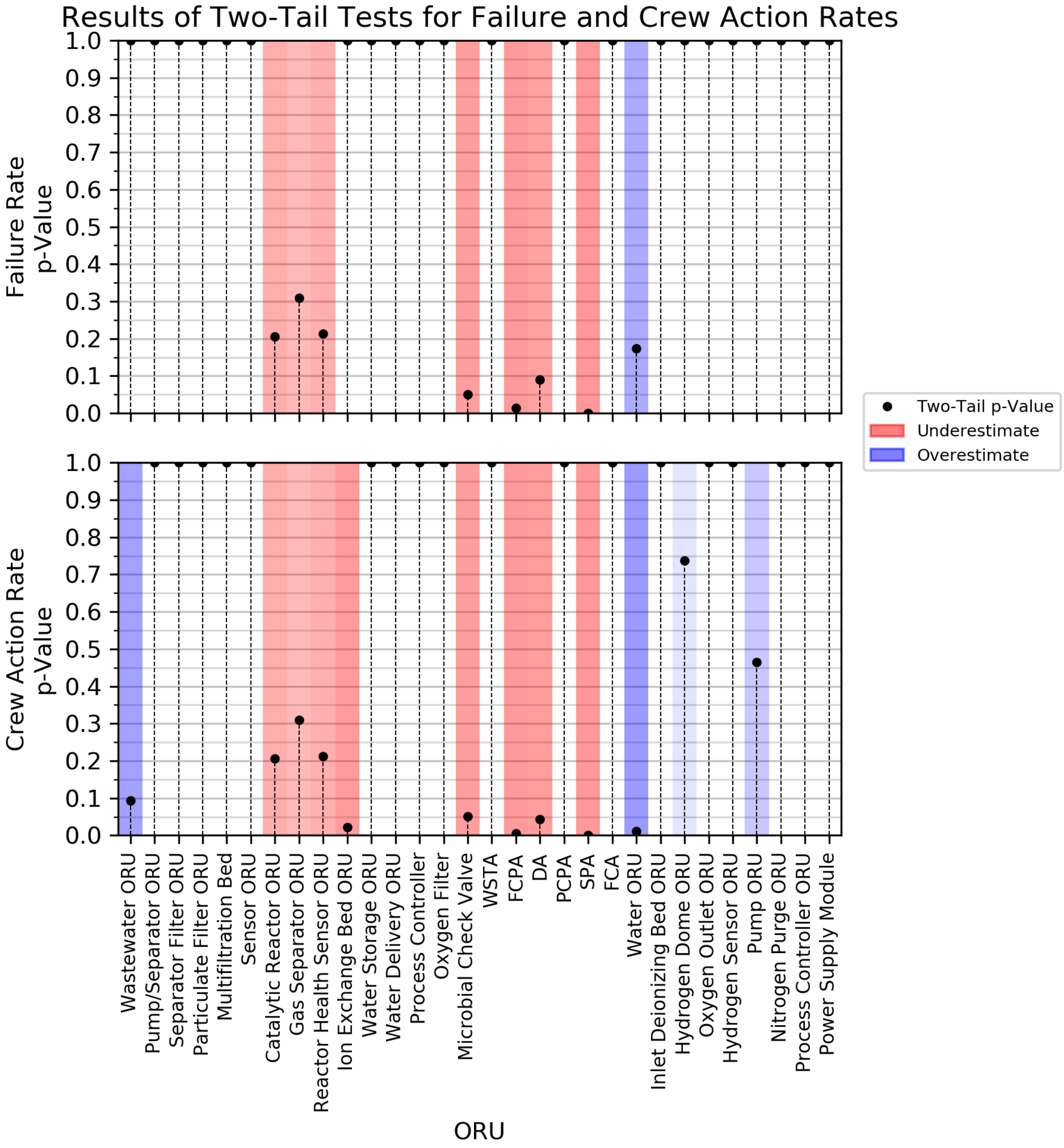

Figure 5.14: Results of two-tailed hypothesis testing of ORU failure rates (top) and crew action rates (bottom). Black dots indicate the p-value for each ORU. The vertical strip for each ORU is shaded red if the predicted rate was an underestimate (i.e. the number of observed events was greater than expected), and blue if the rate was an underestimate. The intensity of the shading corresponds to the p-value, with more significant disagreement between prediction and observation indicated with darker shading.

number of failures is statistically in agreement with the estimated failure rate. For 22 of the ORUs, the p-value was equal to 1, indicating that the observed result was the most likely outcome – a very good match between the model and the observation. Four ORUs – the Microbial Check Valve, Fluids Control and Pump Assembly (FCPA), Distillation Assembly (DA), and Separator Plumbing Assembly (SPA) – have p-values below 0.1, and two of these (the FCPA and SPA) have p-values below 0.05. In all of these cases, the observed number of failures was greater than the expected number, meaning that the failure rate was underestimated.

Results are similar for crew action rates; for 18 of the ORUs examined here, the observed outcome was the most likely outcome, according to the model. Seven ORUs have a crew action rate p-value less than 0.1: the Wastewater ORU, Ion Exchange Bed ORU, Microbial Check Valve, FCPA, DA, SPA, and Water ORU. Of these, all have a p-value below 0.5 except the Wastewater ORU and Microbial Check Valve. In addition, the number of observed crew actions was higher than expected for all of these ORUs except the Wastewater ORU and Water ORU.

A detailed assessment of the causes behind the underestimated failure rates and crew action rates described here requires much more in-depth examination of the design and operating conditions of each of these ORUs and the specific root causes of the failures and maintenance activities documented in MDC, and is beyond the scope of this dissertation. However, a preliminary examination indicates that some of the ORUs with the most significantly underestimated failure rates are complex assemblies within the WPA and UPA that involve rotating equipment and/or handle fluid flow in microgravity. The FCPA is a peristaltic pump used to transfer urine into the DA, where a rotating centrifuge is used to evaporate water, separating it from brine. Another ORU with an underestimated failure rate, the SPA, handles two-phase flow, separating gas from liquid. The Microbial Check Valve combines a mechanical check valve with an ion exchange resin, which releases iodine in order to prevent contamination during water transfer. While this dissertation does not draw any specific conclusions regarding the reasons for these underestimated failure rates, these data suggest that rotating equipment, multi-phase flow, and valves are important areas for system reliability. Therefore, investment in robust design and testing for these types of systems may be particularly valuable from a supportability perspective for future beyond-LEO spacecraft.

Overall, these results indicate that, for most ORUs examined here, the observed behavior of the system during the test period is in agreement with the failure and crew action rate estimates generated via Bayesian updating. This is not a confirmation that these rate estimates are necessarily accurate or precise; it is simply an indication that there is not a statistically significant difference between expected and observed behavior for those ORUs. However, for some ORUs, the number of failures and crew actions that occurred during the test period was higher than would be expected if the rate estimates were accurate, indicating that those rate estimates were statistically significant underestimates. As a result, POS assessments associated with those are likely to be overestimates and the number of spares provided during allocation optimization is likely to be insufficient to achieve POS_R. In addition, the estimated amount of crew time associated with those ORUs may not be high enough. For a very small number of ORUs, the number of observed events was lower than would be expected, indicating that those rates may be overestimates. In two cases (the Wastewater ORU and Water ORU), the crew action rate was a statistically significant overestimate. Therefore, the amount of crew time associated with these ORUs may be overestimated.

Figure 5.15 compares the CMMTTR estimate in MADS to the recorded maintenance crew time for events during the Bayesian update period at the first half of the timeline examined here. The crew time of each maintenance event, along with the average across all events, is plotted for each ORU that experienced a maintenance action during that time period, with each maintenance crew time value normalized by the CMMTTR recorded in MADS. Since CMMTTR is a deterministic estimate, rather than a distribution, this comparison does not perform any hypothesis testing, as was done for failure and crew action rate estimates. However, these results indicate that, for all ORUs that experienced maintenance events in the Bayesian update period, the average amount of time required per maintenance event exceeded

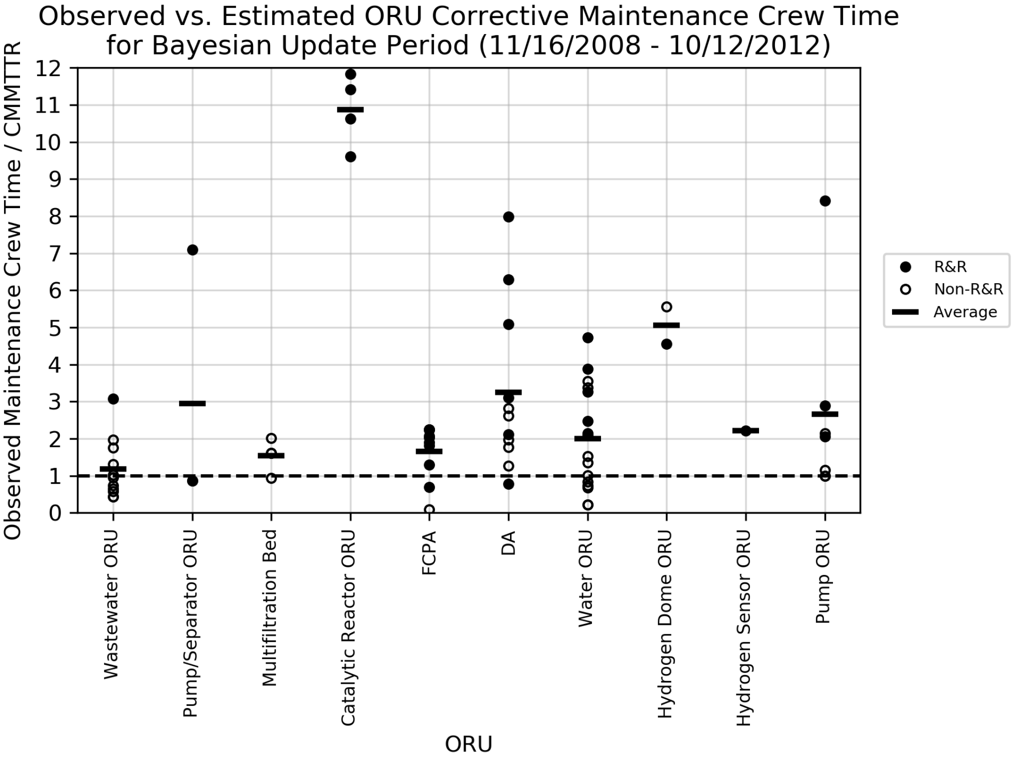

Figure 5.15: Comparison of observed and estimated ORU maintenance crew time. The crew time associated with each event is normalized by the CMMTTR estimate in MADS for comparison.

CMMTTR. In some cases, such as the Wastewater ORU, the discrepancy is small. However, in some cases the observed average is significantly higher – up to nearly eleven times higher than CMMTTR, in the case of the Catalytic Reactor ORU. One possible source of this discrepancy may be different definitions of "maintenance crew time" for different sources. For example, it is possible that MDC records include all crew time associated with a particular maintenance activity – including planning and discussion with ground support teams, finding tools and spares, setting up, and cleaning up afterwards – while the CMMTTR refers only to the act of removing and replacing the failed ORU. As a result, crew time analyses using CMMTTR will likely underestimate the amount of maintenance crew time recorded in MDC.

As with all models, the validity of MCROSS outputs depends upon the validity of the associated inputs, and analysts examining the supportability of any system should carefully examine the quality of their input data. One key element of this is the definition of events and terms, such as the determination of which types of maintenance events should be categorized as corrective or preventative, and what should be included in maintenance crew time estimates. Future validation efforts (discussed in Chapter 9) will investigate the source data for these analyses in greater depth.

5.2.3 Crew Time Results Validation

To assess the validity of crew time results, the expected maintenance crew time assessed by MCROSS and EMAT/ECTM were compared to actual maintenance crew time demands on board the ISS during the fifteen simulated missions. Due to the discrepancy between the MADS CMMTTR estimate and the observed average maintenance crew time during the Bayesian update period identified in Figure 5.15 and discussed above this analysis was performed twice: once using the estimated CMMTTR, and once using the observed average.

The version of EMAT available for this validation exercise did not include crew time assessment via ECTM, and therefore the results shown here are based on a recreation of the methodology described by Mattfeld et al. [7] and Stromgren et al.

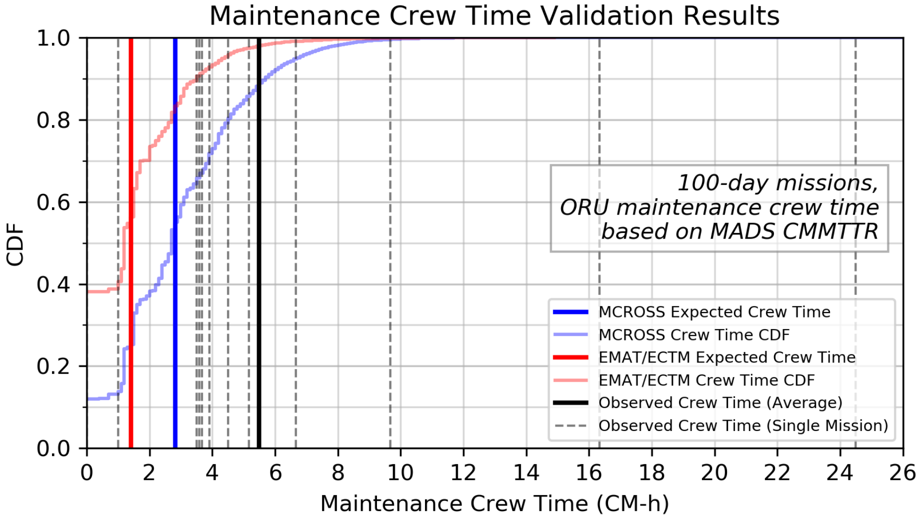

Figure 5.16: Results of maintenance crew time validation, comparing MCROSS (blue) and EMAT/ECTM (red) results to observed maintenance crew time demands on ISS (black) for the fifteen 100-day simulated missions examined here when the MADS CMMTTR estimate is used for each ORU. The observed maintenance crew time for each mission is indicated by vertical dotted lines.

[100]. A binomial distribution defined by the number of days in the mission and the daily probability of failure is used for Monte Carlo simulation of the number of failures experienced by each ORU, which is then multiplied by the associated maintenance crew time amount to obtain the total maintenance crew time associated with that ORU. Ten thousand Monte Carlo simulations were executed, with the daily probability of failure for each ORU defined by drawing a failure rate from the empirical uncertainty distribution associated with that ORU's error factor (i.e. a scaled version of the observed distribution from historical data; see Section 2.1.5 for more detail). Both the expected value and the distribution of the amount of crew time required were calculated for both models, with the crew time step size Δt was set to 0.1.

Figure 5.16 shows the results of crew time analysis in the case where the MADS CMMTTR estimate is used for each ORU. The amount of crew time actually used on-orbit for these ORUs during each the fifteen 100-day simulated missions examined here, indicated by dotted black vertical lines, ranged from 0 CM-h (which occurred for four missions) to 24.5 CM-h The majority of missions experienced maintenance crew

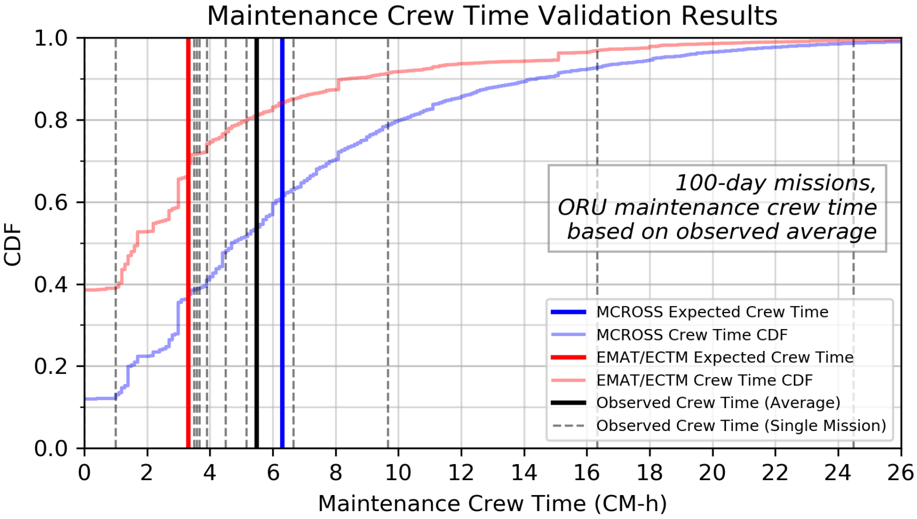

Figure 5.17: Results of maintenance crew time validation, comparing MCROSS (blue) and EMAT/ECTM (red) results to observed maintenance crew time demands on ISS (black) for the fifteen 100-day simulated missions examined here when the average maintenance crew time observed during the Bayesian update period is used for each ORU. The observed maintenance crew time for each mission is indicated by vertical dotted lines.

time demands of approximately 3.5 to 7 CM-h, and the average observed maintenance crew time was 5.5 CM-h, indicated by the solid black vertical line. In this case, MCROSS and EMAT/ECTM estimated an expected maintenance crew time of 2.8 CM-h and 1.4 CM-h, respectively. This significant underestimate is expected, due to the fact that the observed maintenance crew time associated with each ORU was higher than the CMMTTR in all cases, sometimes significantly so (see Figure 5.15).

Figure 5.17 addresses this by repeating the analysis with CMMTTR replaced by the observed average maintenance crew time for each ORU – that is, the average maintenance crew time per event based on events observed during the Bayesian update period. When these updated maintenance crew time estimates are used, MCROSS calculates expected crew time to be 6.3 CM-h, and EMAT/ICOM calculates expected crew time to be 3.3 CM-h. Put another way, MCROSS overestimated observed average crew time requirements (5.5 CM-h) by 14.5%, while EMAT/ECTM underestimated by 40%. This underestimate is expected, and is a result of the fact that EMAT/ECTM, like all other maintenance crew time models found in the literature,

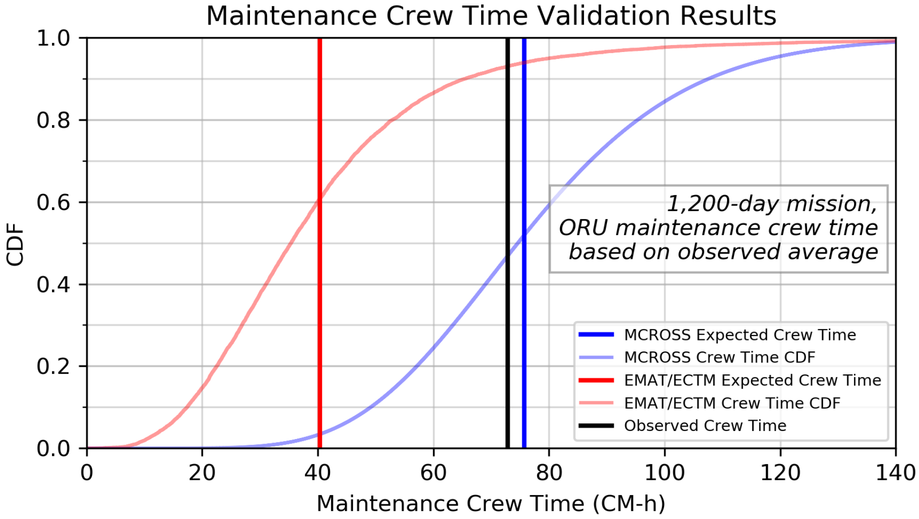

Figure 5.18: Results of crew time validation for a single 1,200-day mission, comparing MCROSS (blue) and EMAT/ECTM results to observed ISS maintenance crew time demands for the last 1,200 days of the test period, indicated by the vertical black line.

uses ORU failure rates to drive crew time demand, and therefore does not consider the impact of non-R&R events. MCROSS, on the other hand, uses a separate crew action rate to capture the impact of these non-R&R events and therefore provides a more complete assessment of maintenance crew time requirements.

In order to examine the validity of the model for application to longer missions, MCROSS and EMAT/ECTM were used to examine the same system with a mission endurance of 1,200 days – the required endurance for a Mars transit mission [11]. Average observed maintenance crew time requirements were used for each ORU. Model outputs were then compared to the amount of maintenance crew time used on ISS for the last 1,200 days of the test period. The results of this analysis are shown in Figure 5.18. The total maintenance crew time observed on the ISS for the last 1,200 days examined here (i.e. the last 12 simulated missions) was 72.8 CM-h. MCROSS calculated an expected maintenance crew time of 75.7 CM-h, an overestimate of 3.9%, and EMAT/ECTM calculated an expected maintenance crew time of 40.3 CM-h, an underestimate of 44.6%. These results reinforce the importance of including non-R&R events in maintenance crew time assessments.

It is important to note that the EMAT/ECTM crew time underestimates are not due to any error or miscalculation in the EMAT model. Instead, they are the result of the scope of maintenance crew time that is captured in EMAT/ECTM – and, based on the results of the literature review presented in Chapter 2, every other maintenance crew time model. Specifically, these models only include crew time associated with R&R events, since crew time evaluation is driven by the same failure rates that are used for spares calculations. While this approach accurately models the amount of crew time spent on R&R of failed ORUs, it does not capture crew time associated with non-R&R events, and therefore underestimates total maintenance crew time requirements. There is nothing inherent to the EMAT/ICOM modeling methodology (or any other crew time assessment methodology) which prevents them from using crew action rates instead of failure rates for crew time evaluation. The contribution of this research is the separation of spares and crew time demands into two different stochastic processes in order to expand the scope of crew time analyses to include this non-R&R time and improve accuracy.

5.2.4 Spares Allocation and Mass Results Validation

This section validates spares allocation and mass outputs of MCROSS in two ways. First, the spares mass required as a function of POS_R is evaluated across a range of POS values using both MCROSS and EMAT, and the results are compared. Mission endurances of 100 days and 1,200 days are both examined. Then, the spares allocation and associated POS assessment output by MCROSS is compared to actual observed ISS maintenance demands during each of the fifteen simulated 100-day missions created by splitting the test period. A two-tailed binomial test is used to calculate the p-value associated with the observed number of successful missions – that is, missions in which the MCROSS spares allocation would be sufficient to cover all demands during the mission – based on the hypothesis that the POS calculated by MCROSS for that allocation is correct.

Comparison to EMAT

Figure 5.19 shows the spares mass requirements for a 100-day mission with the ORUs listed in Table 5.2, as calculated by MCROSS and EMAT. Figure 5.20 shows the same results for a 1,200-day mission. Results are shown for all required POS values below 0.9999. To examine epistemic uncertainty, EMAT assesses POS and confidence (see Section 2.1.5 for more details); therefore, in order to enable a direct comparison, MCROSS was modified to include an assessment of the POS-confidence distribution surrounding each spares allocation. This modification, unlike the rest of MCROSS, required Monte Carlo simulation. 1,000 Monte Carlo simulations were used for the EMAT evaluation, since this is the maximum amount currently available in the tool; 50,000 Monte Carlo simulations were used to evaluate confidence levels for MCROSS outputs. The median POS as well as the interquartile range (i.e. the interval between the 25th and 75th percentiles of the confidence distribution) are shown for both models, along with the standard MCROSS POS output, which is an evaluation of POS that accounts for epistemic uncertainty. EMAT outputs are shown in red, while MCROSS outputs are shown in blue; the dotted blue line indicates the MCROSS POS assessment, while the solid red and blue lines and surrounding shaded regions indicate the median and interquartile for EMAT and MCROSS, respectively. Note that POS values are shown on a logarithmic scale, and confidence levels refer to POS, not mass.

