



NAVAL POSTGRADUATE SCHOOL

MONTEREY, CALIFORNIA

THESIS

ROLLING HORIZON IMPLEMENTATION OF THE REPLENISHMENT AT SEA PLANNER

by

Daniel S. Wright

September 2023

Thesis Advisor:

Co-Advisor:

Second Reader:

Emily M. Craparo

W. Matthew Carlyle

Jeffrey E. Kline

Approved for public release. Distribution is unlimited.

THIS PAGE INTENTIONALLY LEFT BLANK

REPORT DOCUMENTATION PAGE			<i>Form Approved OMB No. 0704-0188</i>	
Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instruction, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188) Washington, DC, 20503.				
1. AGENCY USE ONLY (Leave blank)		2. REPORT DATE September 2023	3. REPORT TYPE AND DATES COVERED Master's thesis	
4. TITLE AND SUBTITLE ROLLING HORIZON IMPLEMENTATION OF THE REPLENISHMENT AT SEA PLANNER			5. FUNDING NUMBERS	
6. AUTHOR(S) Daniel S. Wright				
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) Naval Postgraduate School Monterey, CA 93943-5000			8. PERFORMING ORGANIZATION REPORT NUMBER	
9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES) N/A			10. SPONSORING / MONITORING AGENCY REPORT NUMBER	
11. SUPPLEMENTARY NOTES The views expressed in this thesis are those of the author and do not reflect the official policy or position of the Department of Defense or the U.S. Government.				
12a. DISTRIBUTION / AVAILABILITY STATEMENT Approved for public release. Distribution is unlimited.			12b. DISTRIBUTION CODE A	
13. ABSTRACT (maximum 200 words) To sustain ships on station and create a forward presence overseas, the United States Navy relies on replenishment at sea (RAS) operations. Using these events, shuttle ships provide fuel and materiel to deployed combatants while underway. With a global commitment and dynamic schedules, the Navy faces a complex, operational logistics problem. A mixed integer linear program called the Replenishment at Sea Planner (RASP) provides the optimal schedule; however, schedulers use a heuristic to develop less-than-optimal RAS plans due to RASP's computation time. Ill-planned schedules limit the number of combatants in theater, reduce time-on-station for deployers, and increase costs due to shuttles consuming more fuel. The purpose of this research is to implement a rolling horizon algorithm with future relaxation into the current version of RASP. A rolling horizon partitions the time horizon into subproblems then optimizes each subproblem sequentially using a portion of the previous solution. The analysis examines both computation times and objective values across four scenarios, comparing these results to the monolith model. We then apply the rolling horizon as a heuristic for warm starting the monolith model. Compared to the monolith's performance with a two-hour computation limit, the rolling horizon implementation can provide operational quality replenishment plans, improving theaterwide sustainment and readiness at a cost savings of \$15–22 million over a sixty-day period.				
14. SUBJECT TERMS replenishment, optimization, integer linear programming, scheduling, rolling horizon, receding horizon, time cascades, operational logistics, delivery planner, relax-and-fix, replenishment at sea, (RAS), Replenishment at Sea Planner (RASP)			15. NUMBER OF PAGES 105	
			16. PRICE CODE	
17. SECURITY CLASSIFICATION OF REPORT Unclassified	18. SECURITY CLASSIFICATION OF THIS PAGE Unclassified	19. SECURITY CLASSIFICATION OF ABSTRACT Unclassified	20. LIMITATION OF ABSTRACT UU	

NSN 7540-01-280-5500

Standard Form 298 (Rev. 2-89)
Prescribed by ANSI Std. Z39-18

THIS PAGE INTENTIONALLY LEFT BLANK

Approved for public release. Distribution is unlimited.

**ROLLING HORIZON IMPLEMENTATION OF THE REPLENISHMENT
AT SEA PLANNER**

Daniel S. Wright
Lieutenant Commander, United States Navy
BS, Ohio State University, 2012

Submitted in partial fulfillment of the
requirements for the degree of

MASTER OF SCIENCE IN OPERATIONS RESEARCH

from the

**NAVAL POSTGRADUATE SCHOOL
September 2023**

Approved by: Emily M. Craparo
Advisor

W. Matthew Carlyle
Co-Advisor

Jeffrey E. Kline
Second Reader

W. Matthew Carlyle
Chair, Department of Operations Research

THIS PAGE INTENTIONALLY LEFT BLANK

ABSTRACT

To sustain ships on station and create a forward presence overseas, the United States Navy relies on replenishment at sea (RAS) operations. Using these events, shuttle ships provide fuel and materiel to deployed combatants while underway. With a global commitment and dynamic schedules, the Navy faces a complex, operational logistics problem. A mixed integer linear program called the Replenishment at Sea Planner (RASP) provides the optimal schedule; however, schedulers use a heuristic to develop less-than-optimal RAS plans due to RASP's computation time. Ill-planned schedules limit the number of combatants in theater, reduce time-on-station for deployers, and increase costs due to shuttles consuming more fuel. The purpose of this research is to implement a rolling horizon algorithm with future relaxation into the current version of RASP. A rolling horizon partitions the time horizon into subproblems then optimizes each subproblem sequentially using a portion of the previous solution. The analysis examines both computation times and objective values across four scenarios, comparing these results to the monolith model. We then apply the rolling horizon as a heuristic for warm starting the monolith model. Compared to the monolith's performance with a two-hour computation limit, the rolling horizon implementation can provide operational quality replenishment plans, improving theaterwide sustainment and readiness at a cost savings of \$15–22 million over a sixty-day period.

THIS PAGE INTENTIONALLY LEFT BLANK

Table of Contents

1	Introduction	1
1.1	Background	1
1.2	Research Problem and Objectives	2
1.3	Contributions and Outline	4
2	Literature Review	5
2.1	Optimization Techniques and Rolling Horizons	5
2.2	Studies on Naval Replenishments and Routing	6
3	Model and Applications	9
3.1	Rolling Horizon Approach	9
3.2	Rolling Horizon Formulation of RASP.	13
3.3	RASP File Flow.	26
3.4	Warm Start.	29
4	Experimentation and Results	31
4.1	Methodology	31
4.2	Hardware and Software	32
4.3	Scenario 1: Pacific Transit with Zone Restriction	32
4.4	Scenario 2: Pacific Transit without Zone Restriction	37
4.5	Scenario 3: 7th Fleet Operations	42
4.6	Scenario 4: 90-Day Replenishment Plan	50
5	Conclusions and Future Work	57
5.1	Future Work	58
	Appendix: Tabulated Results	61
	List of References	79

List of Figures

Figure 1.1	USN operational logistics	1
Figure 1.2	Connected RAS	2
Figure 2.1	Myopic cascade	7
Figure 3.1	Myopic rolling horizon	10
Figure 3.2	Rolling horizon with future relaxation	11
Figure 3.3	Categorization of shuttle leg selection	12
Figure 3.4	RASP file flow	27
Figure 3.5	Batch output example	29
Figure 3.6	Process flow with optional warm start	30
Figure 4.1	Objective value against time for scenario 1	34
Figure 4.2	Configuration bubble plot for scenario 1	35
Figure 4.3	DFM sawtooth for scenario 1	36
Figure 4.4	Optimal solution treemap for scenario 1	37
Figure 4.5	Objective value against time for scenario 2	38
Figure 4.6	Configuration bubble plot for scenario 2	39
Figure 4.7	Rolling horizon configuration bubble plot for scenario 2	40
Figure 4.8	DFM sawtooth for scenario 2	41
Figure 4.9	Best solution treemap for scenario 2	42
Figure 4.10	Objective value against time for scenario 3	44
Figure 4.11	Configuration bubble plot for scenario 3	45

Figure 4.12	Rolling horizon configuration bubble plot for scenario 3	46
Figure 4.13	Efficiency and effectiveness trade-off plot for scenario 3	47
Figure 4.14	Objective value against time using a warm start	49
Figure 4.15	Optimality gaps for scenario 3	50
Figure 4.16	Objective value against time for scenario 4	52
Figure 4.17	Objective value progression for scenario 4	53
Figure 4.18	Rolling horizon vs. warm start (1-hour limit)	55
Figure 4.19	Rolling horizon vs. warm start (2-hour limit)	55
Figure 4.20	Rolling horizon vs. warm start (3-hour limit)	56

List of Tables

Table 3.1	Sections of the RASP Output File	28
Table 4.1	Scenario 1 configurations	33
Table 4.2	Scenario 2 configurations	43
Table 4.3	Scenario 4 pivot table	51
Table 4.4	Performance summary for 90-day scenario	52
Table 4.5	Rolling horizon vs warm start summary	54
Table A.1	Scenario 1 results (1-30)	62
Table A.2	Scenario 1 results (31-60)	63
Table A.3	Scenario 2 results (1-30)	64
Table A.4	Scenario 2 results (31-60)	65
Table A.5	Scenario 3 results (1-30)	66
Table A.6	Scenario 3 results (31-39)	67
Table A.7	Scenario 4 warm start results (1-30)	68
Table A.8	Scenario 4 warm start results (31-60)	69
Table A.9	Scenario 4 warm start results (61-90)	70
Table A.10	Scenario 4 warm start results (91-120)	71
Table A.11	Scenario 4 warm start results (121-150)	72
Table A.12	Scenario 4 warm start results (151-174)	73
Table A.13	Scenario 4 rolling horizon results (1-30)	74
Table A.14	Scenario 4 rolling horizons results (31-60)	75
Table A.15	Scenario 4 rolling horizon results (61-90)	76

Table A.16	Scenario 4 rolling horizon results (91-120)	77
Table A.17	Scenario 4 rolling horizon results (120-135)	78

List of Acronyms and Abbreviations

A2AD	anti-access/area denial
CLF	Combat Logistics Force
CONSOL	consolidated cargo replenishment at sea
CSG	carrier strike group
CSV	comma-separated values
DFM	diesel fuel marine
DOD	Department of Defense
GAMS	General Algebraic Modeling System
GTP	geo-time posit
INREP	in-port replenishment
KML	Keyhole Markup Language
MILP	mixed-integer linear program
NPS	Naval Postgraduate School
RAM	random access memory
RAS	replenishment at sea
RASP	Replenishment at Sea Planner
UNREP	underway replenishment
USMC	U.S. Marine Corps
USN	U.S. Navy

THIS PAGE INTENTIONALLY LEFT BLANK

Executive Summary

To facilitate a forward posture that enables the ability to respond to crises, the U.S. Navy (USN) often resupplies its surface combatants while underway (Department of the Navy 2001). By transferring fuel, repair parts, provisions, and mail underway, a replenishment at sea (RAS) greatly increases a combatant's readiness without resorting to an in-port replenishment (INREP) and reduction in time-on-station. For each USN Fleet, the responsibility of creating replenishment plans supporting complex operations with a limited number of shuttles belongs to a team of in-theater schedulers. With both customers and shuttles traversing vast geographic areas while tracking various commodities, the optimal solution finding the balance between effectiveness and efficiency for a multi-month schedule depends upon thousands of decisions while satisfying operational constraints.

Realizing that manual scheduling is both labor-intensive and unable to reach optimality in terms of minimizing delivery costs while meeting demand, the USN's Fleets utilize the Replenishment at Sea Planner (RASP). Developed by a team from the Naval Postgraduate School (NPS), RASP is a mixed-integer linear program (MILP) that captures the many inputs and constraints required, then provides the optimal replenishment plan, schedule, and associated administrative reports (Brown et al. 2017). This model can achieve quality solutions; however, the size of the MILP results in large computation times, preventing practical use at the operational level where planners need to produce schedules in a matter of hours, not days. Due to this limitation, schedulers often use a lower-quality but faster heuristic incorporated within the RASP workbook to coordinate replenishments. Lower-quality replenishment plans reduce time-on-station for combatants, increase fuel burn and costs for shuttles, and limit the number of supported units in theater.

In other applications concerning optimization, modelers can implement a rolling horizon heuristic to reduce computation times for large MILPs. A rolling horizon partitions the time horizon into subproblems then optimizes each subproblem sequentially using a portion of the previous solution. By applying this approach to the RASP model, our research seeks to answer the following questions:

1. Can the implementation of a rolling horizon approach enable RASP to achieve

- operational-quality solutions in reduced computation time?
2. Can a solution obtained via rolling horizon provide an effective warm start for the monolith model?

To decrease optimality drift and avoid infeasibility, we use a rolling horizon approach with future relaxation for the RASP model. A myopic rolling horizon approach imparts the risk of infeasibility or major solution quality reductions due to drift that occurs traversing through the full horizon. Each iterative solve does not consider future constraints, potentially resulting in solutions that drift away from the full horizon's optimal solution. In contrast to the two categories in the myopic version, a rolling horizon with future relaxation divides binary variables into binary, fixed, and relaxed variables depending on where the current window lies as shown in Figure 1. With each iteration covering the full horizon, future periods influence the binary variables within the window, decreasing optimality drift without significantly increasing computational costs. More importantly, applying future relaxation to RASP's rolling horizon guarantees that the model satisfies time, speed, and distance requirements to satisfy pinned events and avoid infeasibility.