Figures 5.19 and 5.20 show that MCROSS and EMAT median POS assessments are in close agreement and follow similar trends, though the interquartile range for EMAT is wider than the one calculated by MCROSS. For the shorter, 100-day mission (Figure 5.19), the standard MCROSS POS output also trends closely with the median values for both models. For the 1,200-day case (Figure 5.20), the MCROSS standard output corresponds to a higher mass (and/or lower POS value) than both median values. This divergence is expected, since the underlying POS distribution is highly skewed. Longer missions magnify the effect of failure rate uncertainty, resulting in a wider POS distribution. Since the MCROSS POS calculation is an evaluation of total

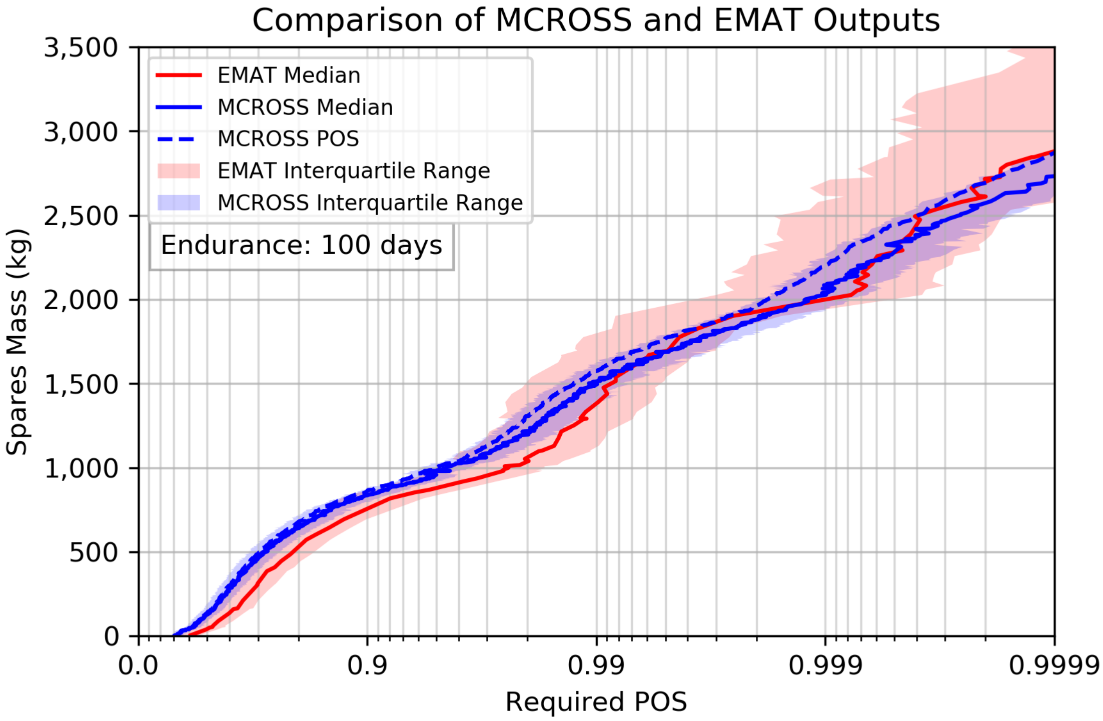

Figure 5.19: Comparison of MCROSS (blue) and EMAT (red) spares mass assessments for a 100 day mission. The median and interquartile range of the POS distribution is shown by the solid line and shaded region for each model, and the POS calculated by MCROSS is shown by the dotted blue line.

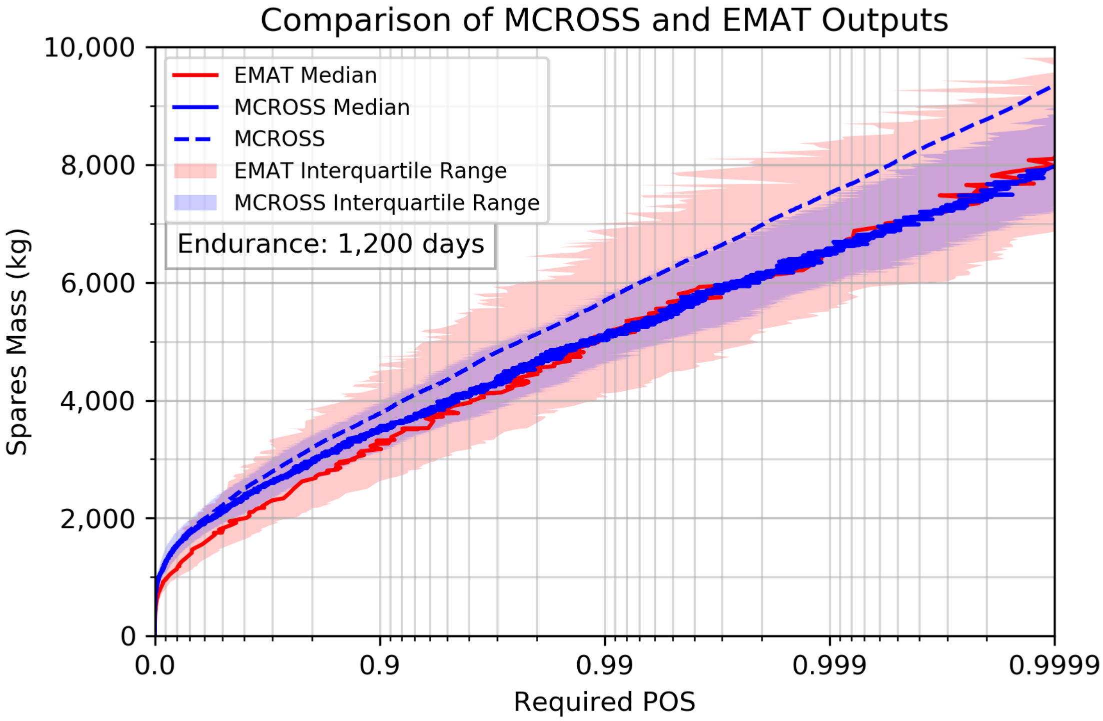

Figure 5.20: Comparison of MCROSS (blue) and EMAT (red) spares mass assessments for a 1,200 day mission. The median and interquartile range of the POS distribution is shown by the solid line and shaded region for each model, and the POS calculated by MCROSS is shown by the dotted blue line.

risk, including both epistemic and aleatory uncertainty, it is expected that it would be higher than the median of the POS distribution, especially when significant amounts of epistemic uncertainty are present. Put another way, a higher level of confidence would be required to achieve the same total level of risk for a given POS level.

Comparison to Actual ISS Maintenance Demands

Validation of MCROSS spares allocations is executed in a similar manner to that used to investigate ORU failure rates in Section 5.2.2. First, MCROSS is used to identify the optimal spares allocation for a 100-day mission a specified POS requirement POS_R, as well as the actual POS achieved by that allocation. (Since POS is a function of the discrete number of spares provided, it is unlikely to match POS_R exactly; in practice, the calculated POS for the optimal allocation is slightly greater than POS_R.) Then, this allocation is evaluated against actual ISS spares demands in each of the fifteen simulated missions in the test period described above in order to determine whether or not that allocation would have been sufficient. Missions for which the allocation would have been sufficient are considered a success, and the number of successful missions is used to calculate a two-tailed p value based on the binomial distribution defined by the number of test missions and the calculated POS. That is, the p-value is the sum of the probability mass associated with all outcomes of equal or lesser likelihood than the observed outcome, based on the hypothesized distribution of outcomes defined by the MCROSS-calculated POS. Since these analyses all focus on POS associated with spares, there was no limit set on the amount of crew time available for maintenance in the MCROSS used to generate spares allocations; the probability of sufficient crew time is therefore effectively set to 1.

The validity of the POS value calculated by MCROSS is strongly dependent upon the validity of the failure rate estimates input into the model. Investigation of input parameters (Section 5.2.2) indicated that failure rates were underestimated to a statistically significant level for a small number of ORUs examined here. Put another way, those failure rate estimates failed a validation test against observed behavior. In order to investigate the impacts of invalid inputs on MCROSS, the spares allocation

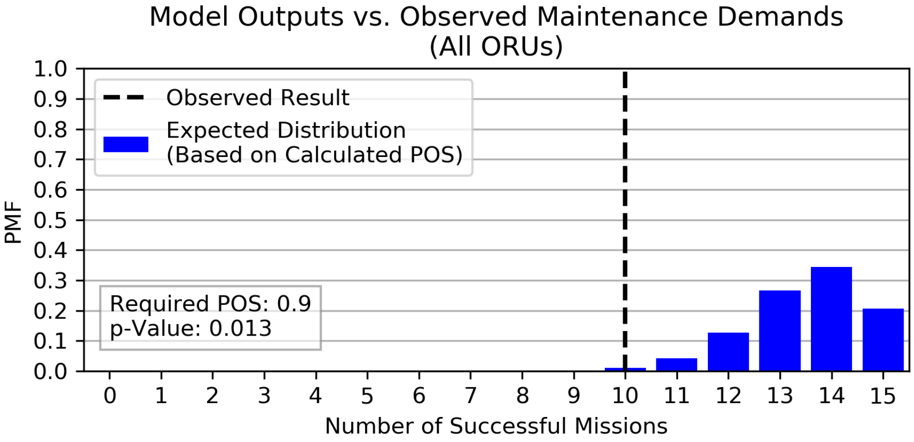

Figure 5.21: Comparison of MCROSS outputs to observed cases in which the spares allocation would have been sufficient, based on actual ISS maintenance demands. This assessment includes all ORUs in Table 5.2, even those whose failure rates were found to be underestimated in Section 5.2.2. Spares were allocated to meet a POS requirement of 0.9. The expected distribution of the number of successes is shown in blue, and the observed number of successes is indicated by the vertical line.

testing described in this section is performed twice: once using the full set of ORUs (including those with significantly underestimated failure rates), and once with only the subset of ORU for which failure rate estimates were successfully validated against observed behavior. That is, in the second analysis, only ORUs for which the failure rate test indicated a p-value greater than 0.05 were included.

The first test, using all ORUs, is executed for a required POS of 0.9. In this case, the spares allocation output by MCROSS is sufficient in ten of the fifteen simulated missions. Figure 5.21 shows the binomial distribution resulting from the calculated POS of the allocation output by MCROSS (approximately 0.90075) compared to the observed number of successes. The expected number of successes, based on the calculated POS, is 13.5. The p-value in this case is 0.013, indicating that the observed number of successes is significantly different from what would be expected if the calculated POS were correct. Specifically, the model has significantly overestimated POS; the calculated POS was just over 0.9, but only two thirds of the missions were successful.

This result is not unexpected, given that the examination of input parameters

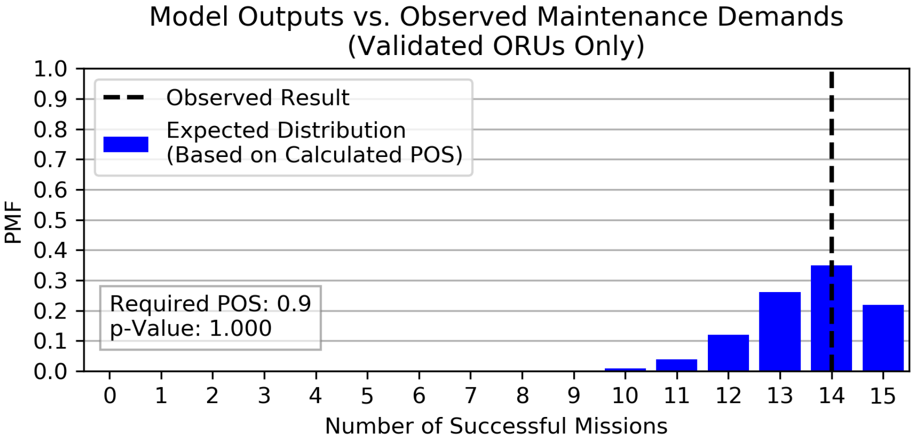

Figure 5.22: Comparison of MCROSS outputs to observed cases in which the spares allocation would have been sufficient, based on actual ISS maintenance demands. ORUs identified as having underestimated failure rates in Section 5.2.2 were not included, in order to use only validated model inputs. Spares were allocated to meet a POS requirement of 0.9. The expected distribution of the number of successes is shown in blue, and the observed number of successes is indicated by the vertical line.

presented in Section 5.2.2 (see Figure 5.14) indicated that the failure rates associated with two of the ORUs – FCPA and SPA – were significantly underestimated (i.e. $p < 0.05$). As a result, the POS associated with these ORUs was likely overestimated, and the overall POS was likely overestimated. Examination of the specific ORU(s) causing insufficiency in each of the five failed missions revealed that a lack of sufficient spares for either the FCPA and SPA were responsible for four of them. Therefore, this research concludes that the difference between modeled and observed results in this case is primarily driven by invalid inputs (i.e. underestimated failure rates).

In order to investigate the validity of the model when given valid inputs, the analysis was repeated using a subset of the initial ORU list. Specifically, the FCPA and SPA were removed in order to leave only ORUs for which failure rate testing (Section 5.2.2) had resulted in $p > 0.05$. The results are shown in Figure 5.22. In this case, the observed number of failures is fourteen, which is also the most likely outcome under the calculated POS. With a p-value of 1, these results indicate that model outputs are in agreement with observed results when inputs are a valid valid representation of the system.

In order to investigate a range of POS values, the analysis was repeated for required POS values of 0.8 and 0.99. The results from these analysis are presented in Figure 5.23, along with the previous results for a POS of 0.9 for comparison. When spares allocations are selected to meet a POS requirement of 0.8, the allocation is still sufficient for fourteen of the fifteen missions (the expected value is 12). The *p*-value in this case is 0.331, which is lower, but still within the range of probable outcomes. At the higher POS requirement of 0.99, the MCROSS spares allocation is sufficient in all missions, matching the most likely outcome. These results indicate that MCROSS spares allocations and associated POS assessments are not statistically different from observed results in this range.

Overall, these results indicate that, given valid inputs, the spares allocation and associated POS value output by MCROSS are in agreement with observed ISS maintenance demands for the test period examined here. However, they also highlight the importance of using valid input data for these analyses. Underestimated failure rates for just two out of the 30 ORU examines here had a significant effect on the validity of model outputs, with MCROSS significantly overestimating POS when those overestimated rates were included. In application, this overestimated POS could result in a significantly underestimated spares mass requirement and/or underestimated risk, for a particular mission. It is therefore critical that the input parameters for MCROSS (or any other supportability model) be carefully examined, and that analysts bear in mind the potential impacts that under- or overestimated input parameters could have on the results of the analysis.

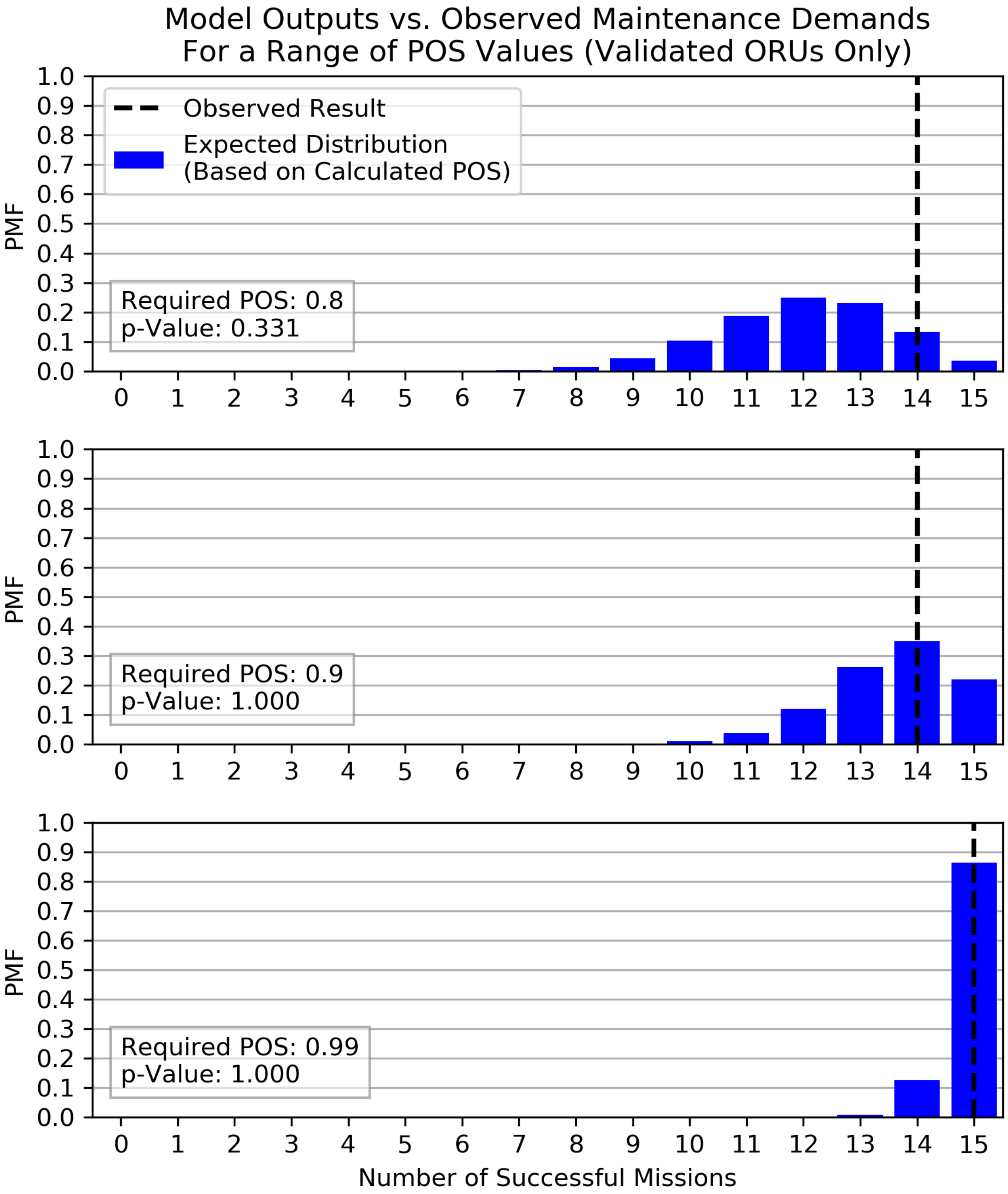

Figure 5.23: Comparison of MCROSS outputs to observed cases in which the spares allocation would have been sufficient, based on actual ISS maintenance demands, for a required POS of 0.8 (top), 0.9 (middle), and 0.99 (bottom). ORUs identified as having underestimated failure rates in Section 5.2.2 were not included, in order to use only validated model inputs. The expected distribution of the number of successes is shown in blue, and the observed number of successes is indicated by the vertical line.

5.3 Summary

This chapter has presented a verification and validation of the supportability modeling framework described in Chapters 3 and 4, called MCROSS. Model outputs and intermediate results were compared to the results of Monte Carlo simulation in Section 5.1 in order to verify model performance and investigate the error introduced by some modeling assumptions. Specifically, comparisons were made between Monte Carlo and model results with regard to

- the use of the gamma distribution to approximate the lognormal failure rate distribution, and the resulting impacts on POS calculations;
- the expected value and variance based method for determining ORU characteristics as a function of the characteristics of the components within it, including the approximation used for component redundancy; and
- the numerical characteristic function inversion approach used to model crew time and ODM feedstock demands.

This verification process showed that model outputs closely matched the results of Monte Carlos simulation for the majority of the examined cases.

The largest source of error was the redundancy approximation (Figure 5.9 on page 230), which this analysis showed to have potentially large differences in POS values – a magnitude of 0.15 to 0.25 – when evaluating POS significantly less than 0.9. These results indicated that the MCROSS model can significantly overestimate POS (and therefore will likely underestimate spares mass requirements) when redundancy is present and the POS of interest is less than 0.9. However, the amount of error present rapidly reduces as higher POS values are investigated, dropping below 0.03 for all POS above 0.9 examined here. Since most supportability analyses are interested the higher end of the POS range (i.e. values greater than 0.9), these verification results indicate that, while care must be taken if lower POS values are to be investigated, this model provides an accurate assessment of POS in the region of interest.

In addition, examination of results from the numerical characteristic function inversion crew time and ODM model highlighted the importance of the step size parameters Δt and Δm. Smaller step sizes did not necessarily result in lower levels of error; instead, this verification indicated that the most accurate results are obtained when most or all of the crew time and mass values associated with each ORU are multiples of the step size used (see Figure 5.11 on page 232). The largest error observed in the case examined here showed the model underestimating POS by 0.14, though as with other sources of error the amount of error present reduced significantly as higher POS values were investigated.

Once model verification was complete, Section 5.2 validated MCROSS crew time and spares outputs by comparing them to results from EMAT as well as to actual spares demands on the ISS. On-orbit corrective maintenance data for three ISS ECLSS subsystems (WPA, UPA, and OGA) were compiled for the eight-year time period between November 16, 2008 and November 20, 2016. The first half of the dataset was used to define system parameters via Bayesian updating. The second half was split into fifteen 100-day simulated missions for testing purposes.

Before supportability analysis, key input parameters – failure rates, crew action rates, and maintenance crew times – were validated against the ISS data on an ORU-by-ORU basis in Section 5.2.2. This input parameter validation indicated that the number of failures observed during the test period was in agreement with failure and crew action rates for a majority of the ORUs examined here, though in some cases the failure and crew action rates were significantly underestimated, while in even fewer cases the crew action rates were significantly overestimated. However,estimated maintenance crew time requirements per action, captured as CMMTTR, were typically underestimated, sometimes significantly.

Section 5.2.3 validated the MCROSS crew time model via comparison to both EMAT and ISS crew time demands. An initial analysis was carried out using CMMTTR (Figure 5.16 on page 251), and showed that, as expected, both models significantly underestimated crew time requirements when the amount of crew time associated with each particular maintenance action was underestimated. When the maintenance crew time estimates associated with each ORU were updated to the average values observed during the first half of the dataset (i.e. the portion used for Bayesian updating), corrective maintenance crew time estimates were much more accurate. The MCROSS expected corrective maintenance crew time estimate was within 0.8 CM-h (14.5%) of the observed average crew time for the 100-day test missions. When the analysis was repeated for a 1,200-day mission and compared to actual maintenance crew time demands during the last 1,200 days of the ISS test period, the MCROSS expected corrective maintenance crew time estimate was within 2.9 CM-h (3.9%) of the observed value.

The first portion of Section 5.2.4 compared MCROSS and EMAT mass outputs for a range of POS values, examining 100- and 1,200-day missions. Since EMAT evaluates risk in terms of POS and confidence, MCROSS was modified to include a Monte Carlo confidence assessment, and the median and interquartile range of the output POS was compared for each model. These results showed close agreement between the two models, though the interquartile range for EMAT was wider than that of MCROSS. In addition, these results showed that the total POS calculated by MCROSS – which includes both epistemic and aleatory uncertainty – diverges from the median calculated by either EMAT or MCROSS for longer missions, showing a lower POS for the same mass, or a higher mass required to achieve the same POS, particularly for high POS values. This result is expected, since the POS confidence distribution will be wider when there is greater uncertainty, and longer missions magnify the impact of failure rate uncertainty. When there is more uncertainty, a higher degree of confidence is required to achieve the same level of overall risk for a given POS value.

Finally, MCROSS spares allocations for a variety of POS values were compared with observed ISS spares demand in order to validate MCROSS POS estimates. An initial analysis using all ORUs – including the ones identified in Section 5.2.2 as having underestimated failure rates – showed that, as expected, the spares allocation output by MCROSS was not sufficient to the desired level, based on observed spares demand (Figure 5.21). In order to remove the effect of invalid inputs from the spares model validation, the two ORUs with significantly underestimated failure rates were removed and the analysis was repeated. This analysis, along with follow-on analyses examining different POS levels (Figure 5.23 on page 262), showed that the observed number of simulated missions (derived from ISS spares demand) for which the MCROSS spares allocation was sufficient was in agreement with the POS estimate calculated by MCROSS.

Some elements of the model were not included in this validation, and are left to future work. Specifically, the ODM model is not included due to lack of data for validation; ISM has not yet been incorporated into any space mission supportability plan. Preventative maintenance was also not included, in order to maintain the scope of this analysis at a manageable level and provide the focus required to assess corrective maintenance in depth. Preventative maintenance is considered the least complex element of this model, since it is based on deterministic system characteristics that are themselves typically determined by operational decisions – that is, the desired maintenance frequency for life-limited ORUs. Future work, described in Chapter 9, will expand upon this validation to examine all elements of supportability.

Overall, the verification and validation presented in this chapter demonstrates that the MCROSS methodology is sound, with the exception of modeling of redundancy at low POS levels. In addition, comparison of model outputs to real-world maintenance demands on the ISS indicate that, when model inputs are valid representations of the system, MCROSS outputs are in agreement with observed system behavior. However, this validation has also shown the importance of valid inputs. Analysis of both crew time and spares allocations indicated that model outputs can diverge significantly from observed system behavior when input parameters are not good representations of the system. A model is only as good as its inputs, and it is critical that supportability analysts using MCROSS (or any other model) carefully consider the validity of the failure rates, maintenance crew time estimates, and other parameters used in the analysis.

Chapter 6

Case Study: Notional System

This chapter presents a demonstration of MCROSS using a notional system, defined below. MCROSS is used to assess mass and maintenance crew time requirements for a range of POS levels and mission endurances, and a sensitivity analysis on ODM system mass is used to identify levels at which ODM-based architectures are or are not Pareto-optimal. These results are discussed as an example of how the supportability modeling capabilities provided by MCROSS can be used to inform system and mission design.

The notional system examined here is intended to be representative of spacecraft systems, but it is not based on any existing system and is meant only for demonstration purposes. Its characteristics were selected to provide for a wide range of supportability strategies while keeping the size of the tradespace small enough to enable enumeration of all feasible solutions in a reasonable amount of time. The mass and crew time associated with this notional system for a baseline mission is compared to the results from a series of sensitivity analyses investigating the impact of changes in key parameters. In addition, the Pareto frontier identified by filtering the enumerated tradespace is compared to the one identified via optimization in order to validate the optimization process. The number of evaluations of both partial and full assignments to the supportability strategy decision variables required to obtain the Pareto-optimal set is compared to the number of evaluations required for full enumeration to examine the efficiency of the optimization algorithm and the bounding functions defined in Section 4.3.

The remainder of this chapter is organized as follows. Section 6.1 describes the notional system examined here, providing a system description as well as set of modeling parameters. Section 6.2 examines the supportability characteristics of this system for a notional baseline case using full enumeration. Section 6.3 compares the Pareto frontier obtained via full enumeration and filtering to one obtained via optimization in order to validate the optimization process and evaluate its efficiency. Section 6.4 performs a series of sensitivity analyses with respect to key parameters. Finally, Section 6.5 summarizes and discusses the results of this case study.

6.1 Notional System Description

The notional system examined in this case study consists of thirteen items arranged in the three-level indenture structure shown in Figure 6.1. The item list is shown in Table 6.2. Items 5 through 13 are components, and item 2 is a subsystem that could be distributed across multiple modules. Items 8, 9, 10, 12, and 13 are manufacturable, meaning that ODM could be used to cover spares demands. Items 6 and 8 are life-limited, and require replacement every 7,200 h (300 d) and 3,600 h (150 d), respectively. The only opportunities for redundancy are items 8, 9, and 13, each of which could use up to 1 redundant instance. Items 10, 11, and 12 are all potentially common with each other; therefore, the potential commonality matrix has entries $\hat{C}_{ii} = 1$ along the diagonal and $\hat{C}_{ij} = 1$ for all pairs of i and j drawn from $\{10, 11, 12\}$.

The modeling parameters for this case study are shown in Table 6.1. Redundancy and ODM both incur a 5% mass overhead. The redundancy effectiveness factor is 0.98, while parts that are manufactured on demand have a 10% overhead applied to failure and crew action rates. The baseline ODM system mass is 20 kg. The impacts of varying m_{odm} are explored as part of the sensitivity analysis in Section 6.4. Crew time and ODM feedstock mass are examined in step sizes of 0.25 CM-h (that is, 15 minutes of crew time) and 0.1 kg, respectively, and the maximum inversion error is

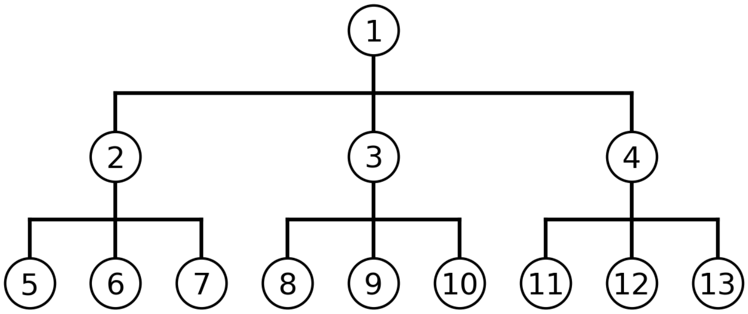

Figure 6.1: Indenture structure of the notional system used in this case study.

Table 6.1: Case study modeling parameters.

Parameter	Symbol	Value	Units
Redundancy Mass Overhead Factor	γ_r	0.05	
Redundancy Effectiveness Factor	$\gamma_{\Lambda r}$	0.98	
ODM Mass Overhead Factor	γ_{odm}	0.05	
ODM Failure / Action Rate Factor	$\gamma_{\Lambda odm}$	0.1	
ODM System Mass	m_{odm}	20	kg
Crew Time Discretization	Δt	0.25	CM-h
ODM Feedstock Discretization	Δm	0.1	kg
Maximum Inversion Error	ϵ_{max}	1×10^{-8}	

set to 1×10^{-8}.

6.2 Baseline Case

This section presents an analysis of a baseline case for this system, which consists of a 500-day mission with a POS requirement of 0.995, corresponding to a 1 in 200 chance of insufficient maintenance resources. The maximum amount of crew time available for maintenance is 600 CM-h, or 5% of total mission endurance (25 days). This allows for approximately 8.4 hours of maintenance activities per week for this system.

Figure 6.2 shows the resulting mass-crew time tradespace. The system defined above has 72,224 feasible supportability strategies, which require approximately 13.7 minutes of computational time to evaluate using MCROSS. Pareto-optimal strategies

Table 6.2: Item list for the notional system examined here. The associated indenture structure is shown in Figure 6.1.

Item	Parent	Component	Subsystem	Maximum Functional Distribution	Maintenance Crew Time (CM-h)	Manufacturability	QPA	Baseline Unit Mass (kg)	Life Limit (h)	Duty Cycle	Mean Failure Rate (h^{-1})	Failure Rate Error Factor	Mean Crew Action Rate (h^{-1})	Crew Action Rate Error Factor	K-Factor	Mass Scaling Factor	Maximum Redundancy
ϕ	p	θ	ψ	η_{max}	t	$\hat{\xi}$	$\hat{q}$	m	ℓ	d	$\bar{\lambda}$	ε	$\bar{\lambda}_C$	ε_C	κ	γ	r_{max}
1	0				0.50		1										
2	1		1	2	1.75		1										
3	1				0.50		2										
4	1				0.75		1										
5	2	1			7.00		1	20.0		1	5.0×10^{-6}	2.0	5.5×10^{-6}	2.0	1.2	0.6	0
6	2	1			2.00		1	15.0	7,200	1	5.0×10^{-7}	2.5	6.0×10^{-7}	2.3	1.1	0.6	0
7	2	1			2.50		1	60.0		1	1.5×10^{-6}	2.0	2.1×10^{-6}	1.9	1.1	0.6	0
8	3	1			1.25	1	1	6.0	3,600	1	1.0×10^{-5}	2.0	1.5×10^{-5}	2.1	1.3	0.6	1
9	3	1			1.00	1	2	7.5		1	3.5×10^{-5}	2.2	7.0×10^{-5}	2.0	1.2	0.6	1
10	3	1			1.50	1	1	10.5		1	5.2×10^{-6}	2.1	7.8×10^{-6}	1.8	1.2	0.6	0
11	4	1			1.50		1	10.0		1	3.0×10^{-6}	2.0	4.5×10^{-6}	1.9	1.3	0.6	0
12	4	1			1.00	1	1	11.0		1	4.0×10^{-6}	2.0	6.0×10^{-6}	2.1	1.1	0.6	0
13	4	1			1.25	1	3	2.5		1	5.1×10^{-5}	1.8	6.0×10^{-5}	2.0	1.1	0.6	1

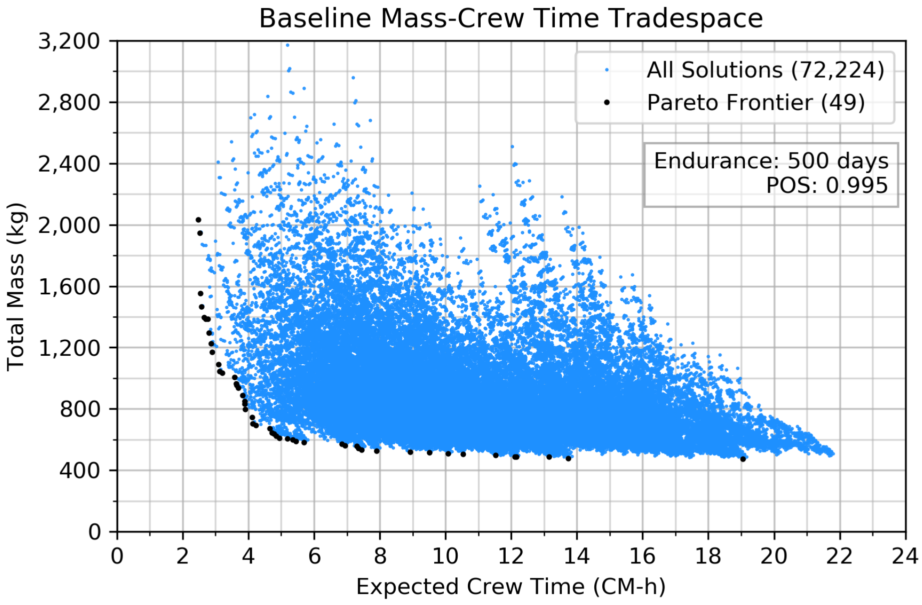

Figure 6.2: Mass-crew time tradespace for notional case. All 72,224 feasible solutions are plotted in blue, and the 46 Pareto-optimal points are plotted in black.

are plotted in black. All dominated strategies are plotted in blue. Total mass ranges from 475.0 kg to 3,170.5 kg, while expected crew time ranges from 2.5 CM-h to 21.8 CM-h. The Pareto frontier contains 49 solutions, and has corner points at (2,033.0 kg, 2.5 CM-h) and (475.0 kg, 19.0 CM-h).

As expected, there is a distinct tradeoff between mass and maintenance crew time for this system. The fact that this tradeoff exists can be identified directly from the indenture structure and maintenance crew time values listed in Table 6.2, which show that (for this notional system) lower level maintenance requires increased crew time. However, this MCROSS analysis reveals the specific shape of this tradeoff – specifically, the sharpness of it. Starting from the highest-mass, lowest-crew-time corner point (2,033.0 kg, 2.5 CM-h) mass can be decreased by nearly 1,340 kg, to 693.8 kg, while only increasing expected crew time marginally, to 4.2 CM-h. Continuing along the Pareto frontier, however, the crew-time-cost of lowering mass begins to increase rapidly. At the low-mass, high-crew-time end of the Pareto frontier, a tripling of expected maintenance crew time results in a mass savings of just over 100 kg.

Figure 6.3 shows the supportability strategies associated with each Pareto-optimal

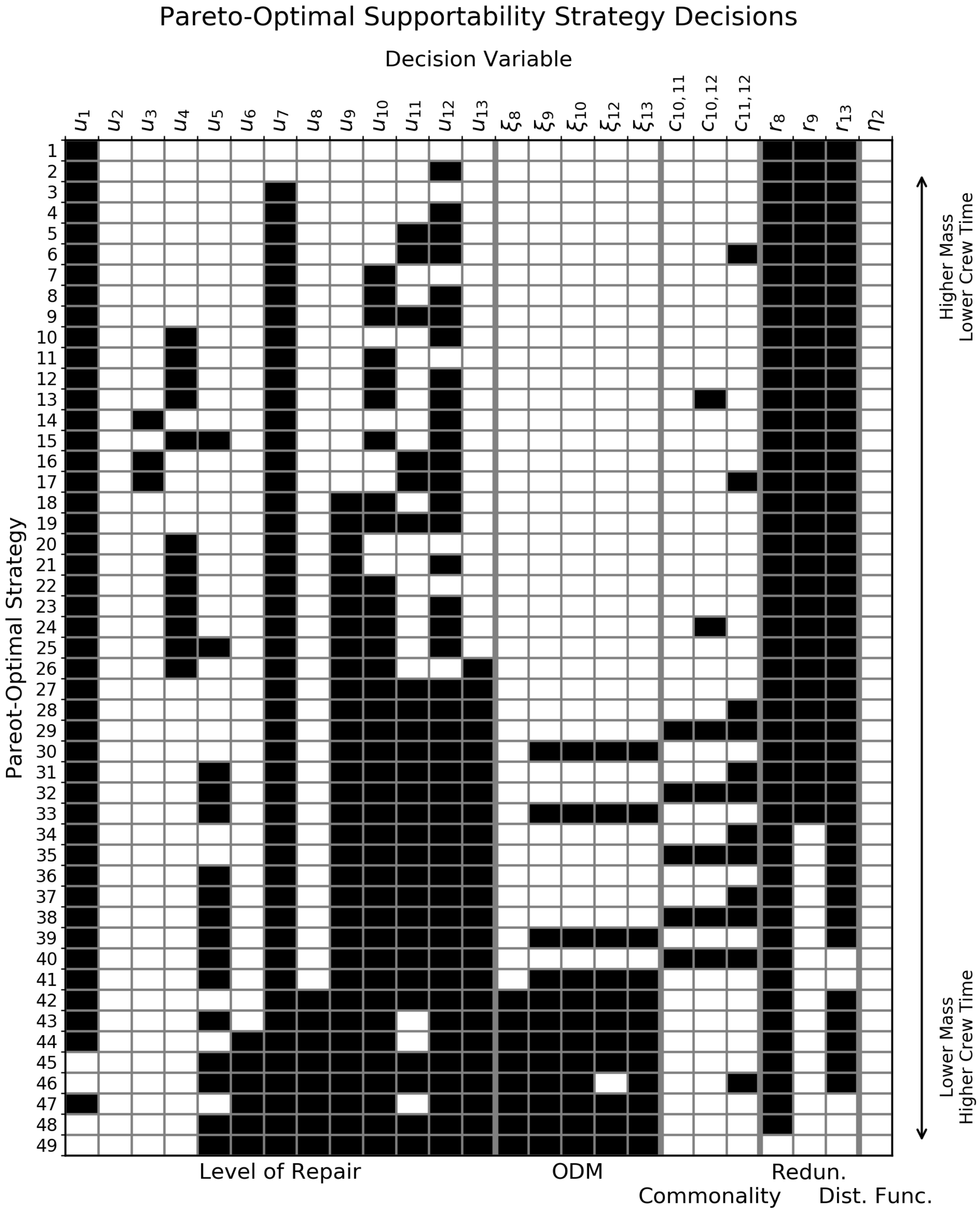

Figure 6.3: Supportability strategy decisions along the Pareto frontier. Each row represents a Pareto-optimal strategy, sorted in order of decreasing mass and increasing crew time (see Figure 6.4). Columns correspond to decision variables, and labeled at the top with the specific variable and at the bottom with the associated strategy element. Values are indicated by the cell fill color, with white corresponding to the low end of that variable's domain and black corresponding to the high end.

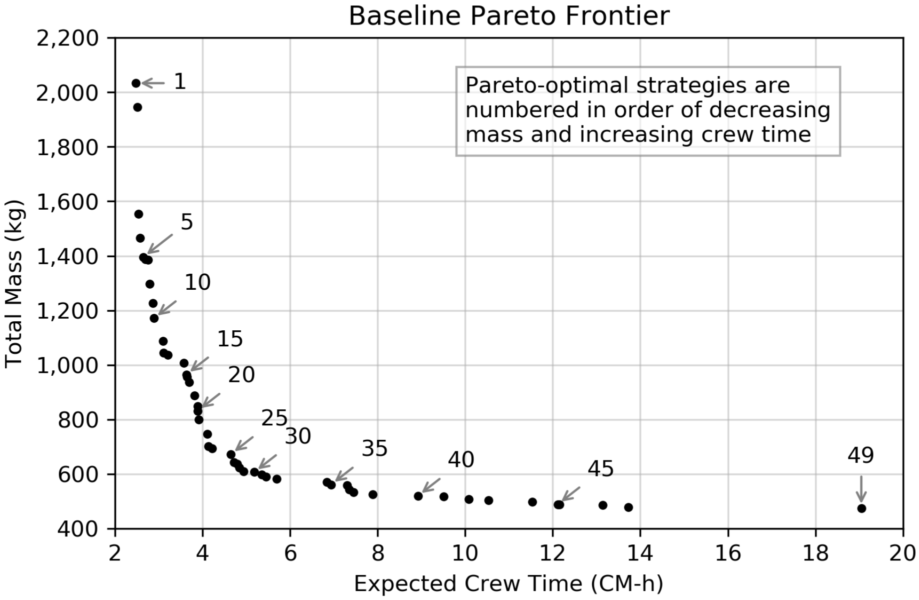

Figure 6.4: Pareto frontier for notional case study. Each Pareto-optimal solution is numbered in order of decreasing mass and increasing crew time, from the upper left corner point to the lower right. These numbers correspond to the row in Figure 6.3 showing the supportability strategy decisions associated with that point.

solutions, which are shown in Figure 6.4. Each strategy is numbered in order of descending mass and increasing crew time, from the top left corner point to the lower right corner point of the frontier. The rows of Figure 6.3 correspond to these points (in order), and the columns indicate the particular supportability strategy used. White cells indicate that a supportability strategy decision variable takes the lowest value in its domain, while black cells indicate that it takes the highest value. For binary variables, these correspond directly to 0 and 1, respectively. Redundancy and distributed functionality are shown in grayscale, with white corresponding to 0 for redundancy and 1 for distributed functionality, and black corresponding to r_{max} and η_{max}, respectively, for that item. The specific variable for each column is labeled at the top, while groups of columns representing the five supportability strategy elements (level of maintenance, ODM, commonality, redundancy, and distributed functionality) are labeled at the bottom and separated by thicker vertical lines. Variables whose domains are constrained to one value by the problem definition, such as commonality decisions that are not included in the potential commonality matrix or ODM decisions for non-manufacturable items, are not shown in order to conserve space. Similarly, ξ_s is not shown since it is easily identified from the ODM decisions ξ.

Figure 6.3 indicates that, as expected, the strategy with the highest mass and lowest crew time chooses the highest possible level of maintenance. The only replaceable item is the root node (item 1), which also has the lowest maintenance crew time. At the other end of the spectrum, the strategy which uses the lowest level of maintenance (i.e. all components are replaceable items) results in the lowest mass and highest crew time. Intermediate strategies show a gradual shift from a high level of maintenance to a low level of maintenance. ODM also becomes more prevalent at the low-mass end of the Pareto frontier. ODM reduces maintenance logistics mass, but can only occur when manufacturable items are replaceable items. Since these items are lower in the indenture structure, they are also correlated with higher crew times.

An interesting trend occurs between commonality and ODM, since items cannot simultaneously use commonality and ODM. ODM is a less constrained version of commonality, so an item that does both would pay the cost of commonality in terms of taking the worst characteristics of the common set when it would not need to if it just used ODM. In this case, some manufacturable items are potentially common with items that are not manufacturable. If those common pairs are activated, then the manufacturable item cannot use ODM (since doing so would imply that the item it is common with is also manufacturable, which is not the case). As a result, in the region of the Pareto frontier between strategies 29 and 41 there are several instances where application of commonality alternates with application of ODM. In the end, however, ODM provides greater mass reduction than commonality, and the lowest-mass end of the Pareto frontier consists entirely of strategies that use ODM.

The sequence of redundancy decisions also shows the expected trend: strategies that have high mass and low crew time use full redundancy, while strategies with low mass and high crew time use no redundancy. For this system, item 8 was the only life-limited item with a redundancy option, and as a result it is also the last item to give up redundancy. Since it requires replacement every 150 d, three replacements are required on a 500 d mission when no redundancy is present, each of which requires

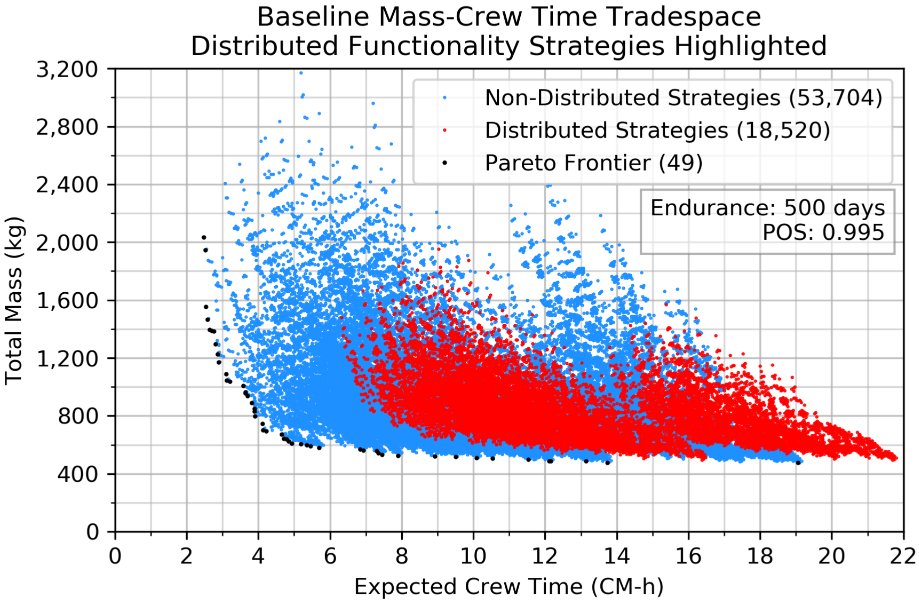

Figure 6.5: Mass-crew time tradespace with solutions using distributed functionality highlighted in red.

0.5-1.25 CM-h of crew time, depending on the level of maintenance configuration. If a single redundant instance is added, the effective life limit is 300 d and only one replacement would be required. Therefore, redundancy for item 8 can reduce preventative maintenance crew time for item 8 by 1 to 2.5 CM-h, a significant impact for a single variable. That additional redundancy would also reduce crew time associated with corrective maintenance actions.

For this notional system and baseline case, no Pareto-optimal strategies use distributed functionality. Figure 6.5 show the positions of strategies that use distributed functionality by highlighting them in red. These strategies that use distributed functionality, are grouped in the lower right section of the tradespace. These strategies tend to result in increased crew time because splitting an item across multiple modules increases the number of that item in operation, and therefore the amount of crew time associated with it. From a mass perspective, these strategies tend to group in the lower half of the tradespace for two reasons. First, due to the constraint between level of maintenance and distributed functionality, the option to split item 2 and its descendants across multiple modules will only be implemented in this problem if the selected level of maintenance is at or below item 2. These strategies will tend to be lower mass due to the lower level of maintenance. In addition, splitting an item can reduce corrective maintenance mass by producing two, smaller items that can use common spares.

The strategies near the knee of the Pareto frontier (approximately strategies 27-33 in Figures 6.3 and 6.4) tend to define maintenance at a low level in the indenture structure while still maintaining redundancy. In some cases (e.g. strategy 29), all commonality options are used; in others (e.g. strategy 30), almost all ODM is used. This combination of approaches balances the reduction in spares mass requirements via lower-level maintenance and commonality/ODM – which all tend to increase crew time requirements – with redundancy, which reduces crew time requirements. Redundancy can also reduce mass if the resulting reduction in the number of R&R events outweighs the increase in ORU mass individual instances of each ORU that incorporates redundant components. Strategies resulting in higher mass (i.e. to the left of the knee point) tend to perform maintenance at higher levels in the indenture structure, do not include ODM or significant commonality, and apply full redundancy. As a result, they experience higher spares mass requirements and lower crew time requirements. Conversely, strategies with lower mass (i.e. to the right of the knee) tend to have low level maintenance decisions with less redundancy, and therefore experience lower mass requirements at the cost of higher crew time.

6.3 Validation of Optimization Strategy

In order to provide a complete accounting of the available supportability tradespace, the above sections made use of a full enumeration of supportability strategies. While this approach is feasible (if slow) for small- to medium-sized problems such as this one, it quickly becomes infeasible as problem size grows; this is the driver behind formulating the supportability strategy optimization problem as a multiobjective Constraint Optimization Problem (COP) and implementing the optimization strategy described in Section 4.3. In order to validate this optimization approach, the results obtained using that approach were compared to the results from full enumeration presented above. OpSat [119, 120] was used as the multiobjective COP solver, while Gecode [121] was used for enumeration.

The optimizer correctly identified all Pareto-optimal strategies in this test case. This means that the Pareto frontier found by the optimizer contained all Pareto-optimal strategies identified via full enumeration, and did not contain any strategies that were not Pareto-optimal. In terms of computational complexity, the optimizer examined a total 10,896 candidate strategies. Of these, 5,780 were partial assignments to the decision variables, and 5,116 were full assignments. In comparison to the 72,224 strategies identified during full enumeration, these results indicate that pruning of subtrees during multiobjective branch and bound reduced the size of the tradespace that needed to be explored (including partial evaluations) to just over 15% of the full tradespace. Even with the additional computational overhead associated with interactive communications between the optimizer and Python evaluator,[a] optimization required approximately 111.5 s (1.86 minutes), or just over 13.5% of the time required for full enumeration. These results highlight the significant increases in computational efficiency resulting from application of multiobjective branch and bound to prune the tradespace. As problem size increases, enumeration-based approaches rapidly become intractable. Application of optimization tools such as OpSat [119, 120] is often necessary for examination of real-world systems, such as the case study presented in Chapter 7.

6.4 Sensitivity Analyses

This section presents a series of sensitivity analyses in order to examine the impact of key parameters on the mass-crew time tradespace, as well as to demonstrate how MCROSS could be used as part of systems analysis. Section 6.4.1 examines how

[a] When full enumeration is used, Gecode [121] can simply return a list of solutions to evaluate, which can then be iterated over by the evaluator. To perform in-the-loop optimization, OpSat [119, 120] and the supportability evaluator have to communicate actively during the optimization process as the solver passes partial/full assignments to the evaluator and waits for an output.

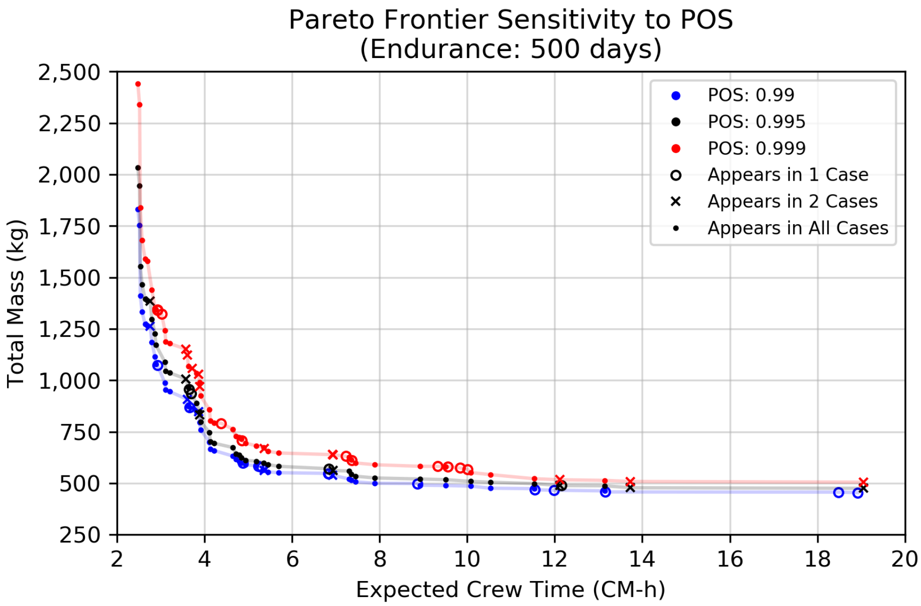

Figure 6.6: Sensitivity of Pareto frontier to changes in POS.

the Pareto frontier shifts as a function of the POS requirement, while Section 6.4.2 examines the impact of mission endurance.