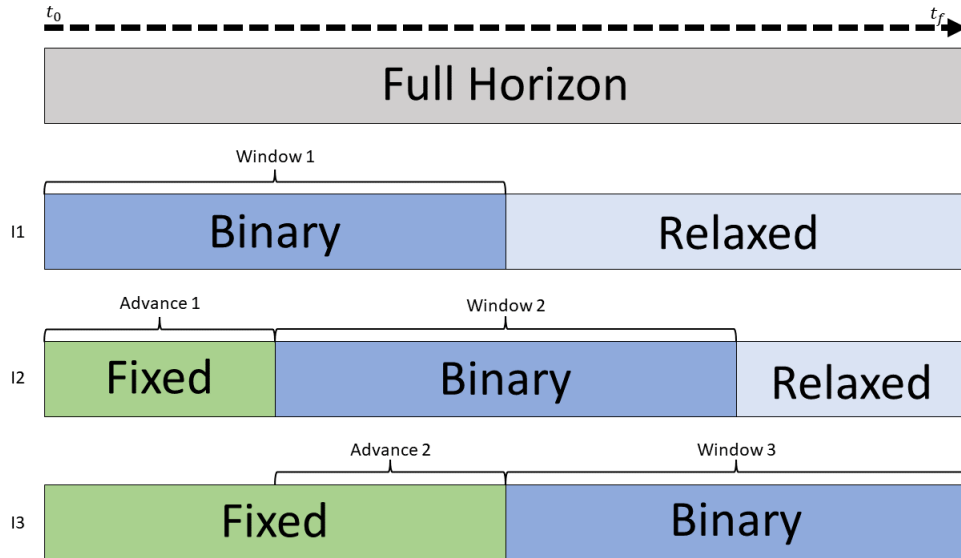


Figure 1. Three iterations of a rolling horizon with future relaxation.

To generate our results, we create four different scenarios with varying complexity to compare rolling horizon configurations against the monolith model. In our analysis, the

primary measures of effectiveness are the costs expressed in the objective function (in \$K) and computation times. We also evaluate the efficacy of using a rolling horizon configuration to warm start the monolith model by comparing absolute optimality gaps under a strict computation time limit. Simulating time-constrained planning for the USN’s 7th Fleet, we estimate that the rolling horizon improves theaterwide sustainment and readiness at a cost savings of \$15-22 million over a sixty-day period as shown in Figure 2.

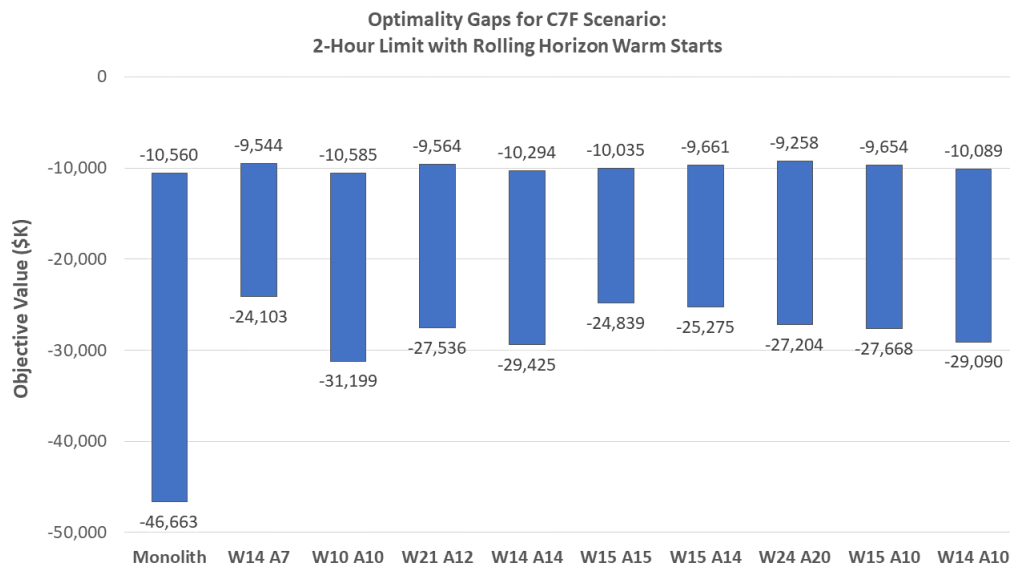


Figure 2. In this experiment, all configurations achieve similar best bounds (tops of bars); however, the warm start applications significantly improve the best integer solution (bottoms of bars) compared to the monolith model.

Our experimental results across these four scenarios support three primary conclusions:

1. The rolling horizon implementation is an effective tool for RASP, leading to operational-quality solutions in operationally-relevant time.
2. Imposing geographic restrictions on shuttles can prevent RASP from generating effective replenishment plans, leading to increased costs.
3. The rolling horizon approach provides a high-quality solution that can be used to warm start the monolith model.

With a faster model for use at the operational level, theaters can potentially increase combatant time-on-station, support more ships in-theater, and reduce operating costs for the Combat Logistics Force (CLF).

References

Brown G, DeGrange W, Price W, Rowe A (2017) Scheduling Combat Logistics Force replenishments at sea for the US Navy. *Naval Research Logistics* 64:677–693, <https://doi.org/10.1002/nav.21780>.

Department of the Navy (2001) *Underway Replenishment*. NWP 4-01.4. Washington, DC, USA, <https://www.maritime.dot.gov/sites/marad.dot.gov/files/2023-01/unrep-nwp04-01.pdf>.

Acknowledgments

I would like to express my genuine appreciation to my advisory team, colleagues, and family for their contributions throughout my journey. Your guidance, support, and commitment have been invaluable throughout this process.

To both my faculty advisors, Professor Matt Carlyle and Professor Emily Craparo, thank you for the countless hours and expertise dedicated toward guiding me through this research. To my second reader, Professor Jeff Kline, thank you for applying your operational and analytical experience in your feedback. To Professor Jerry Brown, thank you for all your previous work in developing this model and providing critical insights along the way. And to Anton Rowe, thank you for your assistance in generating scenarios and navigating the RASP workbook.

I am also appreciative of my past and present colleagues from the Fleet, who have not only been exceptional shipmates but also incredible friends. To my NPS cohort, your fellowship and enthusiasm through this challenging curriculum have truly enriched my time in Monterey.

Lastly, I offer my sincere gratitude to my family. Your love has provided me with the strength to pursue this degree. To Grace and Alice, your devotion, patience, and support during these past two years has been nothing but exceptional. *Thank you.*

THIS PAGE INTENTIONALLY LEFT BLANK

CHAPTER 1: Introduction

1.1 Background

As stated by Admiral Michael M. Gilday, Chief of Naval Operations,

Naval forces provide the United States strategic advantages in position, influence, and flexibility, independent of access to overseas land bases. Our forward posture guarantees our Nation the ability to respond to crisis, blunt gray-zone activities, and preserve a stable and secure global maritime order (2023).