6.4.1 Required POS

In order to examine the impact of the POS requirement on mass, crew time, and optimal supportability strategies, the analysis was repeated for POS values of 0.99 and 0.999, keeping mission endurance fixed at 500 days. Figure 6.6 shows the results of this sensitivity analysis, including the Pareto frontiers for a POS of 0.99 (blue), 0.995 (black), and 0.999 (red). Strategies that are unique to a particular Pareto frontier are indicated by an empty circle, while strategies that appear in two out of the three cases are indicated by an X and strategies that appear in all three Pareto fronts are shown by a dot.

Changes in required POS primarily impact the supportability characteristics of a particular system and strategy by changing the amount of spares that need to be carried in order to achieve that requirement. As a result, expected crew time results for a particular strategy are not affected by changes in required POS, and the only changes in the Pareto frontier related to crew time are driven by the appearance of new Pareto-optimal strategies as the total mass of all strategies changes. Partially as a result of this crew time insensitivity, the Pareto frontiers in all cases appear very close together in the tradespace. However, mass values can be significantly affected.

Higher POS requirements mean that more spares are required (as shown in Figures 5.19 and 5.20), and the results shown in Figure 6.6 indicate that the sensitivity of total mass to required POS is strongly driven by the selected supportability strategy. At the upper left corner of the Pareto frontier, a change in required POS from 0.99 to 0.999 results in a mass increase of over 600 kg. At the middle of the Pareto frontier, the difference drops to just under 150 kg, and at the lower right corner the difference is approximately 65 kg. In terms of percentage of mass growth from a required POS of 0.99 to 0.999, these correspond to approximately 33%, 23%, and 15%, respectively. Therefore, low-mass, high-crew-time supportability strategies are significantly less sensitive to changes in POS than high-mass, low-crew-time strategies. This is likely driven by the fact that lower levels of maintenance (which are correlated with low mass and high crew time) enable more efficient spares allocations, and as a result can adapt to higher POS requirements without significant mass increases.

Another important result here is that the set of Pareto-optimal architectures is relatively stable across all three POS levels. In each case, the majority (63-78%) of the solutions in a given Pareto front are solutions that appear in all three Pareto fronts. The 0.99 case has the highest proportion of unique strategies (23%), and this is likely driven by the fact that this shifted POS requirement incurs a lower mass increase for points at the low end of the crew time range than for those at the high end of the range, while keeping crew time the same. That is, the 0.995 and 0.999 POS Pareto frontiers are "tilted" towards higher masses, and that tilt can uncover new strategies that were previously dominated. However, in general these results indicate that Pareto-optimal supportability strategy decisions are relatively robust to changes in risk posture. The costs associated with those strategies may change, but the strategies that will minimize those costs tend to stay the same.

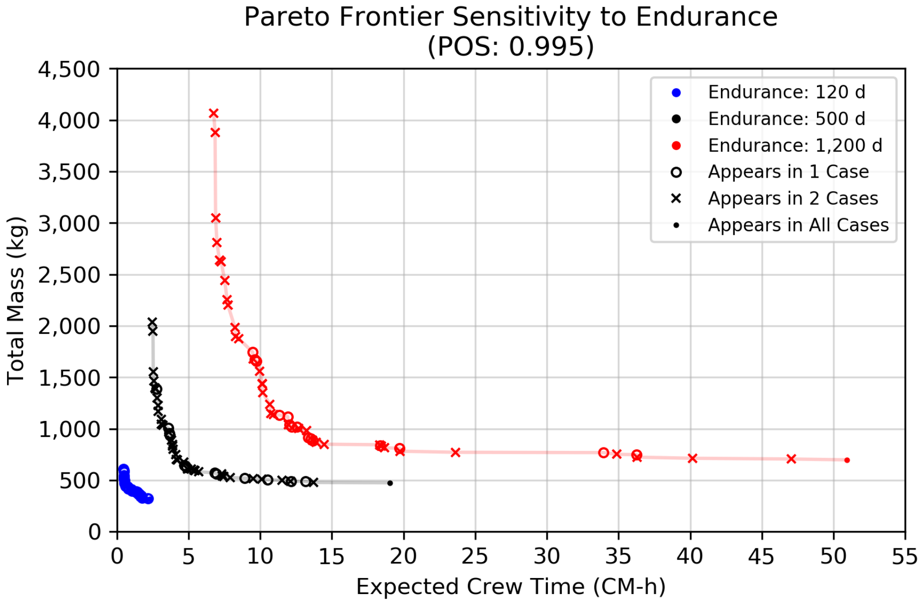

Figure 6.7: Sensitivity of Pareto frontier to changes in endurance.

6.4.2 Mission Endurance

This section performs a similar analysis to the one described in Section 6.4.1, but varies mission endurance instead of POS. Mission endurances of 120 and 1,200 days were examined in addition to the baseline case described above, all with a required POS of 0.995. In addition, in each case the maximum crew time available for maintenance was set to 5% of total mission endurance, or about 8 hours per week. Figure 6.7 shows the results of this analysis, with the Pareto frontiers corresponding to mission endurances of 120, 500, and 1,200 days shown in blue, black, and red, respectively. As before, strategies that are unique to a particular case are indicated by an empty circle, those that appear in two cases are indicated by an x, and those that appear in all cases are indicated by a dot.

Changes in mission endurance have significant impacts on both mass and crew time. Where the changes in POS described above primarily impacted mass by changing the amount of spares that needed to be carried, mission endurance impacts all logistical elements of the mission. Longer missions mean more maintenance events and more crew time requirements. They also require more scheduled replacements, and more spares to achieve the same POS level. As a result, the Pareto frontier moves up and to the right as endurance increases.

In addition, changes in mission endurance can have a significant impact on the set of supportability strategies that are Pareto optimal. In the POS sensitivity analysis above, most supportability strategies appeared in all three cases. In this case, no strategies appeared in every case. However, there do appear to be specific breakpoints in mission endurance that cause shifts in general strategy. Over 92% of strategies that are Pareto-optimal for the 120-day case are unique to that case. One strategy from this case also appears in the 500-day case but not the 1,200-day case, and another appears in the 1,200-day case but not the 500-day case. Only one strategy – the one corresponding to the lowest mass and highest crew time – appears in all three cases. In contrast, approximately 70% of the strategies appearing in the 500-day case also appear in the 1,200-day case, and vice versa. This suggests that there is a family of supportability strategies that are Pareto-optimal for long-endurance missions and a different family of strategies that are Pareto-optimal for shorter missions.

One key difference between the strategies used for shorter missions and those used for longer missions is that redundancy is much less prevalent in the former, particularly for item 8. This is the only life-limited item with the option to use redundancy, but the life limit (150 d) is longer than the 120-day mission being examined, and as a result there are no scheduled replacements. As a result, while redundancy can directly decrease the amount of preventative maintenance crew time required in longer missions by doubling the life limit of that component, this benefit disappears for shorter missions and redundancy for item 8 becomes a dominated strategy relatively quickly when moving from the upper left to the lower right end of the Pareto frontier.

The only Pareto-optimal strategy that appears all three cases is the lowest-mass option, which corresponds to strategy 49 in Figure 6.3. That is, the point on the lower right corner of each Pareto frontier corresponds to the same strategy, which minimizes level of maintenance and maximizes ODM. For this point, increasing endurance from 120 days to 550 days (a 317% increase) changes total mass from 318.9 kg to 475.0 kg (+49%) and expected crew time from 2.3 CM-h to 19.04 CM-h (+731%). A further increase in endurance, to 1,200 days, results in a mass of 697.5 kg (+119% from the 120 day case) and a crew time of 50.9 CM-h (+2,122%). These results indicate that solutions that minimize mass in exchange for higher crew times tend to be more sensitive to changes in mission endurance from the perspective of maintenance crew time, and less sensitive from a mass perspective. At the other end of the Pareto frontier, the upper left corner point is the same for both the 500-day and 1,200-day cases. For this point, increasing endurance from 550 days to 1,200 days (a 140% increase) changes total mass from 2,033 kg to 4,066 kg (+100%) and expected crew time from 2.5 CM-h to 6.8 CM-h (+172%). While proportionally the increase in crew time is still greater than the increase in mass, the changes for this point indicate that strategies that allow higher mass in order to minimize crew time tend to have total mass values that are more sensitive to changes in mission endurance.

6.5 Summary

This chapter has presented the application of MCROSS to a notional case study, designed to have a tradespace that enables full enumeration. The results of a baseline case are presented and examined, and Pareto-optimal strategies are identified, described, and discussed. A series of sensitivity analyses were executed to examine the impacts of changes in POS requirement, mission endurance, and ODM system mass on the set of Pareto-optimal supportability strategies and associated mass and crew time requirements. These analyses provided insights into the response of different supportability strategies to different conditions, as well as the sets of conditions under which particular strategies may be useful. Finally, the results of full enumeration were compared to those generated via solution of the optimal COP defined in Chapter 3 in order to validate that the Pareto frontier is found, and in order to examine the efficiency of the search strategy. This validation confirmed that the optimization approach finds all Pareto-optimal strategies, and does not incorrectly identify non-optimal strategies as part of the Pareto frontier. In addition, optimization significantly reduced both the number of candidate solutions that were explored as well as the amount of time required to perform the analysis.

THIS PAGE INTENTIONALLY LEFT BLANK

Chapter 7

Case Study: ISS Oxygen Generation Assembly

This chapter applies MCROSS to supportability strategy optimization for oxygen generation systems, using the ISS OGA as a baseline. Section 7.1 describes the system examined in this case study, and Section 7.2 presents the results of a baseline analysis examining the mass and crew time required for this system on a notional 1,200-day Mars mission with a POS requirement of 0.995. Section 7.3 then performs a series of sensitivity analyses examining the impact of changes in POS, mission endurance, maintainability (which characterizes the increase in crew time required to implement lower-level maintenance), and ODM system mass. Finally, Section 7.4 discusses and summarizes the results of this case study.

7.1 System Description

This analysis examines variations on the current ISS OGA, which is one of the systems examined in the validation presented in Section 5.2. The OGA uses electrolysis to convert water into hydrogen and oxygen and deliver that oxygen to the spacecraft cabin to supply crew needs. The hydrogen is either sent to a Sabatier Carbon Dioxide Reduction Assembly or vented to space, depending on the particular operational configuration used. A simplified schematic of the ISS OGA is shown in Figure 7.1, showing major ORUs and components. Figure 7.2 shows major OGA ORUs as well as the installed OGA rack configuration [145, 146]. A more detailed description of the ISS OGA is presented by Takada et al. [143] and Takada et al. [105].

In its current configuration, the OGA consists of 10 ORUs. These include the ORUs listed in Table 5.2 as well as the ACTEX unit assembly, which was not included in the validation case due to lack of MADS data. For the purposes of this case study, three to six lower-level items were identified within each of the three largest ORUs. These subassemblies and components typically account for a significant portion of the ORU initial failure rate estimate, but a much smaller percentage of ORU mass, indicating that logistics efficiency could potentially be increased by making them individually replaceable. Particular attention was paid to identifying lower-level items within the Hydrogen ORU,[a] which with a mass of over 120 kg is by far the largest ORU in the OGA and has previously been examined as a potential opportunity to reduce logistics mass by performing maintenance at the subassembly level rather than ORU level [105]. With the inclusion of these lower level items, this case study examines a total of 24 items, arranged in a two-level indenture structure. Since the maintenance logistics mass associated with the current OGA level of maintenance is already high enough to be a significant challenge for deep-space missions, higher levels of maintenance (e.g. performing maintenance by swapping out entire OGAs) were not examined in this case study.

The full ISS dataset described in Section 5.2 is used to update item parameters, using initial failure rate estimates found in MADS and other sources. Where values were not available, they were estimated based on similarity to other components. A Bayesian update was performed at the lowest known level of maintenance by associating failure events described in Section 5.2 with particular subassemblies within each ORU in order to characterize failure rate and crew action rate estimates for each component. ORU-level crew time estimates are updated to the observed average crew time per event for that ORU if there are maintenance events for that ORU in the dataset; otherwise the MADS CMMTTR estimate was used.

[a] Also called the Hydrogen Dome ORU by some sources.

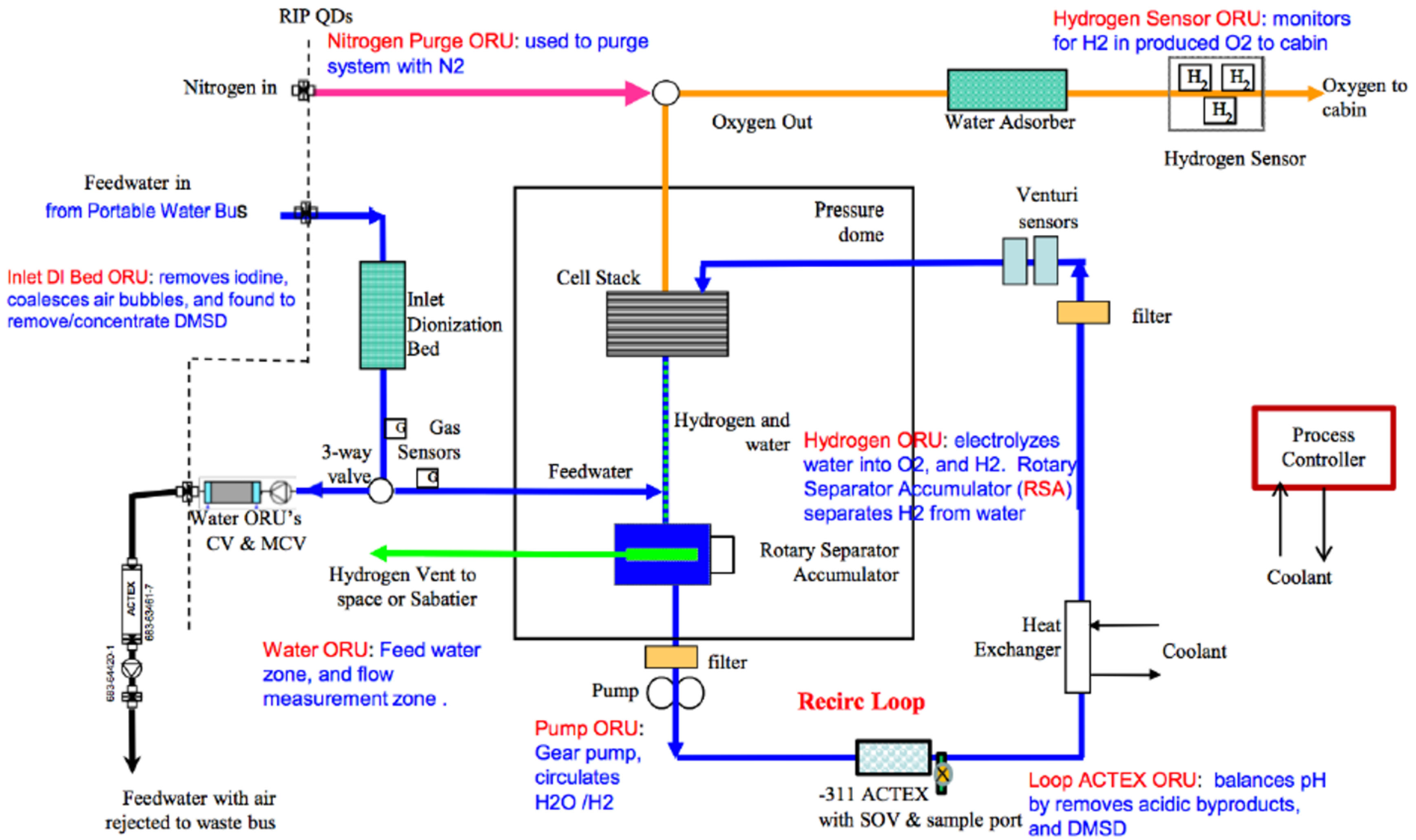
RIP QDs
Nitrogen Purge ORU: used to purge system with N2
Nitrogen in
Hydrogen Sensor ORU: monitors for H2 in produced O2 to cabin
H2
H2
H2
Oxygen to cabin
Oxygen Out
Water Adsorber
Hydrogen Sensor
Feedwater in
from Portable Water Bus
Inlet DI Bed ORU: removes iodine, coalesces air bubbles, and found to remove/concentrate DMSD
Inlet Dionization Bed
Pressure dome
Venturi sensors
Cell Stack
filter
Gas Sensors
G
G
3-way valve
Hydrogen and water
Hydrogen ORU: electrolyzes water into O2, and H2. Rotary Separator Accumulator (RSA) separates H2 from water
Process Controller
Feedwater
Water ORU's CV & MCV
Hydrogen Vent to space or Sabatier
Rotary Separator Accumulator
Coolant
ACTEX
683-63461-7
683-64420-1
Water ORU: Feed water zone, and flow measurement zone .
filter
Heat Exchanger
Coolant
Pump
Recirc Loop
Pump ORU: Gear pump, circulates H2O /H2
-311 ACTEX with SOV & sample port
Loop ACTEX ORU: balances pH by removes acidic byproducts, and DMSD
Feedwater with air rejected to waste bus

Figure 7.1: Simplified schematic of the ISS OGA [105].

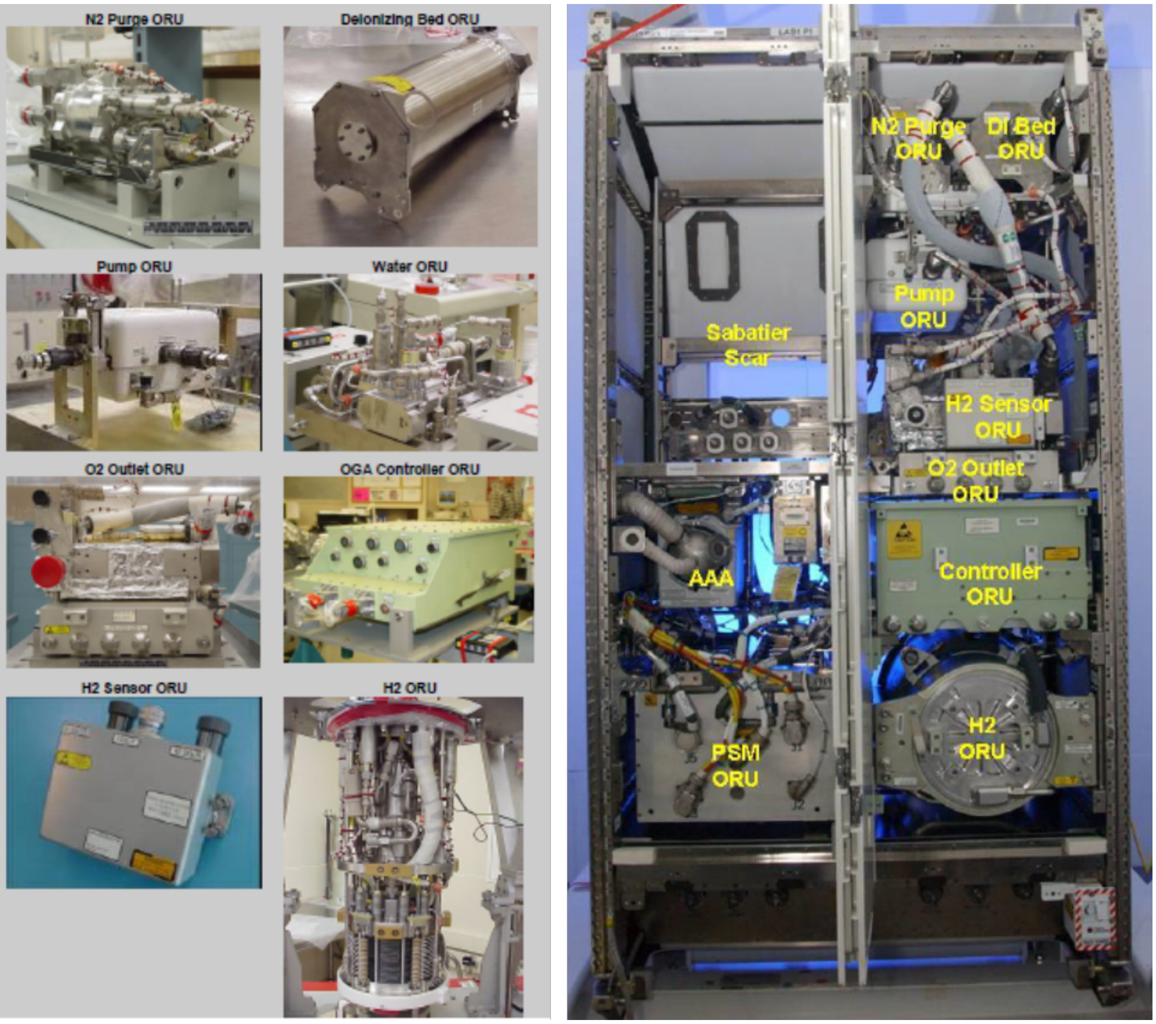

Figure 7.2: Photographs of individual OGA ORUs (left) [145], and the OGA in its ISS rack configuration (right) [146]. The gap in the top left section of the rack in the image on the right is reserved for the Sabatier, which had not yet been installed when this picture was taken.

Since the historical maintenance crew time data listed in MDC and the CMMTTR estimates provided in MADS only relate to the ORU level implemented on ISS, crew time data related to lower levels of maintenance were not available. Therefore, this case study estimates the maintenance crew time associated with lower-level items using a maintainability factor α and the following equation

$$t_i = t_\omega \left(1 + a \ln\left(\frac{\bar{\lambda}_\omega}{\hat{q}_i \bar{\lambda}_i}\right)\right) \tag{7.1}$$

where t, $\bar{\lambda}$, and $\hat{q}$ are crew time, mean failure rate, and QPA, respectively, with a subscript i used to indicate the value associated with a lower-level item and a subscript ω used to indicate the value associated with the current ISS ORU that the item is contained within. This equation specifies that the increase in crew time required to go to a lower level of maintenance for a particular item is proportional to the natural logarithm of the inverse of the proportion of the ORU failure rate that is contributed by the lower-level item. That is, items that contribute more to the ORU failure rate will have a lower crew time cost associated with making them maintainable at a lower level, based on the assumption that these items would be the ones for which the most design effort would be spent to enable lower-level maintenance, if it were an option. Variation of the maintainability parameter a can examine different scenarios in which it is easier/harder (in terms of crew time requirements) to perform maintenance at a lower level. As a baseline, its value is initially set to 0.2.

Other supportability decisions are available throughout the system where appropriate. Potential commonality is specified for valves, which appear in multiple locations within the system. Five items are potentially manufacturable, selected due to their function being primarily due to shape rather than chemical composition or electronic functionality. Redundancy is available for the cell stack and Rotary Separator Accumulator (RSA) within the Hydrogen ORU, since these are primary drivers of failure rates for a large, complex subsystem, as well as the Hydrogen Sensor ORU, which has a short life limit and is therefore a driver of scheduled maintenance mass and crew time. The Hydrogen ORU itself is considered a subsystem that could be split across multiple modules. The supportability strategy tradespace resulting from all of these options is orders of magnitude larger than the one examined in Chapter 6.

It is important to note that all of the supportability options described here, as well as the system characteristics used in this model, are based on the best available information at the time of this dissertation but are not meant as precise predictions of the characteristics and supportability strategies used on future missions. In addition, these systems are examined in the context of a Mars mission, even though they are designed for and operate in a LEO environment, and as a result their characteristics are not necessarily representative of the systems that would be used for future missions. This analysis approach is taken because these systems represent the current state-of-the-art in long-duration ECLSS, they have decades of orbital heritage that has enabled significant reduction in risk [40], and it is considered highly likely that future deep-space life support systems would be evolutions of the technologies and designs used on the ISS rather than entirely new systems. As technology develops, it is fully expected that systems will change, new opportunities for improved supportability strategies will arise, or previously planned approaches will be adjusted in response to new information. This analysis is only intended to indicate what the potential impacts of different strategies could be in order to inform technology investment and system development. That is, this supportability analysis provides context and forecasting, not exact predictions. The insights gained from sensitivity analysis or examination of a range of results for different strategies are likely more valuable from the perspective of mission design than the specific results for a particular point solution.

7.2 Baseline Case

Figure 7.3 shows the mass/crew time Pareto frontier for the OGA on a baseline 1,200-day Mars mission with a POS requirement of 0.995. The maximum crew time allowed for maintenance is set to 5% of total mission endurance (60 days, or 8.4 CM-h

Table 7.1: OGA case study modeling parameters.

Parameter	Symbol	Value	Units
Redundancy Mass Overhead Factor	γ_r	0.05	
Redundancy Effectiveness Factor	$\gamma_{\Lambda r}$	0.98	
ODM Mass Overhead Factor	γ_{odm}	0.05	
ODM Failure / Action Rate Factor	$\gamma_{\Lambda odm}$	0.1	
ODM System Mass	m_{odm}	20	kg
Crew Time Discretization	Δt	0.25	CM-h
ODM Feedstock Discretization	Δm	0.25	kg
Maximum Inversion Error	ϵ_{max}	1×10^{-7}	

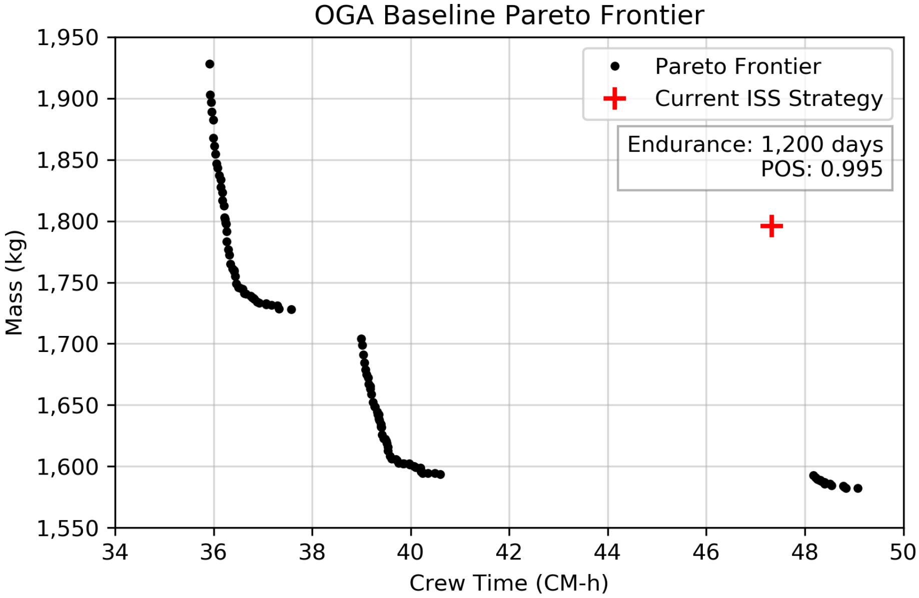

Figure 7.3: Baseline Pareto frontier for OGA case study.

per week). All other modeling parameters used in this analysis are summarized in Table 7.1. MCROSS examined 97,294 candidate strategies, including 91,543 partial and 5,751 full assignments to the supportability strategy decision variables. This analysis was completed in approximately 36 minutes, and identified 109 Pareto-optimal strategies.

7.2.1 Comparison to Current ISS Strategy

For context, the current ISS strategy (i.e. the current ORU definitions, without redundancy, commonality, ODM, or distributed functionality) is indicated by the red plus sign and corresponds to a mass of 1,796.0 kg and 47.3 CM-h of maintenance crew time. Under the assumptions used in this analysis (including the failure rate / crew action rate dataset), this strategy is a significant distance from the Pareto frontier in terms of both mass and crew time. A corner point in the middle of the Pareto frontier, at approximately 1,606.1 kg and 39.6 CM-h, provides a 189.9 kg reduction in mass (-10%) and 7.7 CM-h reduction in maintenance crew time requirements (-16.3%) over the current strategy. The Pareto frontier also contains solutions that provide similar crew time requirements with a 203.2 kg reduction in mass (-11.3%), or similar mass requirements with a 12.0 CM-h reduction in maintenance crew time (-25.4%).

The primary drivers behind the strategies that improve upon the current approach are the application of targeted lower-level maintenance as well as targeted redundancy. Implementing maintenance at a lower level separates subcomponents that are currently primary drivers of their ORU's failure rate (i.e. the failure rate of the current ISS ORU) into a separate ORUs, reducing the amount of mass that must be replaced each time they fail. As noted earlier, lower-level maintenance has previously been identified as a valuable approach for the Hydrogen ORU [105]; this analysis confirms the value of that approach, and extends it to other subsystems. Component-level redundancy reduces crew time requirements by reducing the number of failures or scheduled replacements that occur, and can also reduce mass if the logistics savings outweigh increases in ORU mass. This approach is particularly powerful when applied to components that are key drivers of overall crew time requirements, specifically those with a high failure rate and/or short life limit and a large maintenance crew time per replacement.

These results demonstrate that MCROSS can identify solutions that, from a supportability perspective, are better than existing strategies. Since the results of the analysis are a Pareto-optimal set rather than a single strategy, MCROSS provides decision-makers with a range of options to select from, allowing them to make informed trades between mass and crew time requirements for a given mission and balance supportability considerations against other constraints and mission objectives. This provides a powerful capability to enable effective design to manage supportability mass and crew time requirements on future missions. However, it is important to note that these results – and the Pareto-optimality of the identified solutions – only capture the supportability considerations that are included in the model. There are likely other design-driven constraints regarding the use of redundancy within current systems, or regarding opportunities for lower-level maintenance. As a result, they should always be viewed in the context of other factors that drive design decisions.

For example, the Hydrogen ORU is large in part because it consists of a containment dome surrounding electrolysis equipment. In the event of a failure, this equipment could leak hydrogen and cause a fire hazard, or (in a catastrophic case) even an explosion hazard; the dome serves to contain any damage that might occur. However, this dome was initially included in the design as a conservative safety measure, commercial and military electrolysis systems without containment domes have long safety records, and recent work has examined the possibility of removing the dome or making the inside of the dome accessible for maintenance [105]. Supportability analyses such as this one can be adapted to any constraints derived from more detailed design considerations in order to continue informing the process. For example, if a safety requirement is implemented specifying that the Hydrogen ORU must remain a monolithic unit, then the analysis can easily be re-run while removing lower-level maintenance options on that item. The same is true for other constraints from other design considerations. Overall, supportability analysis is most valuable when integrated into an iterative design process that combines information from a range of system perspectives in order to inform decision-making.

7.2.2 Pareto Frontier Trends

Pareto frontier for this case has three distinct regions, separated by significant gaps. The differences between these sections is primarily driven by changes in the underlying redundancy strategy. The strategy at the top left end of the Pareto frontier (i.e. the highest-mass Pareto-optimal strategy) has a mass of 1,928.2 kg and an expected crew time of 35.9 CM-h. This strategy corresponds to the current level of maintenance decision – that is, the ORUs currently used on the ISS – with the addition of all available redundancy options. From this point, a shift to lower-level maintenance and removal of redundancy on the option with the lowest failure rate, along with some commonality, enable a 200 kg reduction in mass, accompanied by a 1.7 CM-h increase in crew time.

At this point (1,728.1 kg, 37.6 CM-h), the Pareto frontier abruptly shifts to higher maintenance crew time while simultaneously decreasing mass. The resulting point, marking the start of the next segment of the Pareto frontier, has a mass of 1,704.1 kg and an expected crew time of 39.0 CM-h The cause of this shift is the removal of redundancy for the electrolysis cell stack in the Hydrogen ORU. This cell stack is a primary driver of the ISS Hydrogen ORU failure rate, since all major failures of the Hydrogen ORU have been caused by faults within the electrolysis cell stack [105, 143]. It has a relatively high failure rate, and as a result is a significant driver of corrective maintenance crew time requirements and redundancy is valuable from a crew time perspective. Removal of this redundancy does allow for reduced logistics mass, since the electrolysis cell stack is a relatively large subcomponent of the Hydrogen ORU and the inclusion of a redundant instance therefore comes at a relatively high mass cost. However, removal of the redundant instance causes a significant increase in the expected number of maintenance actions associated with the electrolysis cell stack.

In addition, in this model the electrolysis cell stack (along with all other subcomponents of the Hydrogen ORU) has a high maintenance crew time due to the high maintenance crew time associated with the Hydrogen ORU itself in relation to other OGA ORUs and the estimate described in equation 7.1. The higher maintenance crew times associated with the Hydrogen ORU are due in part to the fact that that ORU is one of only four OGA ORUs that experienced a failure during the time period used for Bayesian updating in the dataset discussed in Section 5.2. Of these four, it also experienced the greatest discrepancy between observed average maintenance crew time per failure and predicted CMMTTR (see Figure 5.15). Since ORUs that did not have maintenance actions recorded in the validation dataset used the predicted CMMTTR by default, and this CMMTTR tended to be lower than observed values in general, the net result is that the Hydrogen ORU has a significantly higher maintenance crew time than all other ORUs in this analysis, and as a result lower-level maintenance items derived from that ORU have higher maintenance crew times as well. In short, based on available data and the assumptions used in this analysis the Hydrogen ORU has a higher crew time cost associated with providing maintenance at a level below that currently used on the ISS. However, it is worth noting that it is possible and even likely that the crew times associated with the other ORUs are underestimated, based on the trends observed in Section 5.2, and therefore the significance of the Hydrogen ORU from a crew time perspective may be overstated relative to other ORUs.

The next section of Pareto frontier continues to reduce mass to just below 1,600 kg before another significant jump in crew time occurs. Within this section, a similar trend appears as the one observed in the first section – lower mass and higher crew time are again driven by shifts towards lower levels of maintenance and reduced redundancy, accompanied by limited commonality. The next major jump occurs when redundancy is eliminated on the Hydrogen Sensor ORU. The lowest-mass option that includes this redundancy sits at 1,593.8 kg and 40.6 CM-h. The next Pareto-optimal solution corresponds to a mass decrease of less than a kilogram, but a crew time increase of 7.6 CM-h. The Hydrogen Sensor ORU has a relatively short life limit, and as a result accounts for a very significant amount of preventative maintenance crew time in this system. When redundancy is implemented, this life limit is effectively extended, significantly reducing the number of maintenance actions – and therefore total amount of crew time – required for the Hydrogen Sensor ORU. The sensor itself has a relatively low mass, and as a result this change in redundancy has a much stronger impact on crew time than mass.

From this point forward, all remaining Pareto-optimal strategies involve no redundancy. Level of maintenance decisions are the primary driver of the slight decrease in mass and increase in crew time, though some commonality appears as well. The end point of the Pareto frontier, at 1,582.2 kg and 49.1 CM-h, corresponds to a strategy that almost entirely utilizes the lowest level of maintenance. Two components within the Water ORU which both have low masses, failure rates, and crew action rates remain combined in a higher-level assembly, likely because the crew time benefits of higher-level maintenance on these items outweighs the marginal mass benefit of maintaining each item independently.

7.3 Sensitivity Analyses

This section utilizes MCROSS to perform a series of sensitivity analyses examining the effect of variations of key parameters. These sensitivity analyses provide examples of how this analysis approach could be used during system and mission development to inform decision-making.

7.3.1 Required POS

The first sensitivity analysis examines variation in the required POS, which can inform tradeoffs between mission risk and cost. Figure 7.4 shows mass-crew time Pareto frontiers for the OGA on a 1,200-day mission for required POS values of 0.99, 0.995, and 0.999, corresponding to a 1 in 100, 1 in 200, and 1 in 1000 chance of having insufficient maintenance resources, respectively. Strategies that are Pareto-optimal in all three cases are connected by gray lines. These results indicate that, as with the notional case examined in Chapter 6, higher POS requirements incur greater mass costs, but tend to leave maintenance crew time unaffected. This is because the primary impact of the POS requirement is to change the number of spares required to mitigate risk to the desired level. This case, however, deals with larger system and a significantly longer mission endurance, and as a result, the impacts of changes to POS are much more significant.

Moving from a POS of 0.99 to 0.995 – a change equivalent to halving the risk associated with maintenance resource availability – results in a mass increase of approximately 140-170 kg across the majority of the Pareto frontier. The step from a

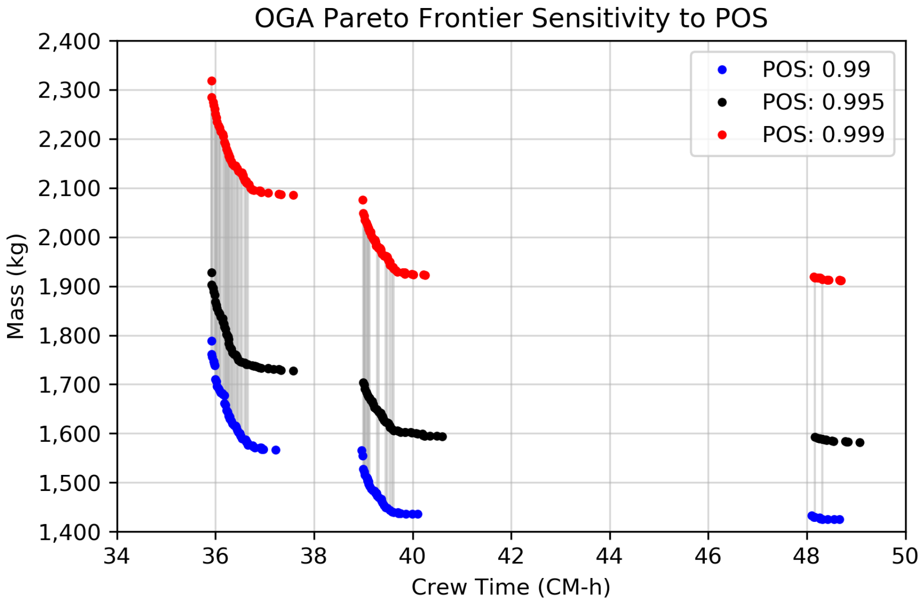

Figure 7.4: Sensitivity of OGA Pareto frontier to changes in POS requirement. The baseline Pareto frontier described in Section 7.2 (1,200 days, POS = 0.995) is shown in black. Strategies that appear in all three Pareto frontiers are connected by gray lines.

POS of 0.995 to 0.999 – which is equivalent to an additional five-fold reduction in maintenance resource-related risk – incurs an additional 320-390 kg mass increase. One key effect at higher POS values, is that in general the mass cost of achieving a higher POS requirement is larger at the high mass / low crew time end of the Pareto frontier, where strategies generally define ORUs at a higher level of maintenance. This higher level of maintenance results in larger ORUs and correspondingly less precisely targeted spares allocations, since more components are gathered into single units. At the other end of the Pareto frontier, where lower-level maintenance is more prevalent, individual spares tend to be smaller and as a result can be more efficiently allocated. Overall, however, all three Pareto frontiers have the same general shape, including the distinct jumps in maintenance crew time requirements described in Section 7.2.

These results also indicate that changes in POS can shift the set of Pareto-optimal strategies when lower-level maintenance is implemented, as indicated by the gray lines in Figure 7.4. In ascending POS order, the Pareto frontiers contain 99, 109, and 112 strategies each, and 43 strategies are Pareto-optimal at all POS levels. The gray lines connect strategies these strategies, which tend to be concentrated towards higher-level maintenance options. This is partially driven by the fact that level of maintenance decisions can impose significant constraints on most remaining supportability strategy decisions, especially at higher levels of maintenance. As a result, strategies that select a higher level of maintenance typically have a smaller tradespace of remaining decisions. At the right end of each section of the Pareto frontier, where maintenance is implemented at a lower level in the indenture structure, there are no strategies that are common across all cases. However, it is important to note that these sets include only the strictly Pareto-optimal strategies; it is likely that if each frontier were expanded to include near-optimal strategies there would be more overlap.

7.3.2 Mission Endurance

This section examines the impact of variation in mission endurance. The baseline mission examined above assumes a mission endurance of 1,200 days. However, the specific mission endurance for a Mars mission may depend on a number of factors, particularly relating to tradeoffs between efficiency and speed in trajectory and transportation system design. In general, a faster transit is more energetic and requires more propellant. However, it could also reduce overall mission endurance and thereby reduce maintenance logistics requirements. The details of transportation system analysis are beyond the scope of this dissertation, but this section characterizes the potential changes in total OGA mass and maintenance crew time requirements that could be achieved by reducing mission endurance.

Figure 7.5 shows the baseline 1,200-day Pareto frontier alongside Pareto frontiers corresponding to 1,100, 1,000, and 900 days. In each case, the required POS is set to 0.995, and the maximum crew time available for maintenance is 5% of the total mission endurance. In general, shorter mission endurances shift the Pareto frontier to lower values in terms of both mass and crew time, as expected – shorter missions require less maintenance resources than longer missions. Using the corner point of the middle section of the Pareto frontier as a reference (i.e. the point located at approximately 39.5 CM-h and 1,602 kg in the baseline case), reducing the baseline mission endurance

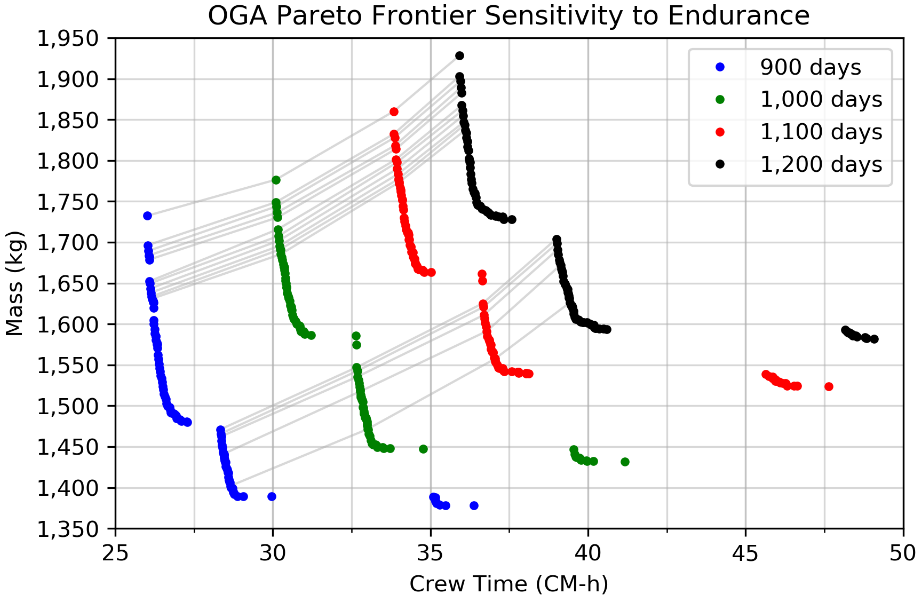

Figure 7.5: Sensitivity of OGA Pareto frontier to changes in mission endurance. The baseline Pareto frontier described in Section 7.2 (1,200 days, POS = 0.995) is shown in black.

by 100 days, or a little over 8%, reduces maintenance and system mass associated with the OGA by approximately 56 kg, to just under 1,550 kg, and maintenance crew time is reduced by approximately 2.5 CM-h. If the mission is shortened by an additional 100 days, to a total endurance of 1,000 days, the total mass associated with that corner point drops to just under 1,454 kg, a reduction of nearly 150 kg from the baseline. Maintenance crew time is also reduced, to approximately 33.1 CM-h. The trend continues for a 900-day mission (which would be particularly quick, for a Mars mission), which places that corner point at approximately 28.6 CM-h and 1,406 kg.

These trends indicate the existence of a breakpoint in mission endurance. This result is also visible in Figure 7.5 by noting that the gap between the frontiers associated with 1,100 (red) and 1,000 (green) days is larger than the one between 1,200 (black) and 1,100 (red) days. This breakpoint, which falls between 1,000 and 1,100 days, is driven by scheduled replacements. The first 100-day reduction from the baseline mission, from 1,200 to 1,100 days, enables mass and crew time reduction largely through reduction in corrective maintenance demand. In both cases, however, the number of scheduled replacements is the same for both life-limited items in this system, and as a result preventative maintenance demands are the same. However, when mission endurance is further reduced to 1,000 days, it falls below a threshold at which the number of replacements is reduced by one for both life-limited items. Coupled with other reductions from corrective maintenance demands, this results in a more significant mass and crew time reduction. Put another way, the second 100-day reduction in mission endurance is more valuable than the first. Further sensitivity analyses, could be used to more specifically identify endurance breakpoints with regard to supportability mass and crew time requirements in order to inform trajectory selection transportation system design alongside other mission considerations.

The shape and structure of the Pareto frontier is similar across all cases, with three distinct sections. However, the specific supportability strategies do change as the endurance shifts. Each Pareto frontier contains 107-109 strategies, but only 16 strategies appear in all four cases. Just under 50% of the Pareto-optimal strategies from the baseline (1,200-day) case are also Pareto-optimal in in the 1,100- and 1,000-day cases, and just under 35% are Pareto-optimal in the 900-day case. Similar trends are present for the other Pareto frontiers. This indicates that changes in endurance do not simply shift the existing Pareto frontier to a new location in objective space, but also impact the strategies that minimize mass and crew time. It is important to note, however, that this analysis only examines the strict Pareto frontier – that is, the set of solutions that are not dominated by any other solution. As with the POS sensitivity analysis described in Section 7.3.1, it is likely that if a fuzzy Pareto frontier were applied then the set of near-optimal solutions would be much more similar across all cases.

7.3.3 Maintainability

As noted in Section 7.1, maintenance crew times for lower-level items are modeling using equation 7.1, which makes use of the maintainability factor a to approximate the crew time cost of performing maintenance at a lower level. The baseline value

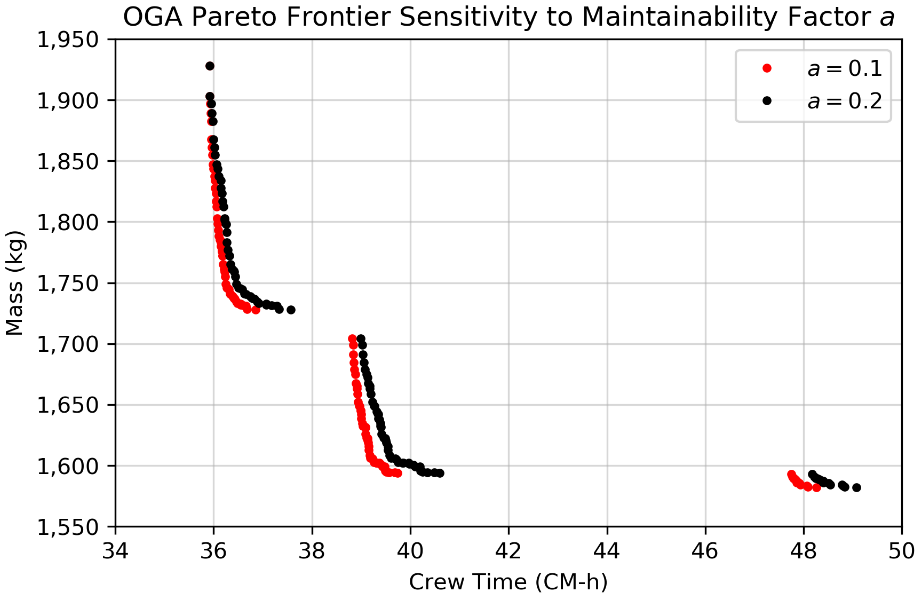

Figure 7.6: Sensitivity of the OGA Pareto frontier to the maintainability factor a. The baseline case ($a = 0.2$) is shown in black, while a notional case with improved maintainability case ($a = 0.1$) is shown in red.

for a is 0.2; this section compares those results to the case where $a = 0.1$ in order to examine the impact of additional investment in maintainability to enable lower-level maintenance without as much increase in crew time.

Figure 7.6 shows the impact of adjusting α to 0.1. As expected, the lower maintainability factor results in a lower crew time penalty for performing maintenance on items below the current ISS ORU level in the indenture structure. Since this factor has no direct impact on spares mass, the net effect of this change is to shift solutions that include lower level maintenance to lower crew times. The shift is larger at lower levels of maintenance, which correspond to lower spares mass, and as a result the maintainability parameter can be thought of as pivoting the Pareto frontier around the upper left corner point, which defines ORUs at the same level in the indenture structure as the current ISS OGA and is therefore unaffected by the maintainability parameter.