To facilitate this forward posture and increase time-on-station, the U.S. Navy (USN) often resupplies its surface combatants while underway (Department of the Navy 2001). Figure 1.1 illustrates how this process works. Commercial, Department of Defense (DOD), and USN assets push food and materiel to forward bases or ports. These ports then load Combat Logistics Force (CLF) shuttles consisting of oilers or dry cargo/ammunition ships. Following this load event, the shuttle leaves port and sails to meet its customer(s), USN surface combatants.

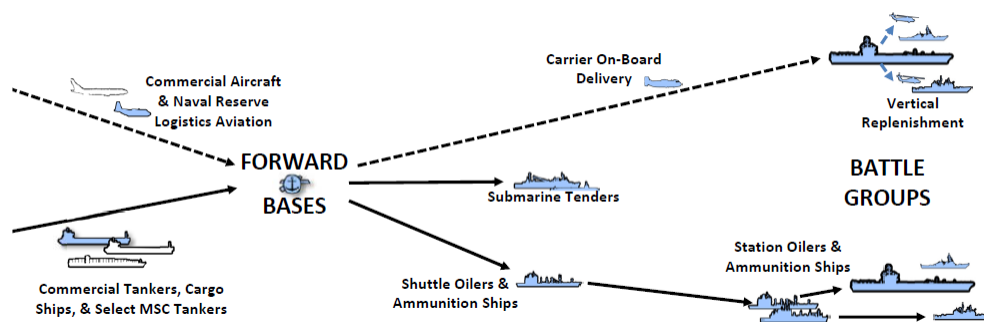


Figure 1.1. This schematic shows a basic representation of the USN's operational logistics network supporting surface combatants. Source: Walton et al. (2019).

At approximately 200 feet apart, both shuttle and customer continue sailing and complete a replenishment at sea (RAS), also known as an underway replenishment (UNREP). During a RAS, the shuttle connects a fuel probe to its customer while helicopters or connected replenishment rigs transfer palletized materiel, as shown in Figure 1.2. By transferring fuel, repair parts, provisions, and mail underway, a RAS greatly increases a combatant's readiness without resorting to an in-port replenishment (INREP) and the resulting reduction in time-on-station.



Figure 1.2. USS Shiloh (CG 67) receives frozen stores from USNS Alan Shepard (T-AKE 3) during a RAS on November 2, 2020. Source: Military Sealift Command (2020).

1.2 Research Problem and Objectives

For each USN Fleet, the responsibility of creating replenishment plans supporting complex operations with a limited number of shuttles belongs to a team of in-theater schedulers. To enable this support of deployed surface combatants, USN doctrine provides eight responsibilities for RAS schedulers:

1. Issue standing UNREP guidance
2. Monitor inventories/quantities of commodities on board assigned CLF ships in company with or enroute to the task force/group

3. Monitor inventory status of commodities on task force/group ships, identify problem areas to the logistics coordinator, and recommend solutions
4. Recommend UNREP and resupply/consol schedules to the logistics coordinator
5. Ensure that UNREP ships pass through the designated rendezvous point on time
6. Coordinate with the forward logistics site representative to prioritize and direct the loading of fleet freight, fresh fruit and vegetables, and dairy products
7. In the event of an anticipated commodity shortage, ensure equitable distribution of available inventories to task force/group ships
8. During extended at-sea operations, when ships are not afforded opportunities for periodic pierside offload of dry plastics trash and/or hazardous material, arrange routine transfer of these materials from on-station forces to disposal facilities ashore via shuttle ship (Department of the Navy 2001)

Although doctrine only lists eight tasks, RAS schedulers face a complicated operational logistics problem. Kress (2002) defines operational logistics as “a collection of means, resources, organizations and processes that share the common goal of sustaining campaigns and large-scale military operations.” Logisticians at the operational level often must balance the competing interests of effectiveness at the tactical level against efficiencies gained at the strategic level (Kress 2002), and RAS scheduling contains the same balancing requirement. Replenishment plans prioritizing unit readiness may never have a customer breach an inventory safety threshold, but these plans may result in inefficient shuttle routing and high commodity expenses, driving up costs in a resource-constrained environment. Conversely, replenishment plans emphasizing efficiency may save costs, but combatants may not have the inventory necessary to respond effectively during routine tasking or a crisis. With both customers and shuttles traversing vast geographic areas while tracking various commodities, the optimal solution finding the balance between effectiveness and efficiency for a multi-month schedule depends upon thousands of decisions while satisfying operational constraints.

Realizing that manual scheduling is both labor-intensive and unable to reach optimality, the USN’s Fleets utilize the Replenishment at Sea Planner (RASP). Developed by a team from the Naval Postgraduate School (NPS), RASP is a mixed-integer linear program (MILP) that captures the many inputs and constraints required, then provides the optimal replenishment plan, schedule, and associated administrative reports (Brown et al. 2017). This model can

achieve quality solutions; however, the size of the MILP results in large computation times, preventing practical use at the operational level where planners need to produce schedules in a matter of hours, not days. Due to this limitation, schedulers often use a lower-quality but faster heuristic incorporated within the RASP workbook to coordinate replenishments. Lower-quality replenishment plans reduce time-on-station for combatants, increase fuel burn and costs for shuttles, and limit the number of supported units in theater.

In other applications concerning optimization, modelers can implement a rolling horizon heuristic to reduce computation times for large MILPs. A rolling horizon partitions the time horizon into subproblems then optimizes each subproblem sequentially using a portion of the previous solution. By applying this approach to the RASP model, our research seeks to answer the following questions:

1. **Can the implementation of a rolling horizon approach enable RASP to achieve operational-quality solutions in reduced computation time?**
2. **Can a solution obtained via rolling horizon provide an effective warm start for the monolith model?**

1.3 Contributions and Outline

This thesis describes a rolling horizon formulation of the RASP using future relaxation to sequentially optimize iterative MILP subproblems. We implement this updated formulation and create a batch running script to test multiple configurations for four scenarios. Lastly, we use the rolling horizon as a heuristic to provide an initial feasible solution to warm start the monolith model and investigate solution quality and convergence of optimality gaps. The remainder of this thesis is organized as follows:

- **Chapter 2** provides a literature review of optimization techniques, the rolling horizon approach, and related research on naval replenishment and routing.
- **Chapter 3** describes the formulation and implementation of the rolling horizon technique into the RASP model.
- **Chapter 4** describes our computational experimentation and results for four operational scenarios.
- **Chapter 5** presents our conclusions and recommendations for follow-on research.