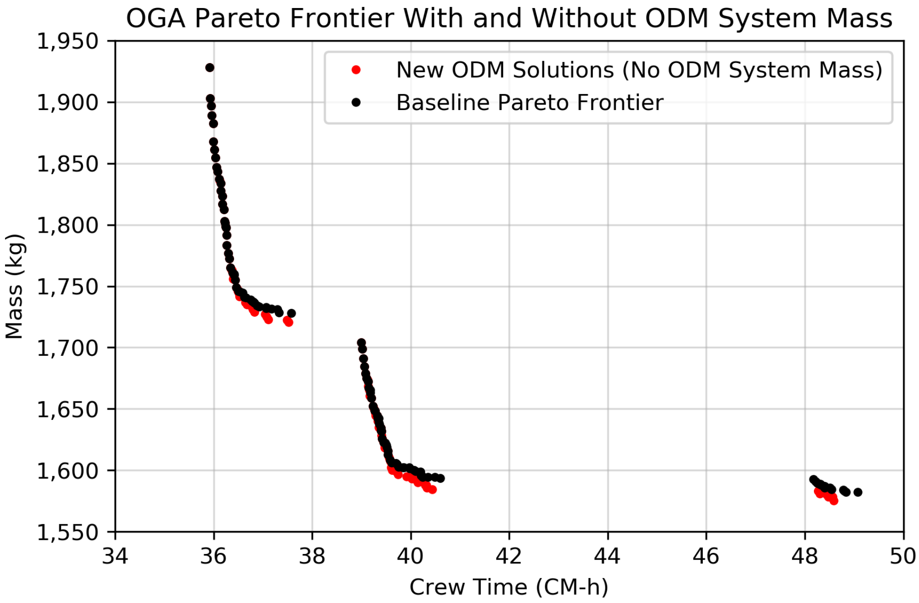

Figure 7.7: Comparison of OGA Pareto frontiers with and without ODM system mass. The baseline Pareto frontier, which does not use ODM, is shown in black. Red points indicate strategies that use ODM that are Pareto optimal when ODM system mass is set to 0.

7.3.4 ODM System Mass

The final sensitivity analysis examines the impact of ODM system mass. None of the Pareto-optimal strategies in the baseline case utilized ODM, which is not unexpected given that only a small proportion of the items examined in this case study were considered candidates for ODM and the benefits of ODM increase as more items utilize it. This section repeats the analysis with ODM system mass m_{odm} set to 0, and examines the gap between the baseline Pareto frontier and the new one in order to determine the allowable ODM system mass for this application. It is important to note that this ODM system mass only relates to OGA maintenance demands – if other systems also make use of ODM, then the resulting maintenance logistics benefits would allow for the use of larger ODM systems.

Figure 7.7 shows the shift in the Pareto frontier when ODM system mass is removed. Points in black show the original, baseline Pareto frontier described in Section 7.2, which does not include any strategies that use ODM. The red points indicate strategies that use ODM that appear on the Pareto frontier when ODM system mass m_{odm} is set to 0. As expected, these new solutions tend to occur in regions of the Pareto frontier that correspond to lower-level maintenance, since manufacturable items tend to appear lower in the indenture structure. The largest gaps between the baseline Pareto frontier and the one in which ODM system mass is removed are approximately 10 kg wide. This indicates that an ODM system of less than 10 kg would be required to provide an optimal solution if that system were used only for manufacturing spares for the OGA. Put another way, the mass margin for an ODM system in this case is approximately 10 kg. This result is not surprising, since only five of the items included in this case study were considered to be candidates for ODM, and this case study only examines one system. As noted earlier, the impacts of ODM increase as more systems use it, and it is expected that broader manufacturability within the system, as well as broader analysis considering more systems with potentially manufacturable items, would increase the mass margin for the ODM system.

The analysis approach used in this section – that is, assuming an ODM system mass of 0 and assessing the gap between the Pareto frontier formed by non-ODM strategies only and the frontier formed by the entire set – is particularly valuable for the assessment of the potential value of ODM, since the characteristics of systems that will be required to manufacture useful components in space are uncertain and continually changing as technology advances. By assessing this gap, an upper bound on the allowable manufacturing system mass can be derived, and trades between manufacturing system mass and the mass and crew time associated with the point at which ODM becomes an optimal strategy can be evaluated directly, rather than through iterated modeling and optimization using different parameters.

7.4 Summary

This chapter has demonstrated the application of the supportability analysis methodology developed in this research to a real-world system, namely the ISS

OGA. Opportunities for lower-level maintenance and other supportability strategy adjustments were defined based on descriptions of current ISS systems. The resulting tradespace of options is orders of magnitude larger than the case examined in Chapter 6, meaning that full enumeration and evaluation would require a prohibitively long time. However, using the multiobjective branch and bound approach described in Chapter 4, MCROSS is able to solve for the Pareto frontier in less than an hour, enabling the evaluation of various sensitivity analyses.

Overall, these results demonstrate how MCROSS can be used to inform system and mission development by characterizing the mass and crew time tradeoffs for a given concept, as well as how those tradeoffs – and the underlying supportability strategies – shift in response to changes in other parameters, such as POS and mission endurance. In the baseline case, strategies were identified that offered significant mass and crew time reduction relative to the current system configuration, under the assumptions of this case study. Over the course of this baseline and several sensitivity analyses, MCROSS was used to examine hundreds of thousands of automatically generated supportability strategies, identify Pareto-optimal strategies, and characterize tradeoffs between mass and crew time under a variety of assumptions. Iterative analysis provides a powerful tool to integrate supportability considerations into system and mission development alongside other considerations.

The ability to examine hundreds of thousands of options and identify optimal solutions is enabled by the Constraint Optimization Problem (COP) formulation developed as part of this research, described in Chapters 3 and 4. Previous supportability analysis models are focused on the assessment of individual strategies and ORU lists, with some capability to include redundancy and specific, pre-defined lower levels of maintenance. As a result, they do not offer multiobjective optimization over a range of supportability strategies. A comparable analysis to this one, conducted using existing models, would require extensive and significant effort to define analysis cases, and would likely be prohibitively time-consuming without a formal encoding and variable interpretation scheme such as the one used by MCROSS. In contrast, the methodology presented in this dissertation presents a means to perform all required analysis in a matter of hours. This accelerated turnaround time for analysis results enables a responsive, even interactive analysis capability whereby decision makers, system experts, and supportability analysts can come together to explore the supportability characteristics of the system, understand how they interact with other decisions, and identify strategies to achieve mission objectives in a risk-informed, cost-effective manner.

THIS PAGE INTENTIONALLY LEFT BLANK

Chapter 8

Discussion

This chapter discusses the methodology presented above. Section 8.1 presents and discusses the contributions of this research, including methodological contributions as well as domain contributions. Section 8.2 lists and discusses key assumptions and limitations, and Section 8.3 describes how the methodology developed as part of this research can be applied – and in several cases, has already been applied – to inform space mission planning, system design, and technology investment.

8.1 Contributions

The contributions of this research can be broadly divided into two categories. The first category consists of methodological contributions, which are advancements in the state-of-the-art in spacecraft supportability analysis. These are the primary contributions of this research, and represent the characteristics of the methodology described in Chapters 3 and 4 that address the research gap described in Chapter 2. This category is further subdivided into contributions related to supportability strategy optimization and those related to supportability modeling. The second category consists of domain contributions, which represent advances in the state of knowledge in the field of space system supportability. These are derived largely from the analyses performed as part of model validation in Chapter 5 and the OGA case study in Chapter 7.

8.1.1 Methodological Contributions

The background presented in Chapter 1 motivated the need for analysis capabilities that could identify and characterize Pareto-optimal supportability strategies that minimize mass and crew time subject to constraints on POS, and the literature review presented in Chapter 2 showed that no existing supportability analysis methodologies provided the required capabilities. The core contribution of this dissertation is the definition and modeling of the supportability strategy optimization problem as a multiobjective Constraint Optimization Problem (COP), which enables the application of formal optimization techniques to identify Pareto-optimal supportability strategies. This optimization capability is supported by other contributions that are specific to the modeling of mass, crew time, and risk for a given supportability strategy. All of these contributions are described in the following sections.

Supportability Strategy Optimization

Chapters 3 and 4 present a new method that enables multiobjective optimization of a wide range of supportability strategies, a capability that was not found in previous models. This contribution consists of a formal definition of and model for the supportability strategy optimization problem, which is structured as a multiobjective COP. The problem formulation includes a generalized system description that captures all information required for supportability analysis (mass, failure rate, etc.) while providing the flexibility required to examine a range of potential supportability strategies. The decision variables defined for this problem encode decisions related to five supportability strategy elements (level of maintenance, ODM, commonality, redundancy, and distributed functionality), and associated constraints define feasible strategies. A method is presented for interpret an assignment to those decision variables, along with a system description, in order to define an ORU list, which is the core starting point for existing system supportability analyses.

Some existing supportability models provide solutions to portions of the supportability strategy problem, but none of the models found during the literature review enable multiobjective optimization over the range of strategies examined here. For the most part, existing models focus on analysis of fixed supportability strategies, rather than optimization. Where all other supportability analysis approaches start with an ORU list (or something very similar to it), the methodology developed in this research provides a way to algorithmically define an ORU list based on a general system description and an assignment to a set of supportability strategy decision variables. This strategy encoding scheme and method for interpreting a given strategy and defining ORUs are novel contributions that enable supportability strategy optimization.

One area of particular interest is the level of maintenance problem – that is, decisions related to the level in the indenture structure at which maintenance actions are implemented. These decisions determine how components are aggregated into ORUs, and are a key driver of mass and crew time requirements. The two models in the literature that come closest to implementing this capability are METRIC and SSM™, but neither fully addresses the problem. Specifically, METRIC examines the multi-indenture problem, which means that LRU and SRU are defined as part of the system description and maintenance can take place at either level, depending on maintenance location and parts availability. This multi-indenture problem is different from the level of maintenance problem, since the analyst must pre-define the specific level of maintenance options (i.e. LRUs and SRU). In contrast, the methodology developed in this research uses the system hierarcy, component characteristics, and a set of decision variables to generate ORUs with much greater flexibility. Any potential aggregation of lower-level items into higher-level ORUs is allowed in MCROSS (subject to constraints based on the system indenture structure and other supportability strategy decisions), while METRIC can only explore pre-defined options. In short, METRIC only optimizes supply-chain characteristics and inventory, while MCROSS optimizes the definition of ORUs within the system itself.

SSM™ does include the ability to examine changes in level of maintenance, but it does so by using a parametric scaling factor to resize ORUs and obtain new mass and crew time estimates. This is a rough approximation of the impacts of changes in level of maintenance that does not capture the interactions between maintaining items at a lower level in the indenture structure and having more distinct items to account for in spares allocation planning. In addition, it does not account for changes in failure rates resulting from different combinations of components into higher-level assemblies. In contrast, MCROSS has the ability to use the indenture structure to combine items into arbitrary ORUs, directly evaluating the impacts of lower-level maintenance by creating and evaluating different ORU lists rather than approximating these impacts using parametric factors.

Redundancy is another area in which previous models include supportability strategy optimization capabilities. RAP approaches have been used to optimize redundancy alongside spares allocations in order to achieve system availability targets, but these models assume that redundancy is applied at the same level of maintenance as spares – that is, redundant items are copies of the same ORUs that are removed and replaced by maintenance actions. Other models, such as approaches used by the ISS program, SSM™. and EMAT, include redundancy as a parameter that can be adjusted, but do not optimize redundancy decisions. When redundancy is implemented at the same level as spares, its impact is primarily to allow continued system operation without requiring maintenance. Effectively, a redundant ORU is simply a pre-installed spare that can activate without requiring a crew maintenance action, and has no impact from a mass perspective. Redundancy above the ORU level (i.e. at the system or subsystem level) can similarly be thought of as providing one additional pre-installed spare for each ORU in that system. While this can increase the amount of time that the crew has to respond to a failure, and can reduce crew time demands, it is typically not an efficient approach from a mass perspective. In contrast, MCROSS examines and optimizes redundancy at the component level. This sub-ORU redundancy can reduce mass and crew time requirements by adding a redundant instance for a specific component within an ORU without adding an additional spare for the entire ORU. Effectively, it allows targeted component redundancy to adjust the failure rate, crew action rate, life limit, and mass characteristics of an ORU, rather than simply changing the number of pre-installed instances of that ORU in the system. As a result, it explores different opportunities to apply redundancy to reduce mass and crew time.

Other supportability strategy decisions – ODM, commonality, and distributed functionality – are either not included in existing supportability models, or are included but not optimized.

Another key element of the supportability strategy optimization problem formulation presented in this dissertation is the method for identifying the lower bounds on mass and crew time for a partial assignment to the decision variables, described in Section 4.3.2. These lower bounds allow for pruning of search paths during the multiobjective branch and bound algorithm used to solve the COP, and are essential for optimization. Comparison of Pareto frontiers found via full enumeration and via optimization in the notional case study presented in Chapter 6 showed that the optimization approach successfully finds all of the strategies in the Pareto frontier. These results indicate that the bounding techniques described in Section 4.3.2 are correct. Future work will continue to refine these bounds to improve optimization efficiency.

Supportability Modeling

This dissertation presents several contributions in terms of supportability modeling in addition to the overarching supportability strategy optimization approach discussed above. These modeling contributions are critical to the overall supportability strategy optimization framework, since they are key elements of the model used to evaluate mass and crew time given an assignment to the decision variables. They are also contributions in and of themselves, since they advance the state of the art in the modeling of space mission supportability mass and crew time requirements.

The first of these contributions is the separation of the stochastic processes describing crew time demands and spares demands. All previous supportability models found in the literature use failure rates to model demand for both spares and corrective maintenance crew time. In doing so, they only track crew time associated with R&R of failed ORUs. However, ISS experience shows that a significant amount of crew time is spent on maintenance activities that do not involve R&R of a failed ORU, and therefore are not captured in the ORU failure rate. Specifically, the validation dataset examined in Chapter 5 show that nearly half of required corrective maintenance events for the WPA, UPA, and OGA between November 2008 and November 2016 were instances in which the ORU in question was not replaced. These non-R&R events accounted for 26% of corrective maintenance crew time for these systems during that time period. Those non-R&R maintenance demands are not captured in the ORU failure rates, and as a result analysis that uses failure rates alone to assess crew time would underestimate crew time requirements. If the failure rates were adjusted to account for these non-R&R events, then spares demands would be overestimated. MCROSS solves this issue by introducing the crew action rate Λ_C for each ORU, which is analogous to but separate from the failure rate Λ. By encoding the demands for mass and crew time in tow separate parameters, the model improves crew time estimates.

Another contribution is the inclusion of the probability of sufficient crew time in overall POS assessment, as well as its impact on the spares allocations required to achieve system POS targets. Other models optimize spares allocations to achieve a given POS requirement, but these models do so independently of any crew time considerations. POS is used to refer exclusively to the probability of sufficient spares. Some models (e.g. EMAT) do calculate distributions of the amount of crew time required, and can therefore calculate the probability of sufficient crew time for a given mission. However, these calculations are typically used to evaluate the crew time associated with a particular percentile of the distribution as a crew time metric, rather than to inform overall risk assessment and spares allocation. In contrast, MCROSS defines POS as the probability that sufficient maintenance resources are provided, regardless of whether those are physical resources like spares or temporal resources like crew time. As a result, the POS value achieved when spares allocations are optimized takes into account the probability of sufficient crew time, and is the overall probability of having sufficient resources to perform required maintenance rather than simply the probability of sufficient spares. If less crew time is available, the probability of insufficient crew time will be higher, and the probability of insufficient spares will need to be correspondingly lower in order to achieve the same overall level of risk. As a result, the amount of spares required to achieve the same level of risk will be higher. By including crew time as a resource in POS considerations, the model presented here provides a more complete accounting of sources of risk.

One of the primary modeling contributions of this research is a method to evaluate the relationship between ODM feedstock mass and POS. ODM approaches to maintenance logistics management, enabled by new ISM technology, has the potential to revolutionize space exploration logistics, providing entirely new ways to supply required parts for space systems and advancing closer to the possibility of self-sufficient spacecraft, outposts, and colonies. ISRU enables the production of useful consumables and raw materials, but without ISM any space operations will be dependent on Earth for manufactured goods such as spare parts and maintenance items. The potential impacts of ODM on risk and logistics must be quantified in order to evaluate the potential benefits of this emerging capability, guide technology investment, and inform system development and mission planning for future applications. Previous models for ODM have focused exclusively on its potential to enable material recycling. While this capability is important, it does not capture the significant impact on maintenance logistics mass that ODM provides simply by enabling commonality of material between different items. The model developed as part of this dissertation (initial versions of which were originally presented at International Conference on Environmental Systems (ICES) [24] and AIAA SPACE [25]) is the first model that captures this impact and allows ODM feedstock mass to be allocated as a maintenance resource alongside traditional spare parts. This ODM analysis capability is valuable in several ways. First, and most directly, it allows for the inclusion of ODM in supportability analyses to inform mission planning and system development. It also enables characterization of the potential value of various ISM technologies by exploring the impacts that those capabilities could have on future missions through a series of case studies and sensitivity analyses. The logistics benefits of a given ISM system, characterized by this ODM model, can be used to inform system development by, for example, helping to bound allowable manufacturing system mass. This type of analysis can identify breakpoints and characterize the breakeven curve relating manufacturing system cost and capability.

The fourth modeling contribution of this methodology is the fact that all required calculations – including evaluation of distributions of the amount of shared resources such as crew time and ODM feedstock and the POS associated with particular amounts of those resources – are performed without any Monte Carlo simulation. Instead, these distributions are evaluated to a relatively high precision, even at extreme values (e.g. POS in excess of 0.999), using a numerical characteristic function inversion approach based on the technique described by Warr [130]. This approach is much less computationally expensive than Monte Carlo simulation of comparable precision, and as a result enables the full analysis of a given supportability strategy in less than a tenth of a second. Rapid evaluation is critical for supportability strategy optimization, since the tradespace of potential solutions is typically very large and COP solution algorithms may need to evaluate tens of thousands of candidate solutions in order to find the Pareto frontier. While there are other models that evaluate supportability characteristics without using Monte Carlo simulation, these models do not assess the risks associated with crew time and do not assess the distribution of ODM feedstock required. As a result, the are limited to the use of expected crew time as a metric. They do not use percentiles of the crew time distribution as an objective function, and they cannot incorporate the probability of sufficient crew time into overall POS calculations. In addition, they cannot assess ODM feedstock mass allocations, and therefore cannot fully include ODM as a supportability strategy. Models that do evaluate crew time distributions, such as EMAT, do so using Monte Carlo analysis. While the few seconds that may be necessary to generate results (depending on the desired precision) may be more than acceptable for the evaluation of a single system and supportability strategy, the amount of computational time required to evaluate many thousands of strategies during optimization would likely be prohibitive.

8.1.2 Domain Contributions

In addition to the methodological contributions described above, this research also provides several domain-specific contributions. First, the examination of supportability data and maintenance histories presented during model validation in Chapter 5 represents the first known in-depth statistical assessment of maintenance crew time, failure rate, and crew action rate parameters for individual ORUs. This assessment provided two key results that are important for supportability assessment. The first is the comparison between CMMTTR estimates and observed maintenance crew time requirements per action, shown in Figure 5.15 on page 249. These results showed that the amount of crew time required to perform maintenance on-orbit was typically higher – sometimes by a large amount – than predicted values. Underestimated maintenance crew time per action has a significant effect on overall maintenance crew time estimates, as shown in Figures 5.16 and 5.17.

The validation analysis also performed a statistical comparison between ORU failure rate and crew action rate estimates, developed via Bayesian updating using the first half of the data set, and the observed number of failures in the second half of the data set. The results of this assessment, shown in Figure 5.14 indicate that, while most estimates were in statistical agreement with the observed number of failures or crew actions, in several cases the difference between predicted and observed values was statistically significant. Unfortunately, in most of the cases where there was disagreement between the rate estimate and the observed number of events, the rate underestimated the number of failures or crew actions. As a result, models using those rate estimates tend to underestimate the amount of spares required for those ORUs, leading to lower-than-expected POS and higher risk (see Figure 5.21).

These results highlight the challenge of performing supportability analysis in the relatively data-poor context of space systems, as well as the impacts that invalid data can have on analysis results. These effects are discussed further in Section 8.2.3 below.

The OGA case study presented in Chapter 7 also contributes to domain knowledge by identifying and characterizing the Pareto frontier of supportability strategies minimizing mass and crew time for the OGA. Under the assumptions used in that case study, the current OGA supportability strategy does not appear on the Pareto frontier. This indicates that there may be significant opportunities to reduce maintenance-related mass and crew time for the OGA. Previous analysis by Takada et al. [105] examined the potential value of specific strategies for reducing mass for the OGA, but candidate strategies were manually generated based on expert opinion, and resulted in point solutions showing examples of potential improvements. In contrast, the results of the case study presented in Chapter 7 provide a range of options along the Pareto frontier that improve upon the current strategy, and characterizes the underlying strategy decisions that lead to those improvements. These results provide much richer information for decision-makers, allowing for informed trades of mass and crew time.

In addition, the sensitivity analysis performed in the OGA demonstrated the ability to iteratively identify and characterize Pareto-optimal solutions under a wide range of different conditions. These sensitivity analyses characterized the impacts of changes in POS, endurance, maintainability, and ODM system mass on mass and crew time requirements, as well as in the underlying supportability strategy. When integrated into an overarching mission analysis process alongside other models for other portions of the mission (e.g. transportation and habitation systems), this type of iterative analysis can help identify breakpoints and help enable risk-informed system development.

8.2 Assumptions and Limitations

This section describes the assumptions and limitations of the methodology described in this dissertation. Section 8.2.1 describes modeling assumptions, while Section 8.2.2 describes factors that are beyond the scope of the current model. Finally, Section 8.2.3 discusses limitations related to data availability.

8.2.1 Key Assumptions

One driving assumption in this model is that failure rates are constant over time. This assumption enables the use of the CFR model for failure rates, which states that the number of random failures that occur in a given time period follows a Poisson distribution; when adapted to use gamma-distributed failure rates, the result is a negative binomial distribution. The CFR assumption is a common one for mission supportability assessment, particularly for early-phase analyses. However, some systems may exhibit failure rates that increase over time as equipment ages, or they may exhibit increased failure rates during an early "burn-in" period. Neither of these effects are included in the stochastic failure model, which is only designed to capture purely random failures. However, a deterministic life limit is included, which can be used to represent component wear-out. Overall, the use of the CFR model likely means that there are some drivers of component failure that are not captured in the model, and that actual risk is higher than calculated here.

Crew time assessment also applies a version of the CFR model, using crew action rates instead of failure rates. In addition, the model currently assumes that all preventative maintenance actions involve R&R. That is, there is no regularly-scheduled recurring maintenance crew time demand that does not involve the use of an ORU. Finally, the amount of crew time required per event is assumed to be constant which, for the analyses presented here, is either the average of the amount of time required for observed failures or the estimated CMMTTR. In reality, the amount of crew time required for a particular maintenance action varies from action to action, and therefore maintenance crew time could be represented as a random variable. There are also likely to be regularly scheduled maintenance actions like inspection or cleaning that would be more appropriately modeled as a deterministic demand (similar to life-limited items) than a portion of a stochastic process. When assessing expected crew time, the use of random variables or expected values to model the amount of crew time required per event is not expected to have a significant impact. However, the evaluation of crew time POS would likely be impacted by the incorporation of additional uncertainty regarding the amount of crew time required per event. Specifically, the additional level of uncertainty likely increases risk for longer missions, and therefore crew time POS may be underestimated by this model. Future work will seek to incorporate uncertain crew time distributions, though there are significant limitations in the data available to define these distributions.

All component maintenance demands are assumed to be independent for the purposes of crew time analysis. That is, a failure in one component does not cause failures in any other components. This is a common assumption in mission supportability analyses, and is due in large part to the fact that this model, like most others, does not simulate day-to-day spacecraft operations and maintenance demands. Instead, it applies ORU-level models (described in Section 4.2) to assess the POS for each item individually, then combines that information to obtain system-level metrics. This abstracted modeling process allows for much faster evaluation of system supportability than other approaches, but it does sacrifice some model fidelity. However, interactions between systems can be modeled using K-factors, which are designed to account for induced failures. If certain combinations of systems are known to induce higher failure rates in other systems, then the K-factors associated with those systems could be adjusted to reflect that knowledge. This is a parametric adjustment, not an exact model of interactions between ORUs, but is appropriate during early-phase modeling to support system development and mission planning. As system designs mature and more information is available regarding interactions between different elements, more detailed models that simulate day-to-day operations can be used to improve mass, crew time, and risk estimates.

Within this model, the impact of component redundancy is captured by dividing failure rates by the number of instances of that component. However, as noted in Section 4.1.1, this is an approximation used to enable the incorporation of redundancy into the flexible supportability strategy modeling framework. When standby redundancy is used, the failure rate of a component is no longer constant over time. Taking the deterministic failure rate case as an example, the time to failure for a component without redundancy follows an exponential distribution, which is the CFR model. When redundancy is present, however, the time to failure follows an Erlang distribution, which is the sum of independent exponential random variables. Section Therefore, standby redundancy violates the CFR assumption used to model POS, and the use of that model with the expected value / variance adjustment described above is an approximation. Section 5.1.2 examined the error introduced by this approximation (see Figure 5.9), and found that it was the largest contributor to modeling error among the factors examined as part model verification. The error introduced by this approximation can be high at mid to low POS values (i.e. POS values below 0.9), but the amount of error rapidly falls off as higher POS values are examined. Since the high end of the POS range is typically the region of interest for supportability analyses, the current model is likely sufficient for most applications. However, potential improvements in the way redundancy is modeled could expand the useful range of the supportability evaluation model described in Chapter 4.

8.2.2 Limitations

The model presented in this dissertation focuses on the assessment of mass and crew time as a function of an acceptable level of risk, characterized by POS. It does not include an assessment of the amount of system downtime that may occur, or of the risk that excessive downtime may lead to unrecoverable system failure. Nor does it include an assessment of the risk that a cluster of failures could overwhelm the crew's abilities to implement repairs – that is, that the total amount of crew time may be sufficient across the entire mission, but not for some subset of the mission. As a result, the risk assessments provided here should be considered partial, accounting only for the risk of insufficient crew time or spare parts at level of the entire mission.