CHAPTER 2:

Literature Review

This chapter completes a brief review of literature and studies concerning naval replenishments, ship routing, optimization techniques, and rolling horizon applications.

2.1 Optimization Techniques and Rolling Horizons

The exclusive use of hard constraints in an optimization model can result in unstable model behavior, including infeasibility with changes in parameters or functions (Royset and Wets 2021). To mitigate this risk, elastic programming (sometimes called constraint softening) is commonly used in optimization. By adding penalty functions that allow constraint violations but induce a per-unit cost added to the objective, the model stabilizes and enjoys a reduced risk of infeasibility (Royset and Wets 2021). The addition of elastic constraints bounds the associated dual variable(s), leading to improved control, performance, and solution quality of the model. (Brown 2004). To provide better replenishment plans and avoid infeasibility, the RASP model contains elastic constraints. Recently, Guo et al. (2020) apply constraint softening to an adaptive cruise control strategy, increasing robustness and avoiding infeasibility due to overresponses.

The introduction of multiple penalty terms in the objective function results in a multi-objective formulation for RASP. When dealing with multiple objectives, modelers often use scalarization to produce a singular, real-value output allowing the use of single objective techniques to solve the problem (Miettinen 1998). As a recent example, Dönmez (2023) evaluates multiple scalarization methods applied toward the aircraft sequencing and scheduling problem.

Modelers have implemented rolling horizon approaches, also known as receding horizons, cascades, or relax-and-fix heuristics, for a wide range of applications. Chand et al. (2002) provide an extensive bibliography of over 200 works, classifying rolling horizon forecast and optimization studies applied toward various industries. Dating back to the 1950s, Modigliani and Hohn (1955) and Johnson (1957) develop models to sequentially optimize production planning, while Wagner and Whitin (1958) implement disjoint planning horizons

to optimize the economic lot-size inventory model. In his experiments using the dynamic lot-size model, Baker (1977) finds the majority of the tested rolling horizons achieve a 10% or better optimality gap, while well-constructed models terminate within 1% of optimality. Sethi and Sorger (1991) incorporate forecast costs and a rolling horizon framework for a generalized, stochastic optimization problem. To overcome the computation times associated with large integer programs to plan production, Dillenberger et al. (1994) introduce a “fix-and-relax” heuristic that partitions the horizon into several periods as well as relax future binary constraints. Lastly, Stadler (2003) implements a rolling horizon with linear relaxation periods to optimize the multi-lot inventory problem.

More recently, modelers have expanded their applications and studies of rolling horizons. Glomb et al. (2022) introduce a new rolling horizon algorithm and effectively implement the new approach toward the tail-assignment problem, minimizing costs for aircraft scheduling. In addition, Joncour et al. (2023) generalize the “relax-and-fit” algorithm with a “restrict-and-fix” algorithm, then apply this method to the vehicle routing problem and portfolio optimization. Analyzing worst case behavior for relax-and-fix heuristics, Absi and van den Heuvel (2019) conclude small time windows do not bound worst case performance as well as larger time windows or greater overlaps do not guarantee better solutions. Employing both relax-and-fix and fix-and-optimize algorithms, Friske et al. (2022) solve the maritime inventory routing problem across a standard time-space network and a fixed charge network flow setup. Marquantand et al. (2015) implement a rolling horizon for an energy hub model, optimizing the conversion and distribution of energy. Recently, Sentinella (2021) applies a rolling horizon to schedule the optimal surface delivery of materiel and personnel in support of U.S. Marine Corps (USMC) amphibious operations.

2.2 Studies on Naval Replenishments and Routing

Identifying a critical need to improve the USN’s logistics, Brown et al. (2017) formulate and develop the RASP tool, which generates optimal replenishment schedules and key administrative reports. Prior to RASP, schedulers relied on experience and generic guidelines to develop replenishment plans, requiring intensive time and effort. Due to the complexity of constructing schedules covering extensive time periods, geographic areas, and mobile forces, these plans seldom satisfied all the constraints and requirements, leading to nonviable plans, short-notice changes, and increased fuel consumption. To avoid these issues,

the RASP model generates candidate transit arcs then selects shuttle routes to minimize the costs associated with shuttle fuel consumption, commodity and event costs, and inventory violations. With the USN’s adoption of RASP, schedulers worldwide use the workbook’s “Quick Solve” heuristic for the generation of schedules, sustaining fleets in far-off places. Other applications of RASP include investigations into the Navy’s safety stock policy as well as the robustness of current logistics programs.

A limited snapshot of Brown’s work shows additional contributions toward sea-based logistics and the speeding up of integer programs with cascades. To minimize costs associated with ocean-going oil tankers, Brown et al. develop an integer program to produce optimal routing (1987). By creating the CLF Planner, Brown and Carlyle optimize force employment with an integer linear program, creating a tool to investigate force structure, basing, policies, and operational plans (2008). Both these studies employ cascades which partition the full models into blocks. The models sequentially solve these blocks, fixing previous integer solutions and linearly relaxing future constraints. Figure 2.1 shows a version of a myopic cascade that iterates through time. Lastly, Washburn and Brown develop the USN’s current WetRoute method, connecting the shortest route on a sphere while avoiding obstacles (2016).

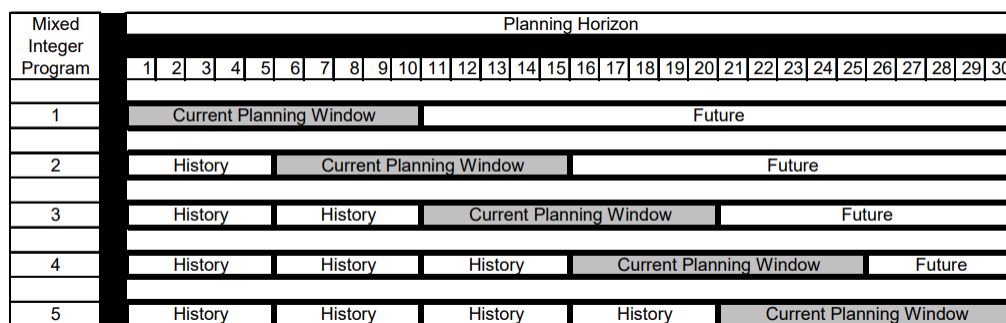


Figure 2.1. A myopic cascade partitioning a horizon into planning blocks based on time intervals. Source: Doyle (2006).