In addition, as a result of the selected metrics, the only type of redundancy included in the model is redundancy at the component level. This is due to the fact that, as discussed in Section 3.2.1, this type of redundancy is the one which has the largest impact on mass and crew time requirements. while redundancy could be implemented at the ORU level or higher, the primary benefits of this approach would be to reduce system downtime or risks resulting from downtime, not to reduce mass or crew time requirements. From a mass perspective, higher-level redundancy is be effectively equivalent to the addition of one spare for each ORU in the redundant system, a strategy which is always dominated by more targeted ORU-level sparing for the same reasons that lower-level maintenance provides lower spares mass. In terms of crew time, redundancy above the ORU level does not decrease demand because failed ORUs still need to be removed and replaced. The only impact is that the crew may have more time to do so while backup systems maintain system functionality.

The current model assumes a homogeneous mission profile, meaning that there are no changes in demand models for different phases of mission operations. Failure rates are constant, life limits are constant, and the amount of available crew time is specified across the entire mission, not on a day-to-day basis. However, this limitation can be circumvented by representing a longer, heterogeneous mission as a series of homogeneous mission phases, each of which is assessed independently. The results of spares and crew time analyses for each of those mission phases could be combined into a single, overall mission characterization, using a similar approach to that used by Do et al. [147] and Do et al. [148].

8.2.3 Data Limitations

The results of the validation conducted in Section 5.2 highlight a critical aspect of supportability modeling: models are only as accurate as their input data. In the case of crew time modeling, CMMTTR estimates were significantly lower than the average observed crew time required per maintenance action in several cases, and as a result crew time estimates produced by MCROSS and EMAT/ECTM significantly underestimated total crew time. For corrective maintenance models, the presence of underestimated failure rates means that supportability models can significantly underestimate risk. When valid inputs are used – in this case, when CMMTTR estimates are replaced by average observed crew time, or when ORUs with underestimated failure rates are removed from the analysis – the model produces valid results. However, it is difficult if not impossible to know *a priori* which ORUs or components have parameter estimates that are overestimated or underestimated. As a result, analysts must take great care to ensure that the input data used in their assessments are as accurate as possible, and implement data collection and validation procedures. Analysis and consumers of analysis results must also bear in mind the fact that the presence of invalid data can significantly degrade the quality of model outputs, and that it is critical to validate models as well as their inputs.

This does not mean that the results of these analyses cannot inform system design. Comparisons between different options using of best-available data can provide key insights that help inform system design and mission development, even if the data are not perfect representations of the underlying system. The dataset will never be perfect, and nor will the model – however, the results can still be useful. In many cases, sensitivity analysis may be more valuable than a single assessment. Examination of a variety of options, and comparison of their characteristics, can guide decisions even if the absolute outputs of the analysis are not necessarily accurate, as long as the relative impacts of different effects are captured. These sensitivity analyses can even be used to guide information-gathering activities such as test campaigns or in-depth reliability analyses of particular ORUs and/or components. For example, the sensitivity of system-level results (e.g. mass, risk, or crew time) to specific parameters (e.g. failure rates or crew action rates) can identify items that have the greatest impact. This type of analysis could identify cases where an underestimated failure rate would be the most costly, and guide investment in testing activities to reduce the risk that the failure rate is underestimated.

Overall, it is important to think of supportability analysis not as a one-time assessment to be performed once the system is fully designed, but rather as an iterative process that can inform system development and mission planning at every step. Supportability analysis is a tool for forecasting future lifecycle properties of a system and their impacts on operations. These forecasts should always be based on the best available data, should always recognize the limitations of the data they are based on, and should be updated as new data become available. One key contribution of this research (discussed in Section 8.1 above) is the fact that all required analysis is conducted without the use of Monte Carlo simulation. MCROSS is designed to enable quick assessment of the supportability tradespace for a given system, because it is designed to support iterative analysis as a decision-support tool throughout the design process.

8.3 Applications

The supportability analysis capabilities developed in this dissertation have a range of potential applications, some of which were demonstrated in the case studies presented in Chapters 6 and 7. MCROSS can be used in a static manner to characterize the tradeoffs between mass and crew time for a given system and mission and identify the set of Pareto-optimal supportability strategies, but it may be more valuable when deployed to examine and characterize sensitivities to key parameters or changes in system design. This includes mission parameters such as endurance or POS requirements, as well as changes in the system itself, such as improved failure rates, reduced mass, or different designs. Sensitivity analysis examining different levels of technological capability, such as changes in the set of maintainable items as a function of the capability of a proposed ISM system, can be used to evaluate the potential benefits of that technology and inform investment.

The supportability analysis techniques developed and described in this dissertation have already informed NASA decision-making on a range of issues, including:

- ISM technology development: The ODM model developed for this research was used to help identify and characterize the potential value of the ability to manufacture spares on demand for future exploration missions in collaboration with the ISM Project Team at NASA Marshall Space Flight Center (MSFC) [25]. The information generated from that analysis was presented to MSFC senior leadership, including the Center Director, in order to inform investments in ISM technology development as well as efforts to adapt existing system designs to enable ODM of components in high-value application areas.

- System redesign to facilitate lower-level maintenance: While it did not involve the level of maintenance optimization described in this dissertation, and used EMAT for spares analysis instead of MCROSS, supportability analysis such as the ones described in this dissertation were used to evaluate the potential logistics benefits of a range of proposed changes to the ISS OGA in collaboration with Advanced Exploration Systems (AES) engineers at NASA Johnson Space Center (JSC) and MSFC [105]. Supportability analyses were able to identify the mass savings resulting from lower-level maintenance on the Hydrogen ORU, as well as identify key breakpoints in reliability growth for particular components that were used to inform the selection of high-value targets for reliability growth efforts.

- Test planning and use of the ISS as a system testbed: The models described in this dissertation were used to characterize the value that the ISS has provided as a platform for system testing, in terms of improved failure rate estimates and resulting reductions in both logistics mass and risk [40]. The results of that analysis, which were discussed with the Associate Administrator for Human Exploration and Operations (HEO), highlighted the importance of testing and ISS operations for system maturation and are helping to inform test campaign planning for future system development.

Overall, the supportability analysis methodology described in this dissertation represents a powerful tool for understanding the relationship between mass, crew time, and risk for future exploration systems. When integrated with other analysis capabilities examining other elements of mission design, such as ECLSS resource demands or trajectory analysis, supportability analysis helps provide a clearer picture of system cost, performance, and risk, and helps inform decisions to develop effective systems.

THIS PAGE INTENTIONALLY LEFT BLANK

Chapter 9

Conclusions

This dissertation presents a methodology for modeling and optimization of supportability strategies for crewed spacecraft. The supportability strategy optimization problem, defined in Chapter 3, formulates this challenge as a multiobjective Constraint Optimization Problem (COP) seeking to minimize mass and crew time required for maintenance, subject to a constraint on required POS. This problem formulation includes the definition of a set of decision variables that can be used to encode a variety of supportability strategy decisions, including decisions related to level of maintenance, ODM, commonality, redundancy, and distributed functionality. Chapter 4 then defines a model for evaluating the supportability characteristics of a given assignment to those decision variables, or for evaluating the bounds on objective values. Chapter 5 presents a verification and validation of this analysis methodology, which is implemented in a tool called MCROSS. Chapters 6 and 7 presented two case studies demonstrating MCROSS's capabilities and how it could be used to inform system development, and Chapter 8 discusses the implications of this modeling and optimization capability, as well as key assumptions and limitations. This chapter summarizes the contributions of this research and presents future work and conclusions.

9.1 Summary of Contributions

The key contributions of this research, discussed in greater detail in Section 8.1, fall into two broad categories: methodological contributions, which consist of advancements in the state-of-the-art in analysis capabilities, and domain contributions, which consist of advancements in the state of knowledge for space mission analysis. Methodological contributions are summarized below.

- A method for multiobjective optimization of spacecraft supportability strategies, minimizing mass and maintenance crew time subject to a constraint on risk, which consists of:
  - A formal definition and model of the supportability strategy optimization problem using a set of decision variables that can encode a range of supportability strategy elements, constraints that define feasible assignments to those variables, inputs and parameters describing a system and mission, and objective values.
  - A method for interpreting full assignments to the supportability strategy decision variables in order to evaluate the mass and maintenance crew time associated with a given strategy for a particular system and mission.
  - A method for calculating the lower bounds on mass and maintenance crew time associated with a partial assignment to the decision variables for a given system and mission.
- A supportability model that advances the current state of the art in the following ways:
  - Improved maintenance crew time estimates via inclusion of non-R&R events and separation of stochastic processes and parameters describing crew time and spares demands.
  - Improved risk assessment via inclusion of the probability of sufficient crew time in overall POS calculations and spares allocation optimization.

  - A ODM model that enables the evaluation of the POS associated with a given amount of ODM feedstock, which enables the assessment of ODM as a supportability strategy and optimization of feedstock allocations alongside traditional spares.
  - Evaluation of crew time and ODM feedstock distributions without the use of Monte Carlo simulations, which enables fast supportability assessment and facilitates large-scale optimization and tradespace exploration.

The domain contributions, which primarily arise from the validation analysis described in Chapter 5 and case study described in Chapter 7, are:

- Comparison of CMMTTR estimates to observed maintenance crew time requirements, indicating that on-orbit maintenance times are typically significantly longer than predicted.

- Statistical evaluation of failure rate and crew action rate estimates generated via Bayesian updating in comparison to observed failures. This analysis indicated that some rate estimates were not in statistical agreement with observed behavior, and in several cases the observed number of failures was significantly higher than would be predicted, leading to underestimated crew time requirements and overestimated POS (i.e. underestimated risk). This highlights the challenge of obtaining good data for space supportability analysis, and the risks involved with underestimated rates.

- Characterization of the Pareto frontier of supportability strategies minimizing mass and crew time for the OGA (under the assumptions of the case study described in Chapter 7), indicating the tradeoffs between these two metrics as well as the underlying optimal strategies.

- Sensitivity analyses showing the impact of changes in POS, endurance, maintainability, and ODM system mass on the set of Pareto-optimal strategies for the OGA.

9.2 Future Work

One area of future work is improvement of the mass and crew time bounds calculations described in Section 4.3.2. The tightness of these bounds is a key driver of optimization efficiency, since they influence the amount of pruning that occurs during search. However, if the bounds are incorrect, they may inadvertently prune a region of the tradespace that contains optimal solutions. The results of the comparison of full enumeration to optimization in Chapter 6 show that all Pareto optimal solutions are correctly identified during optimization, indicating that the current bounds are admissible and do not cause inadvertent pruning. Future work will seek to refine these bounds to make them tighter, helping to accelerate the optimization process, while still ensuring that they are correct.

The model presented here examines mass, crew time, and the risk that the amount of mass or crew time provided will not be sufficient. However, as noted in Section 8.2, another potential source of risk is system downtime – that is, there is a risk that there may be sufficient crew time to conduct maintenance, but that maintenance may not occur quickly enough. System downtime may itself be another metric to include in the model, alongside risk of failure due to downtime. Incorporation of these metrics into the model would expand its ability to examine other supportability strategies, such as redundancy at the ORU level or above.

In addition, the approximation used to model redundancy was identified as the largest source of error during model validation in Section 5.2. Though this error primarily exists in a low-POS region that is typically not of interest during supportability analysis, an improved model could expand the valid range of assessments incorporating redundancy. Future efforts can identify more accurate redundancy modeling approaches and incorporate them into the overall MCROSS framework in order to enable the use of this model to examine a wider range of POS values.

As noted in Section 8.2.1, this model used deterministic crew time values – that is, the amount of crew time required per maintenance event for a given ORU was assumed to be the same each time. Future work will seek to expand this model to incorporate uncertain maintenance crew times in order to more accurately capture risk associated with crew time. However, current crew time data is very limited, and deals primarily with the level of maintenance currently implemented on the ISS. Additional data must be gathered to assess crew time requirements at various levels of maintenance even in a deterministic sense. A more significant effort would be required to assess maintenance crew time per event as a random variable, and the effort (and test time) required to collect these data will need to be weighed against the potential benefits that it could provide.

The supportability strategies examined in this research are just a few of the many decisions that need to be made in order to formulate and execute exploration missions. The supportability decisions described here are coupled with each other; they will almost certainly be coupled with other decisions related to transportation architecture, crew size, or even procurement and testing strategies. As a result, one of the key areas of future work for this research will be the incorporation of interfaces to facilitate the use of MCROSS alongside other space mission modeling tools, such as HabNet [42, 147] and SpaceNet [149–151], in order to facilitate holistic space mission analysis and optimization.

9.3 Conclusions

This dissertation presents a new methodology for the modeling and optimization of supportability strategies, called MCROSS. This methodology encodes the supportability strategy optimization problem as a multiobjective COP, defining a set of decision variables that capture a wide range of supportability strategies and a model for assessing the mass and crew time associated with those strategies (or partial assignments to the strategy decision variables) as a function of a system and mission description, including the definition of an acceptable level of risk. The associated supportability evaluation model contains several novel contributions that improve assessment of crew time and risk, enable evaluation of ODM, and facilitate rapid evaluation of hundreds of thousands of candidate strategies. This formulation enables the systematic exploration of the supportability strategy tradespace and optimization in order to identify Pareto-optimal supportability strategies that minimize mass and crew time. The methodology is verified via comparison to Monte Carlo simulation, validated via comparison to current state-of-the-art supportability analysis tools and ISS flight maintenance data, and demonstrated in two case studies.

Supportability is a critical consideration for future beyond-LEO missions. Maintenance considerations will be a much more significant driver of logistics and risk than they have been in the past, and the strategies that have supported the relatively short-endurance human spaceflight missions of the past half-century will not be effective or, in some cases, feasible, for future missions. New strategies must be developed and implemented, and given the highly challenging, resource-constrained environment of spaceflight operations, these strategies must be optimized to reduce the amount of mass and crew time required for maintenance as much as possible. The methodology presented in this dissertation provides a powerful analysis capability to address this challenge and help inform technology development, system design, and mission planning for future human space exploration.

Bibliography

[1] William Cirillo, Gordon Aaseng, Kandyce Goodliff, Chel Stromgren, and Andrew Maxwell. Supportability for Beyond Low Earth Orbit Missions. In *AIAA SPACE 2011 Conference & Exposition*, Long Beach, CA, September 2011. American Institute of Aeronautics and Astronautics. AIAA-2011-7231.

[2] National Aeronautics and Space Administration. NASA Systems Engineering Handbook. Handbook NASA/SP-2007-6105 Rev1, National Aeronautics and Space Administration, 2007.

[3] Chel Stromgren, Michelle Terry, William Cirillo, Kandyce Goodliff, and Andrew Maxwell. Design and Application of the Exploration Maintainability Analysis Tool. In *AIAA SPACE 2012 Conference & Exposition*, Pasadena, CA, September 2012. AIAA-2012-5323. AIAA-2012-5323.

[4] Henry W. Jones, Edward W. Hodgson, and Mark H. Kliss. Life Support for Deep Space and Mars. In *44th International Conference on Environmental Systems*, Tucson, AZ, July 2014. ICES-2014-074.

[5] Kriss J. Kennedy, Leslie Alexander, Rob Landis, Diane Linne, Carole Mclemore, and Edgardo Santiago-Maltonado. Technology Area 07: Human Exploration Destination Systems Roadmap. Technical report, National Aeronautics and Space Administration, Washington, DC, April 2010.

[6] Kathryn Hurlbert, Bob Bagdigian, Carol Carroll, Antony Jeevarajan, Mark Kliss, and Bhim Singh. Technology Area 06: Human Health, Life Support and Habitation Systems. Technical report, National Aeronautics and Space Administration, Washington, DC, April 2012.

[7] B. Mattfeld, C. Stromgren, H. Shyface, W. Cirillo, and K. Goodliff. Developing a crew time model for human exploration missions to Mars. In *2015 IEEE Aerospace Conference*, March 2015.

[8] Christie Bertels. Crew Maintenance Lessons Learned from ISS and Considerations for Future Manned Missions. In *SpaceOps 2006*, Rome, Italy, June 2006. American Institute of Aeronautics and Astronautics.

[9] Chel Stromgren, Kandyce E. Goodliff, William Cirillo, and Andrew Owens. The Threat of Uncertainty - Why Using Traditional Approaches for Evaluating

Spacecraft Reliability Are Insufficient for Future Human Mars Missions. In *AIAA SPACE 2016*, Long Beach, CA, September 2016. American Institute of Aeronautics and Astronautics. AIAA-2016-5307.

[10] Andrew Owens, Olivier De Weck, Chel Stromgren, Kandyce E. Goodliff, and William Cirillo. Supportability Challenges, Metrics, and Key Decisions for Future Human Spaceflight. American Institute of Aeronautics and Astronautics, September 2017. AIAA 2017-5124.

[11] Kandyce Goodliff, Chel Stromgren, Michael Ewert, James Hill, and Cherice Moore. Logistics Needs for Future Human Exploration Beyond Low Earth Orbit. In *AIAA SPACE 2017*, Orlando, FL, September 2017. American Institute of Aeronautics and Astronautics. AIAA 2017-5122.

[12] Craig C Sherbrooke. *Optimal inventory modeling of systems: multi-echelon techniques*. Kluwer Academic, Boston, 2004. OCLC: 55535467.

[13] Leif Anderson, Katrina Carter-Journet, Neil Box, Denise DiFilippo, Sean Harrington, David Jackson, and Michael Lutomski. Challenges of Sustaining the International Space Station through 2020 and Beyond: Including Epistemic Uncertainty in Reassessing Confidence Targets. In *AIAA SPACE 2012 Conference & Exposition*, Pasadena, CA, September 2012. American Institute of Aeronautics and Astronautics. AIAA-2012-5320.

[14] Alessandro Birolini. *Reliability Engineering: Theory and Practice*. Springer, Berlin, Heidelberg, 2004.

[15] Department of Defense. Department of Defense Handbook: Reliability Growth Management. Handbook MIL-HDBK-189, United States Department of Defense, February 1981.

[16] National Aeronautics and Space Administration. NASA Technical Standard: Planning, Developing and Managing an Effective Reliability and Maintainability (R&M) Program. NASA Technical Standard NASA-STD-8729.1, National Aeronautics and Space Administration, December 1998.

[17] Cecilia Haskins. Systems Engineering Handbook: A Guide for System Life Cycle Processes and Activities. Technical Report INCOSE-TP-2003-002-03.1, International Council on Systems Engineering, August 2007.

[18] Ann D. Montgomery. Logistics: An Integral Part of Cost-Efficient Space Operations. In *Space Mission Operations and Ground Data Systems - SpaceOps '96*, Munich, Germany, September 1996. European Space Agency. S096.8.016.

[19] William F. Fisher and Charles R. Price. Space Station Freedom External Maintenance Task Team Final Report. Technical report, National Aeronautics and Space Administration, Houston, TX, July 1990.

[20] Joseph R. Fragola and Richard H. McFadden. External Maintenance Rate Prediction and Design Concepts for High Reliability and Availability on Space Station Freedom. *Reliability Engineering & System Safety*, 49(3):255–273, January 1995.

[21] Tovey Bachman and Robert Kline. Model for Estimating Spare Parts Requirements for Future Missions. In *Space 2004 Conference and Exhibit*, San Diego, CA, September 2004. American Institute of Aeronautics and Astronautics. AIAA-2004-5978.

[22] Chel Stromgren, Michelle Terry, Bryan Mattfeld, William Cirillo, Kandyce E. Goodliff, Hilary R. Shyface, and Andrew J. Maxwell. Assessment of Maintainability for Future Human Asteroid and Mars Missions. In *AIAA SPACE 2013 Conference & Exposition*, San Diego, CA, September 2013. American Institute of Aeronautics and Astronautics. AIAA-2013-5328.

[23] Kevin E. Lange and Molly S. Anderson. Reliability Impacts in Life Support Architecture and Technology Selection. In *42nd International Conference on Environmental Systems*, San Diego, CA, July 2012. AIAA-2012-3491. AIAA-2012-3491.

[24] Andrew Owens, Sydney Do, Andrew Kurtz, and Olivier de Weck. Benefits of Additive Manufacturing for Human Exploration of Mars. In *45th International Conference on Environmental Systems*, Bellevue, WA, July 2015. International Conference on Environmental Systems. ICES-2015-287.

[25] Andrew Owens and Olivier de Weck. Systems Analysis of In-Space Manufacturing Applications for the International Space Station and the Evolvable Mars Campaign. In *AIAA SPACE 2016*, Long Beach, CA, September 2016. American Institute of Aeronautics and Astronautics. AIAA-2016-5394.

[26] James F. Russell, David M. Klaus, and Todd J. Mosher. Applying Analysis of International Space Station Crew-Time Utilization to Mission Design. *Journal of Spacecraft and Rockets*, 43(1):130–136, January 2006.

[27] James F. Russell and David M. Klaus. Maintenance, reliability and policies for orbital space station life support systems. *Reliability Engineering & System Safety*, 92(6):808–820, June 2007.

[28] K. Leath and J.L. Green. A Comparison of Space Station Utilization and Operations Planning to Historical Experience. In *44th Congress of the International Astronautical Federation*, Graz, Austria, October 1993. International Astronautical Federation. IAF-93-T.5.522.

[29] Andrew Owens and Olivier de Weck. Increasing the Fidelity of Maintenance Logistics Representation in Breakeven Plots. In *46th International Conference on Environmental Systems*, Vienna, Austria, July 2016. International Conference on Environmental Systems. ICES-2016-344.

[30] Julie Wertz. Chapter 24.2: Space System Risk Analysis. In James R. Wertz, David F. Everett, and Jeffery J. Puschell, editors, *Space Mission Engineering: the New SMAD*, Space Technology Library. Microcosm Press, Hawthorne, CA, 2011.

[31] Michael Stamatelatos and Homayoon Dezfuli. Probabilistic Risk Assessment Procedures Guide for NASA Managers and Practitioners. Technical Report NASA/SP-2011-3421, National Aeronautics and Space Administration, Washington, DC, December 2011.

[32] Thomas J. Kelly. *Moon Lander: How We Developed the Apollo Lunar Module.* Smithsonian Institution Press, Washington, DC, 2001.

[33] Douglas R. Cooke. Commercial Crew Transportation System Certification Requirements for NASA Low Earth Orbit Missions. Technical Report ESMD-CCTSCR-12.10, National Aeronautics and Space Administration Exploration Systems Mission Directorate, December 2010.

[34] David F. Everett. Chapter 14: Overview of Spacecraft Design. In James R. Wertz, David F. Everett, and Jeffery J. Puschell, editors, *Space Mission Engineering: the New SMAD*, Space Technology Library. Microcosm Press, Hawthorne, CA, 2011.

[35] Mars Architecture Steering Group. Human Exploration of Mars Design Reference Architecture 5.0 Addendum. Technical Report NASA/SP-2009-566-ADD, National Aeronautics and Space Administration, July 2009.

[36] Mars Architecture Steering Group. Human Exploration of Mars Design Reference Architecture 5.0. Technical Report NASA/SP-2009-556, National Aeronautics and Space Administration, July 2009.

[37] National Aeronautics and Space Administration. Human Exploration of Mars Design Reference Architecture 5.0 Addendum #2. Technical Report NASA/SP-2009-566-ADD2, National Aeronautics and Space Administration, March 2014.

[38] Harry Jones and Grant Anderson. Need for Cost Optimization of Space Life Support Systems. In *47th International Conference on Environmental Systems*, Charleston, SC, July 2017. International Conference on Environmental Systems. ICES-2017-83.

[39] Melanie L. Grande, Matthew Carrier, William Cirillo, Kevin D. Earle, Christopher A. Jones, Emily Judd, Jordan J. Klovstad, Andrew C. Owens, David M. Reeves, and Matthew A. Stafford. Mega-Drivers to Inform NASA Space Technology Strategic Planning. In *2018 AIAA SPACE and Astronautics Forum and Exposition*, Orlando, FL, September 2018. American Institute of Aeronautics and Astronautics. AIAA 2018-5137.

[40] Andrew Owens and Olivier de Weck. International Space Station Operational Experience and its Impacts on Future Mission Supportability. In *48th International Conference on Environmental Systems*, Albuquerque, NM, July 2018. International Conference on Environmental Systems. ICES-2018-198.

[41] Andrew C. Owens, Olivier L. de Weck, Chel Stromgren, Kandyce Goodliff, and William Cirillo. Accounting for Epistemic Uncertainty in Mission Supportability Assessment: A Necessary Step in Understanding Risk and Logistics Requirements. In *47th International Conference on Environmental Systems*, Charleston, SC, July 2017. International Conference on Environmental Systems. ICES-2017-109.

[42] Sydney Do. *Towards Earth Independence - Tradespace Exploration of Long-Duration Crewed Mars Surface System Architectures.* Doctoral Thesis, Massachusetts Institute of Technology, 2016.