NPS has an extensive collection of theses concerning vessel routing and naval logistics, influencing and building upon RASP (Brown 2021). To circumvent software restrictions on government computers, Diaz (2010) develops a heuristic to provide a feasible solution to the CLF Planner. In Miller (2012), Deleon (2015), Blackburn (2016), and Baker (2019), the

authors investigate or develop tools concerning the optimal routing or allocation of USN vessels. Additional work by Doyle (2006), Long (2011), Hathorn (2013), Crawford (2014), Cribbs (2016), and Deiter (2022) analyzes policies, practices, and settings for optimal vessel routing or replenishments. Addressing degraded environments, Lantz (2014) provides ship scheduling to avoid cyclones, while Colburn (2015) and Krenz (2018) investigate logistics support in an anti-access/area denial (A2AD) setting.

CHAPTER 3:

Model and Applications

This chapter describes our rolling horizon approach and the associated changes to the RASP model formulation and Python implementation. In addition to various configurations for the rolling horizon, the user can also select a warm start option to solve a scenario by first constructing a feasible solution using a rolling horizon approach, then initializing the full “monolith” with this solution for further polishing.

3.1 Rolling Horizon Approach

For large MILPs such as RASP, current solvers often struggle to return optimal or even feasible solutions in reasonable time. The objective function and constraints of a MILP are linear, but the formulation restricts a portion of decision variables to be integer or binary. Due to the loss of convexity and with n integer variables, the branch and bound algorithm and its variants found in commercial software can require computation times that are exponential in n (Sierksma and Zwols 2015).

The idea behind a rolling horizon approach is to decrease the computation time by breaking up the full planning horizon into sequential subproblems. To characterize the configuration of a rolling horizon, we use the terms “window” and “advance.” The window describes the set of time periods in the subproblem where integer variables are kept integer (as opposed to being relaxed), while the advance defines the amount of time moved forward from one subproblem to the next. The advance determines the amount of overlap between solves and cannot exceed the length of the window. To abbreviate configurations, we use a “WXXAXX” notation, where “WXX” represents the window length while “AXX” represents the advance. For example, W20A10 denotes a window of 20 days and an advance of 10 days.

Figure 3.1 shows a schematic representation of a myopic rolling horizon approach. Whereas the monolith model considers binary variables throughout the full horizon, the myopic rolling horizon divides binary variables into binary and fixed variables depending on where the window currently lies. The first iteration solves a smaller MILP within the initial window. The window then advances forward, and the rolling horizon fixes any binary variable before

the second window to the value obtained in the first iteration. With a new window and fixed variables defined, the subsequent iteration solves the next subproblem. This algorithm repeats until the window covers the end of the full horizon. If the last window of the rolling horizon extends beyond the end of the full horizon, the algorithm shortens the window to align with the full horizon's termination.

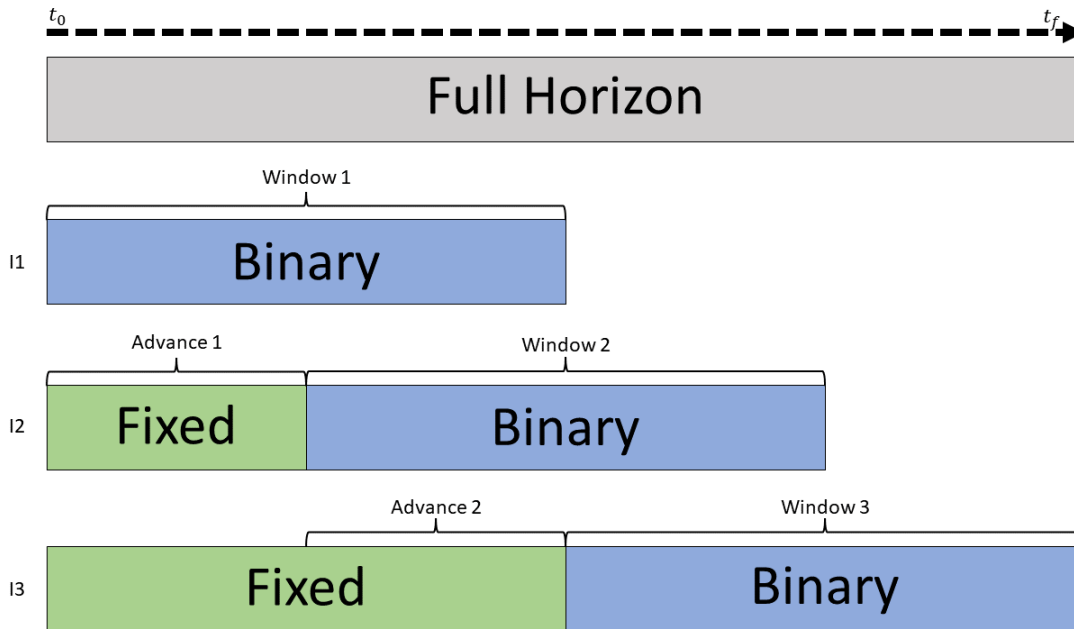


Figure 3.1. An example of a myopic rolling horizon showing three iterative solves. Windows show the size of the MILP subproblem, while the advance determines how much overlap occurs between solves.

A myopic rolling horizon approach imparts the risk of infeasibility or major solution quality reductions due to drift that occurs traversing through the full horizon. Each iterative solve does not consider future constraints outside of the current window, potentially resulting in solutions that drift away from the full horizon's optimal solution. As a result of this drift, an iteration of the myopic rolling horizon may also produce a set of constraints that cannot be satisfied, leading to infeasibility. This risk of missing a requirement presents itself with RASP's "pinned events." Pinned events are critical replenishments requiring a specified date and location due to operational necessities such as a customer arriving into a fleet's area of responsibility. These events are expressed as hard constraints in RASP. If a myopic

rolling horizon approach were to be implemented in RASP, solutions obtained early in the rolling horizon process may position shuttles in locations from which it is impossible to satisfy the time, speed, and distance requirements of a pinned event that occurs in a future window.

To decrease optimality drift and avoid infeasibility, we use a rolling horizon approach with future relaxation for the RASP model. In contrast to the two categories in the myopic version, a rolling horizon with future relaxation divides binary variables into binary, fixed, and relaxed variables depending on where the current window lies. Figure 3.2 provides a schematic illustration of the future relaxation feature. In each iteration, integer or binary variables that appear after the current window are replaced with their linear relaxation. With each iteration covering the full horizon, future periods influence the binary variables within the window, decreasing optimality drift without significantly increasing computational costs. More importantly, applying future relaxation to RASP's rolling horizon guarantees that the model satisfies time, speed, and distance requirements to satisfy pinned events and avoid infeasibility.