[43] Richard P. Reysa, John P. Lumpkin, Dina El Sherif, Robert Kay, and David E. Williams. International Space Station (ISS) Carbon Dioxide Removal Assembly (CDRA) Desiccant/Adsorbent Bed (DAB) Orbital Replacement Unit (ORU) Redesign. In *37th International Conference on Environmental Systems*, Chicago, IL, July 2007. SAE International. SAE 2007-01-3181.

[44] Dina El Sherif and James C. Knox. International Space Station Carbon Dioxide Removal Assembly (ISS CDRA) Concepts and Advancements. In *35th International Conference on Environmental Systems*, Rome, Italy, July 2005. SAE 2005-01-2892.

[45] Layne Carter, Barry Tobias, and Nicole Orozco. Status of ISS Water Management and Recovery. In *42nd International Conference on Environmental Systems*, San Diego, CA, July 2012. American Institute of Aeronautics and Astronautics. AIAA-2012-3594.

[46] Layne Carter. Status of the Regenerative ECLS Water Recovery System. In *40th International Conference on Environmental Systems*, Barcelona, Spain, July 2010. American Institute of Aeronautics and Astronautics. AIAA-2010-6216.

[47] Samuel I. Wald. Advanced Habitation Strategies for Aggressive Mass Reduction. In *AIAA SPACE 2015 Conference and Exposition*, Pasadena, CA, August 2015. American Institute of Aeronautics and Astronautics. AIAA-2015-4449.

[48] Afreen Siddiqi and Olivier L. De Weck. Spare Parts Requirements for Space Missions with Reconfigurability and Commonality. *Journal of Spacecraft and Rockets*, 44(1):147–155, January 2007.

[49] Harry Jones. Common Cause Failures and Ultra Reliability. In *42nd International Conference on Environmental Systems*, San Diego, CA, July 2012. American Institute of Aeronautics and Astronautics. AIAA-2012-3602.

[50] H. Y. (Jannivine) Yeh, Frank F. Jeng, Cheryl B. Brown, Chin H. Lin, and Michael K. Ewert. Advanced Life Support Sizing Analysis Tool (ALSSAT) Using Microsoft® Excel. In *31st International Conference on Environmental Systems*, Orlando, FL, July 2001. SAE International. SAE 2001-01-2304.

[51] Karen M.B Taminger, Robert A. Hafley, and Dennis L. Dicus. Solid Freeform Fabrication: An Enabling Technology for Future Space Missions, April 2002.

[52] Melanie Bodiford, Julie Ray, Scott Gilley, James Kennedy, and Richard Howard. Are We There Yet?....Developing In-situ Fabrication & Repair Technologies to Explore and Live on the Moon and Mars. American Institute of Aeronautics and Astronautics, January 2005. AIAA-2005-2624.

[53] National Research Council. *3D Printing in Space*. July 2014.

[54] T.J. Prater, Q.A. Bean, R.D. Beshears, T.D. Rolin, N.J. Werkheiser, R.M. Ordonez, R.M. Ryan, and F.E. Ledbetter III. Summary Report on Phase I Results from the 3d Printing in Zero-G Technology Demonstration Mission, Volume I. Technical Report NASA/TP-2016-219101, National Aeronautics and Space Administration, Huntsville, AL, July 2016.

[55] Mallory M. Johnston, Mary J. Werkheiser, Michael P. Snyder, and Jennifer Edmunson. 3d Printing In Zero-G ISS Technology Demonstration. In *AIAA SPACE 2014*, San Diego, CA, August 2014. American Institute of Aeronautics and Astronautics. AIAA-2014-4470.

[56] Kenneth Cooper, Carole McLemore, and Theodore Anderson. Cases for Additive Manufacturing on the International Space Station. In *50th AIAA Aerospace Sciences Meeting including the New Horizons Forum and Aerospace Exposition*, Nashville, TN, January 2012. American Institute of Aeronautics and Astronautics. AIAA-2012-0517.

[57] J. C. Grenouilleau, O. Housseini, and F. Pérès. In-Situ Rapid Spares Manufacturing and Its Application to Human Space Missions. pages 42–48. American Society of Civil Engineers, February 2000.

[58] Robert M. Bagdigian, Jason Dake, Gregory Gentry, and Matt Gault. International Space Station Environmental Control and Life Support System Mass and Crewtime Utilization In Comparison to a Long Duration Human Space Exploration Mission. In *45th International Conference on Environmental Systems*, Bellevue, WA, July 2015. International Conference on Environmental Systems. ICES-2015-094.

[59] Robert Shishko, René Fradet, Sydney Do, Serkan Saydam, Carlos Tapia-Cortez, Andrew G. Dempster, and Jeff Coulton. Mars Colony in situ resource utilization: An integrated architecture and economics model. *Acta Astronautica*, 138:53–67, September 2017.

[60] Gerald Sanders. ISRU - An overview of NASA's current development activities and long-term goals. In *38th Aerospace Sciences Meeting and Exhibit*, Reno,NV,U.S.A., January 2000. American Institute of Aeronautics and Astronautics.

[61] Gerald B. Sanders and William E. Larson. Progress Made in Lunar In Situ Resource Utilization under NASA's Exploration Technology and Development Program. *Journal of Aerospace Engineering*, 26(1):5–17, January 2013.

[62] Michael Flynn and Sanders D. Rosenberg. In Situ Production of High Density Polyethylene and Other Useful Materials on Mars. July 2005.

[63] J. Edmunson and C. A. McLemore. In Situ Manufacturing is a Necessary Part of Any Planetary Architecture. In *Concepts and Approaches for Mars Exploration*, Houston, TX, June 2012.

[64] Carole A. McLemore, John C. Fikes, Charles A. Darby, James E. Good, and Scott D. Gilley. Fabrication Capabilities Utilizing in Situ Materials. In *AIAA SPACE 2008 Conference & Exposition*, San Diego, CA, September 2008. American Institute of Aeronautics and Astronautics.

[65] Brian Dunbar. Mercury Crewed Flights Summary, March 2018.

[66] David R. Williams. The Gemini Program (1962-1966), December 2004.

[67] David R. Williams and E. Bell II. NASA Space Science Data Coordinated Archive, March 2017.

[68] Brian Dunbar. The Skylab Crewed Missions, August 2017.

[69] Amiko Kauderer. Space Shuttle Mission Archives, August 2011.

[70] David M. Harland. *The story of Space Station Mir*. Springer ; Published in association with Praxis Publishing, Berlin ; New York : Chichester, UK, 2005.

[71] Mark Garcia. Visiting Vehicle Launches, Arrivals and Departures, November 2018.

[72] National Aeronautics and Space Administration. Reference guide to the International Space Station. Technical Report NP-2010-09-682-HQ, National Aeronautics and Space Administration, Washington, DC, 2010.

[73] National Research Council. *Pathways to Exploration: Rationales and Approaches for a U.S. Program of Human Space Exploration*. June 2014.

[74] Matthew A. Simon, Samuel I. Wald, A. S. Howe, and Larry Toups. Evolvable Mars Campaign Long Duration Habitation Strategies: Architectural Approaches to Enable Human Exploration Missions. In *AIAA SPACE 2015 Conference and Exposition*, AIAA SPACE Forum, Pasadena, CA, August 2015. American Institute of Aeronautics and Astronautics. AIAA-2015-4514.

[75] Tariq Malik. NASA's Space Shuttle By the Numbers: 30 Years of a Spaceflight Icon. *Space.com*, July 2011.

[76] Jeremy S. (Jeremy Sundermeyer) Agte. *Multistate analysis and design : case studies in aerospace design and long endurance systems*. Doctoral Thesis, Massachusetts Institute of Technology, 2011.

[77] Paul I. Barton. *Mixed-Integer and Nonconvex Optimization*. Massachusetts Institute of Technology, Cambridge, MA, 2011.

[78] Stuart J. Russell and Peter Norvig. *Artificial Intelligence: a Modern Approach*. Prentice-Hall Series in Artificial Intelligence. Prentice-Hall, Upper Saddle River, NJ, 3rd edition, 2010.

[79] Jean-Christophe Grenouilleau. Chapter 28: Space Logistics Support. In Wiley J. Larson and Linda K. Pranke, editors, *Human Spaceflight Mission Analysis and Design*, Space Technology Series. McGraw-Hill Higher Education, New York, NY, 1999.

[80] Craig C. Sherbrooke. Metric: A Multi-Echelon Technique for Recoverable Item Control. *Operations Research*, 16(1):122–141, February 1968.

[81] John A. Muckstadt. A Model for a Multi-Item, Multi-Echelon, Multi-Indenture Inventory System. *Management Science*, 20(4-part-i):472–481, December 1973.

[82] F.M. Slay. VARI-METRIC: An Approach to Modeling Multi-Echelon when the Demand Process is Poisson with a Gamma Prior. Technical Report AF501-2, Logistics Management Institute, Washington, DC, 1984.

[83] F. Michael Slay, Tovey C. Bachman, Robert C. Kline, T.J. O'Malley, Frank L. Eichorn, and Randall M. King. Optimizing Spares Support: The Aircraft Sustainability Model. Technical Report AF501MR1, Logistics Management Institute, McLean, VA, October 1996.

[84] Craig C. Sherbrooke. VARI-METRIC: Improved Approximations for Multi-Indenture, Multi-Echelon Availability Models. *Operations Research*, 34(2):311–319, April 1986.

[85] Stephen C. Graves. A Multi-Echelon Inventory Model for a Repairable Item with One-for-One Replenishment. *Management Science*, 31(10):1247–1256, October 1985.

[86] John Joseph Shynk. *Probability, random variables, and random processes: theory and signal processing applications*. Wiley, Hoboken, NJ, 2013.

[87] Dimitris Karlis and Evdokia Xekalaki. Mixed Poisson Distributions. *International Statistical Review*, 73(1):35–58, January 2007.

[88] Robert Kline and Craig Sherbrooke. Inventory based upon system availability for the Space Station Freedom. American Institute of Aeronautics and Astronautics, November 1991.

[89] Robert C. Kline and Craig C. Sherbrooke. Estimating Spares Requirements for Space Station Freedom Using the M-SPARE Model. Technical Report NS101R2, Logistics Management Institute, Bethseda, MD, July 1993.

[90] Bennett Fox. Discrete Optimization Via Marginal Analysis. *Management Science*, 13(3):210–216, November 1966.

[91] Andrei Sleptchenko and Matthieu van der Heijden. Joint optimization of redundancy level and spare part inventories. *Reliability Engineering & System Safety*, 153:64–74, September 2016.

[92] Robert C. Kline and Tovey C. Bachman. Estimating Spare Parts Requirements with Commonality and Redundancy. *Journal of Spacecraft and Rockets*, 44(4):977–984, July 2007.

[93] United States Government Accountability Office. International Space Station: Approaches for Ensuring Utilization through 2020 Are Reasonable but Should Be Revisited as NASA Gains More Knowledge of On-Orbit Performance. Report to Congressional Committees GAO-12-162, United States Government Accountability Office, Washington, DC, December 2011. GAO-12-162.

[94] Roberto Vitali and Michael G. Lutomski. Derivation of Failure Rates and Probability of Failures for the International Space Station Probabilistic Risk Assessment Study. In *International Conference on Probabilistic Safety Assessment and Management*, Berlin, June 2004.

[95] Department of Defense. Military Handbook: Reliability Prediction of Electronic Equipment. Handbook MIL-HDBK-217F, United States Department of Defense, December 1991.

[96] Homayoon Dezfuli, Dana Kelly, Curtis Smith, Kurt Vedros, and William Galyean. Bayesian Inference for NASA Probabilistic Risk and Reliability Analysis. Technical Report NASA/SP-2009-569, National Aeronautics and Space Administration, Washington, DC, June 2009.

[97] United States General Accounting Office. Space Station: Improving NASA's Planning for External Maintenance. Report to the Chair, Government Activities and Transportation Subcommittee, Committee on Government Operations, House of Representatives GAO/NSIAD-92-271, United States General Accounting Office, Washington, DC, July 1992.

[98] Brian T. Soldon. *The International Space Station Comparative Maintenance Analysis Model (CMAM)*. Master of Science, Naval Postgraduate School, Monterey, CA, September 2004.

[99] Chel Stromgren. Personal communication (email), July 2018.

[100] Chel Stromgren, Felipe Escobar, Steven Rivadeneira, William Cirillo, and Kandyce Goodliff. Predicting Crew Time Allocations for Lunar Orbital Missions Base on Historical ISS Operational Activities. In *AIAA SPACE Forum 2018*, Orlando, FL, September 2018. American Institute of Aeronautics and Astronautics. AIAA 2018-5407.

[101] Chel Stromgren, Felipe Escobar, Molly S. Anderson, Imelda Stambaugh, Miriam J. Sargusingh, and Kandyce E. Goodliff. Assessment of Desired ECLSS Closure Rates for Human Mars Missions. In *AIAA SPACE and Astronautics Forum and Exposition*, Orlando, FL, September 2017. American Institute of Aeronautics and Astronautics. AIAA 2017-5123.

[102] Bryan Mattfeld, Chel Stromgren, Hilary Shyface, David Komar, William Cirillo, and Kandyce E. Goodliff. Trades Between Opposition and Conjunction Class Trajectories for Early Human Missions to Mars. In *AIAA SPACE 2014 Conference and Exposition*, San Diego, CA, August 2014. American Institute of Aeronautics and Astronautics. AIAA 2014-4333.

[103] Matthew Simon, Kara Latorella, John Martin, Jeff Cerro, Roger Lepsch, Sharon Jefferies, Kandyce Goodliff, David Smitherman, Carey McCleskey, and Chel Stromgren. NASA's advanced exploration systems Mars transit habitat refinement point of departure design. In *2017 IEEE Aerospace Conference*, pages 1–34, Big Sky, MT, USA, March 2017. IEEE.

[104] Eric McVay, Christopher A. Jones, and Raymond G. Merrill. Cis-Lunar Reusable In-Space Transportation Architecture for the Evolvable Mars Campaign. In *AIAA SPACE 2016*, Long Beach, California, September 2016. American Institute of Aeronautics and Astronautics. AIAA 2016-5493.

[105] Kevin Takada, Ahmed E. Ghariani, Steven Van Keuren, and Andrew C. Owens. Oxygen Generation Assembly Design for Exploration Missions. In *48th International Conference on Environmental Systems*, Albuquerque, NM, July 2018. International Conference on Environmental Systems. ICES-2018-133.

[106] Frank A. Tillman, Ching-Lai Hwang, and Way Kuo. Optimization Techniques for System Reliability with Redundancy – A Review. *IEEE Transactions on Reliability*, R-26(3):148–155, August 1977.

[107] Zhigang Tian, Gregory Levitin, and Ming J. Zuo. A joint reliability–redundancy optimization approach for multi-state series–parallel systems. *Reliability Engineering & System Safety*, 94(10):1568–1576, October 2009.

[108] Karin S. de Smidt-Destombes, Nicole P. van Elst, Ana Isabel Barros, Harm Mulder, and Jan A.M. Hontelez. A spare parts model with cold-standby redundancy on system level. *Computers & Operations Research*, 38(7):985–991, July 2011.

[109] Jeremy Agte, Nicholas Borer, and Olivier de Weck. Design of Long-Endurance Systems With Inherent Robustness to Partial Failures During Operations. *Journal of Mechanical Design*, 134(10):100903, 2012.

[110] Nicholas Borer and Jeremy Agte. A Multistate Design Methodology for Effecting Robust Mission Performance of Long Endurance UAVs. American Institute of Aeronautics and Astronautics, January 2012.

[111] Oussama Adjoul, Khaled Benfriha, Améziane Aoussat, and Yacine Benabid. New Approach for the Joint Optimization of the Design and Maintenance of Multi-Component Systems by Integration of Life Cycle Costs. In *IRF2018*, Lisbon, Portugal, July 2018. INEGI/FEUP. Paper #7153.

[112] José Alexandre Matelli and Kai Goebel. Resilience evaluation of the environmental control and life support system of a spacecraft for deep space travel. *Acta Astronautica*, August 2018.

[113] Liang Zhao, Christian Bühler, and Andreas M. Hein. 3d-Printing on Mars: Trade-off Between In-situ Spare Parts Production on Mars and Spare Parts Supply from Earth. Technical report, July 2014. Working Paper.

[114] Patrick J. Sears and Koki Ho. Impact Evaluation of In-Space Additive Manufacturing and Recycling Technologies for On-Orbit Servicing. *Journal of Spacecraft and Rockets*, pages 1–11, August 2018.

[115] David Komar, Jim Hoffman, Aaron Olds, and Mike Seal. Framework for the Parametric System Modeling of Space Exploration Architectures. In *AIAA SPACE 2008 Conference & Exposition*, San Diego, California, September 2008. American Institute of Aeronautics and Astronautics.

[116] Larry Toups, Matthew Simon, David Smitherman, and Gary Spexarth. Design and Parametric Sizing of Deep Space Habitats Supporting NASA's Human Space Flight Architecture Team. In *2012 Global Space Exploration Conference*, Paris, May 2012. International Astronautical Federation. GLEX-2012.05.3.5.x12280.

[117] Peter J. Stuckey, Thibaut Feydy, Andreas Schutt, Guido Tack, and Julien Fischer. The MiniZinc challenge 2008-2013. *AI Magazine*, 35(2):55–60, 2014.

[118] Paolo Dragone. PyMzn, 2017.

[119] Brian C. Williams and Robert Ragno. Conflict-directed A* and its Role in Model-based Embedded Systems. *Special Issue on Theory and Applications of Satisfiability Testing, Journal of Discrete Applied Math*, 155(12):1562–1595, June 2003.

[120] Robert Ragno. Solving Optimal Satisfiability Problems Through Clause-Directed A*. Master's thesis, Massachusetts Institute of Technology, May 2002.

[121] Christian Schulte, Mikael Lagerkvist, and Guido Tack. Gecode: The Generic Constraint Development Environment, May 2018.

[122] Leif Anderson, Sean Harrington, Ojei Omeke, and Douglas Schwaab. Improving the Estimates of International Space Station (ISS) Induced "K-Factor" Failure Rates for On-Orbit Replacement Unit (ORU) Supportability Analyses. American Institute of Aeronautics and Astronautics, September 2009.

[123] Gregory J. Gentry and Steven F. Balistreri Jr. International Space Station (ISS) Environmental Control and Life Support (ECLS) System Overview of Events: 2017-2018. In *48th International Conference on Environmental Systems*, Albuquerque, NM, July 2018. International Conference on Environmental Systems.

[124] Ernesto Estrada and Desmond J. Higham. Network Properties Revealed through Matrix Functions. *SIAM Review*, 52(4):696–714, January 2010.

[125] N. Nethercote, P.J. Stuckey, R. Becket, S. Brand, G.J. Duck, and G. Tack. MiniZinc: Towards a standard CP modelling language. In *Proceedings of the 13th International Conference on Principles and Practice of Constraint Programming*, volume 4741 of *Lecture Notes in Computer Science*, pages 529–543. Springer, 2007.

[126] Charles E. Ebeling. *An introduction to reliability and maintainability engineering.* Waveland Press, Long Grove, Ill, 2nd ed edition, 2010. OCLC: ocn449888067.

[127] Kunio Shimizu and Edwin L. Crow. History, Genesis, and Properties. In Edwin L. Crow and Kunio Shimizu, editors, *Lognormal distributions: theory and applications*, number vol. 88 in Statistics, textbooks and monographs, pages 1–26. M. Dekker, New York, 1988.

[128] S. A. Shaban. Poisson-Lognormal Distributions. In Edwin L. Crow and Kunio Shimizu, editors, *Lognormal distributions: theory and applications*, number vol. 88 in Statistics, textbooks and monographs, pages 195–210. M. Dekker, New York, 1988.

[129] Geoffrey J. McLachlan and David Peel. *Finite mixture models.* Wiley series in probability and statistics. Applied probability and statistics section. Wiley, New York, 2000.

[130] Richard L. Warr. Numerical Approximation of Probability Mass Functions via the Inverse Discrete Fourier Transform. *Methodology and Computing in Applied Probability*, 16(4):1025–1038, December 2014.

[131] Frederick S. Hillier and Gerald J. Lieberman. *Introduction to Operations Research.* McGraw-Hill Series in Industrial Engineering and Management Science. McGraw-Hill, Boston, 7th edition, 2001.

[132] Oliver Johnson, Ioannis Kontoyiannis, and Mokshay Madiman. Log-concavity, ultra-log-concavity, and a maximum entropy property of discrete compound Poisson measures. *Discrete Applied Mathematics*, 161(9):1232–1250, June 2013.

[133] Mark Bagnoli and Ted Bergstrom. Log-concave probability and its applications. *Economic Theory*, 26(2):445–469, August 2005.

[134] R.L. Graham. An efficient algorithm for determining the convex hull of a finite planar set. *Information Processing Letters*, 1(4):132–133, June 1972.

[135] A.M. Andrew. Another efficient algorithm for convex hulls in two dimensions. *Information Processing Letters*, 9(5):216–219, December 1979.

[136] Mark de Berg, Otfried Cheong, Marc van Kreveld, and Mark Overmars. *Computational Geometry: Algorithms and Applications*. Springer Berlin Heidelberg, Berlin, Heidelberg, 3rd edition, 2008.

[137] Panos M. Pardalos, Antanas Žilinskas, and Julius Žilinskas. Definitions and Examples. In *Non-Convex Multi-Objective Optimization*, volume 123, pages 3–12. Springer International Publishing, Cham, 2017.

[138] Panos M. Pardalos, Antanas Žilinskas, and Julius Žilinskas. Multi-Objective Branch and Bound. In *Non-Convex Multi-Objective Optimization*, volume 123, pages 45–56. Springer International Publishing, Cham, 2017.

[139] Alan K. Mackworth. Consistency in networks of relations. *Artificial Intelligence*, 8(1):99–118, February 1977.

[140] Layne Carter, Kevin Takada, Christopher A. Brown, Jesse Bazley, Daniel Gazda, Ryan Schaezler, and Frank Thomas. Status of ISS Water Management and Recovery. In *47th International Conference on Environmental Systems*, Charleston, SC, July 2017. International Conference on Environmental Systems. ICES-2017-036.

[141] David E. Williams and Gregory J. Gentry. International Space Station Environmental Control and Life Support System Status: 2006 - 2007. In *37th International Conference on Environmental Systems*, Chicago, IL, July 2007. SAE International. SAE 2007-01-3098.

[142] David E. Williams and Gregory J. Gentry. International Space Station Environmental Control and Life Support System Status: 2008 - 2009. In *39th International Conference on Environmental Systems*, Savannah, GA, July 2009. SAE International. SAE 2009-01-2415.

[143] Kevin C. Takada, Ahmed E. Ghariani, and Steven Van Keuren. Advancing the Oxygen Generation Assembly Design to Increase Reliability and Reduce Costs for a Future Long Duration Mission. In *45th International Conference on Environmental Systems*, Bellevue, WA, July 2015. International Conference on Environmental Systems. ICES-2015-115.

[144] Ronald L. Wasserstein and Nicole A. Lazar. The ASA's Statement on p-Values: Context, Process, and Purpose. *The American Statistician*, 70(2):129–133, April 2016.

[145] Robert M. Bagdigian and Dale Cloud. Status of the International Space Station Regenerative ECLSS Water Recovery and Oxygen Generation Systems. July 2005. 2005-01-2779.

[146] Robert J. Erickson, John Howe, Galen W. Kulp, and Steven P. Van Keuren. International Space Station United States Orbital Segment Oxygen Generation System On-orbit Operational Experience. *SAE International Journal of Aerospace*, 1(1):15–24, June 2008. 2008-01-1962.

[147] Sydney Do, Andrew Owens, and Olivier de Weck. HabNet – An Integrated Habitation and Supportability Architecting and Analysis Environment. In *45th International Conference on Environmental Systems*, Bellevue, WA, July 2015. International Conference on Environmental Systems. ICES-2015-289.

[148] Sydney Do, Andrew Owens, Koki Ho, Samuel Schreiner, and Olivier de Weck. An Independent Assessment of the Technical Feasibility of the Mars One Mission Plan – Updated Analysis. *Acta Astronautica*, 120:192–228, March 2016.

[149] Olivier L. de Weck, David Simchi-Levi, Robert Shishko, Jaemyung Ahn, Erica L. Gralla, Diego Klabjan, Jason Mellein, Sarah A. Shull, Afreen Siddiqi, Brian K. Baristow, and Gene Y. Lee. SpaceNet v1.3 User's Guide. NASA Technical Report NASA/TP-2007-214725, NASA/MIT, January 2007.

[150] Gene Lee, Elizabeth Jordan, Robert Shishko, Olivier de Weck, Nii Armar, and Afreen Siddiqi. SpaceNet: Modeling and Simulating Space Logistics. In *AIAA SPACE 2008 Conference & Exposition*, San Diego, California, September 2008. American Institute of Aeronautics and Astronautics.

[151] Paul Grogan, Howard Yue, and Olivier De Weck. Space Logistics Modeling and Simulation Analysis using SpaceNet: Four Application Cases. In *AIAA SPACE 2011 Conference & Exposition*, Long Beach, California, September 2011. American Institute of Aeronautics and Astronautics.