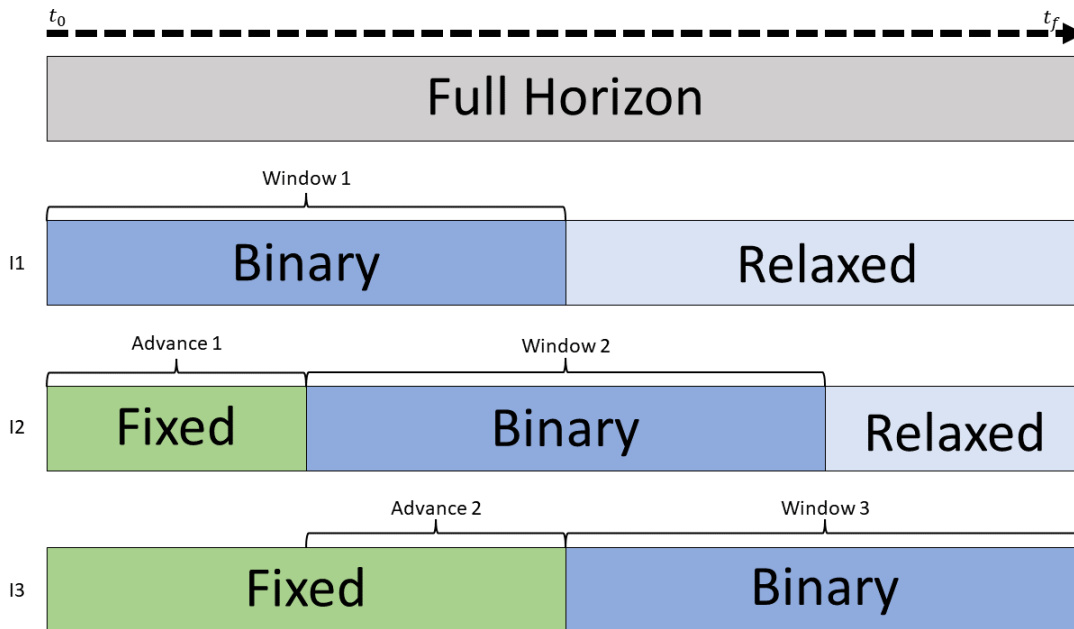


Figure 3.2. An example of a rolling horizon with future relaxation showing three iterative solves. Excluding the last iteration, the model considers relaxed variables past the window to avoid optimality drift and infeasibility.

By categorizing binary variables based on period association, the model establishes the set membership necessary for the formulation with a rolling horizon. For RASP binary variables indexed by a single date, the model determines set membership based on which period contains the date. The binary variables concerning selection of shuttle legs encompass two dates: each transit arc is specified by geo-time posits (GTPs) at the head and tail. A GTP consists of a (date, location) pair that uniquely identifies where and when an event occurs. To categorize the binary variables concerning shuttle leg selection, the model determines set membership based on the arc tail's GTP relative to the window, allowing categorization and at least one binary evaluation. Figure 3.3 illustrates how shuttle leg selection is categorized in this implementation of a rolling horizon with future relaxation.

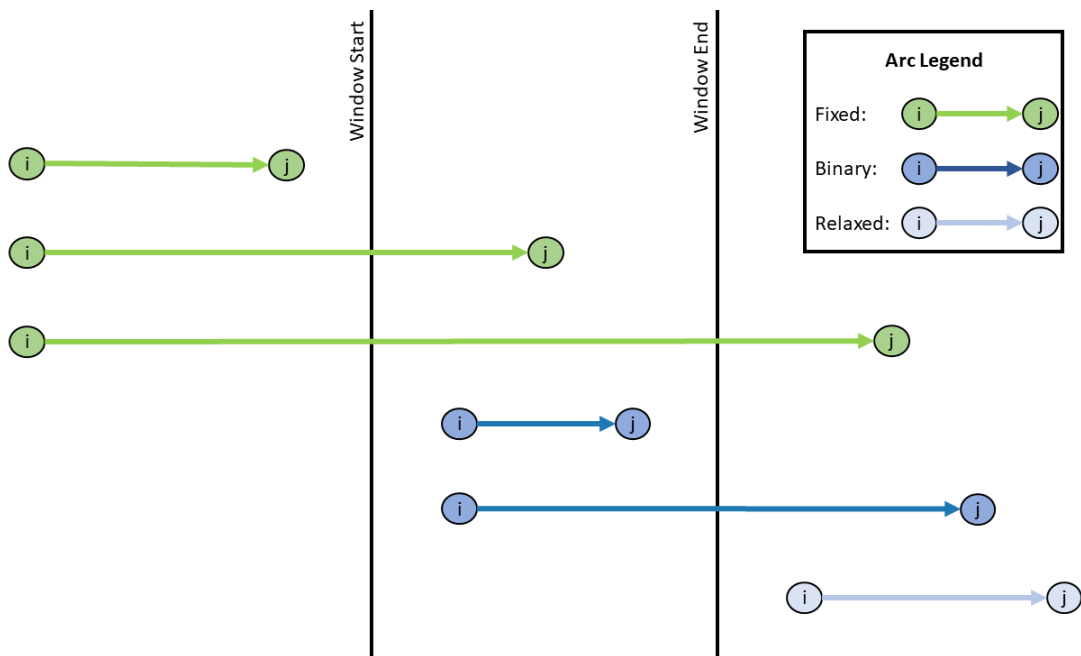


Figure 3.3. A transit arc encompasses two dates: one at its head and the other at its tail. To determine set membership for the selection of shuttle legs, the rolling horizon implementation of RASP uses the arc tail's GTP relative to the window.

3.2 Rolling Horizon Formulation of RASP

Brown et al. (2017) provide the original formulation for RASP and utilize the General Algebraic Modeling System (GAMS) for its implementation. This thesis generates a new formulation with a rolling horizon implementation derived from the RASP version 217 solve file, written in the Python programming language.

3.2.1 Sets and Indices

$c \in C$	commodities
$w \in W$	threshold levels
$a \in A$	storage areas
$m \in M$	customers
$n \in N$	strike group or deployer status
$s \in S$	shuttles
$v \in V$	vendors
$t, t' \in T$	ordinal dates
$i, j, k \in I$	GTPs

3.2.2 Derived Sets and Dictionaries

Derived Sets

$(n, i, m) \in NIM$	strike group composition at a GTP
$(t, i) \in TI$	date and GTP set
$(m, t, s, c) \in MTSC$	feasible commodity transfer during UNREP
$(m, t, s) \in MTS$	feasible UNREP event
$(s, t, v, c) \in STVC$	feasible commodity transfer during load event
$(s, t, v) \in STV$	feasible load event
$(m, t, v, c) \in MTVC$	feasible commodity transfer during INREP
$(m, t, v) \in MTV$	feasible INREP event
$(m, t, c) \in MTC$	feasible customer inventory
$(s, t, c) \in STC$	feasible shuttle inventory
$(v, t, c) \in VTC$	feasible vendor inventory
$(m, t) \in MT$	feasible customer event
$(m, t) \in Cycle_MT$	customer replenishment window for a cycle
$(m, t, t') \in Cycle_MTT$	customer replenishment window with date pairs
$(s, t, a) \in STA$	shuttle storage area
$(v, t, c) \in Limit_VTC$	non-negative feasible vendor inventory
$(s, t) \in ST$	shuttle date availability
$(v, t) \in VT$	vendor date availability
$(w, m, t, c) \in WMTC$	threshold enforcement epoch
$(s, i, j) \in Arc$	transit arcs for shuttles
$(s, i) \in First_SI$	initial GTPs for shuttles
$(s, i) \in Last_SI$	terminal GTPs for shuttles
$(s, i) \in Flow_SI$	candidate GTPs for shuttle voyage

Dictionaries

$GTP_To_T_i = t$	dates corresponding to GTPs for $(t, i) \in TI$
$MTS_To_IJ_{m,t,s} = (i, j)$	transit arcs corresponding to customer-shuttle dates
$STV_To_IJ_{s,t,v} = (i, j)$	transit arcs corresponding to shuttle-vendor dates
$First_SI_To_J_{s,i} = j$	tail GTPs corresponding to initial shuttle GTPs
$SK_To_J_{s,k} = j$	penultimate GTPs corresponding to $(s, k) \in Last_SI$
$Flow_SI_To_I_{s,j} = i$	inflow through GTP j
$Flow_SI_To_K_{s,j} = k$	outflow through GTP j

Rolling Horizon Sets

$Fix_T = \{t \in T \mid t < Win_Start\}$
$Bin_T = \{t \in T \mid Win_Start \leq t \leq Win_End\}$
$Rel_T = \{t \in T \mid Win_End < t\}$
$Fix_I = \{i \in I \mid GTP_To_T_i < Win_Start\}$
$Bin_I = \{i \in I \mid Win_Start \leq GTP_To_T_i \leq Win_End\}$
$Rel_I = \{i \in I \mid Win_End < GTP_To_T_i\}$
$Fix_Arc = \{(s, i, j) \in Arc \mid GTP_To_T_i < Win_Start\}$
$Bin_Arc = \{(s, i, j) \in Arc \mid Win_Start \leq GTP_To_T_i \leq Win_End\}$
$Rel_Arc = \{(s, i, j) \in Arc \mid Win_End < GTP_To_T_i\}$

3.2.3 Parameters

Win_Start	date t for window start
Win_End	date t for window end
$M_RAS_Hrs_m$	hours to complete an UNREP with customer m
$M_Avail_Hrs_{m,t}$	operating hours available for customer m on date t
$M_Init_C_{m,t,c}$	customer m 's initial inventory on date t of commodity c
$M_Uses_C_{m,t,c}$	consumption by customer m on date t of commodity c
$S_Cap_A_{s,a}$	max capacity of shuttle s for storage area a
$S_Avail_Hrs_{s,t}$	operating hours available for shuttle s on date t
$S_Init_C_{s,t,c}$	initial inventory of shuttle s on date t of commodity c
$V_Load_Hrs_v$	hours to load vessel by vendor v
$V_Init_C_{v,t,c}$	initial inventory for vendor v on date t of commodity c
$V_Gets_C_{v,t,c}$	vendor v on date t restocks commodity c
$V_Avail_Hrs_{v,t}$	operating hours available for vendor v on date t
$SM_Cost_{s,m}$	UNREP cost from shuttle s to customer m
$VS_Cost_{v,s}$	load cost from vendor v to shuttle s
$VM_Cost_{v,m}$	INREP cost from vendor v to customer m
$Arc_Cost_{s,i,j}$	transit cost for shuttle s from posit i to posit j
$MWC_Limit_{m,w,c}$	tripwire for customer m for threshold level w for commodity c
$MTC_Max_{m,t,c}$	max inventory amount for customer m on date t of commodity c
$STC_Max_{s,t,c}$	max inventory amount for shuttle s on date t of commodity c
$MT_Penalty_{m,t}$	scheduling penalty for customer m on date t
$VT_Penalty_{v,t}$	scheduling penalty for vendor v on date t
$WMTC_Penalty_{w,m,t,c}$	penalty rate associated with threshold w , customer m , date t , commodity c
$MTC_Reward_{m,t,c}$	reward rate for replenishing customer m on date t with commodity c
$VTC_Expense_{v,t,c}$	expense rate incurred from vendor v on date t for commodity c

3.2.4 Decision Variables

Linear Decision Variables

$M_Has_C_{m,t,c}$	customer m inventory on date t of commodity c
$V_Has_C_{v,t,c}$	vendor v inventory on date t of commodity c
$S_Has_C_{s,t,c}$	shuttle s inventory on date t of commodity c
$M_Low_{w,m,t,c}$	amount below threshold w customer m on date t of commodity c
$V_To_S_{s,t,v,c}$	amount vendor v on date t transfers to shuttle s of commodity c
$S_To_M_{m,t,s,c}$	amount shuttle s on date t transfers to customer m of commodity c
$V_To_M_{m,t,v,c}$	amount vendor v on date t transfers to customer m of commodity c

Binary Decision Variables

$M_RAS_{m,t,s}$	customer m on date t completes an UNREP with shuttle s
$S_Load_{s,t,v}$	shuttle s on date t completes a loadout with vendor v
$M_Rep_{m,t,v}$	customer m on date t completes an INREP with vendor v
$M_Visit_{m,t}$	customer m receives a replenishment on date t
$N_Visit_{n,i}$	strike group n receives a replenishment on/at GTP i
$V_Visit_{v,t}$	vendor v provides a replenishment on date t
$S_Leg_{s,i,j}$	shuttle s leg selection of transit arc (i, j)

Labeling Convention for Binary Decision Variables

$\hat{X}$	fixed variable prior to window
X	binary variable inside window
X^L	relaxed variable after window

3.2.5 Formulation

Objective Function

$$\begin{aligned}
max \quad & \sum_{(m,t,s,c) \in MTSC} MTC_Reward_{m,t,c} * S_To_M_{m,t,s,c} \\
& + \sum_{(m,t,v,c) \in MTVC} MTC_Reward_{m,t,c} * V_To_M_{m,t,v,c} \\
& - \sum_{(s,t,v,c) \in STVC} VTC_Expense_{v,t,c} * V_To_S_{s,t,v,c} \\
& - \sum_{(m,t,v,c) \in MTVC} VTC_Expense_{v,t,c} * V_To_M_{m,t,v,c} \\
& - \sum_{(w,m,t,c) \in WMTC} WMTC_Penalty_{w,m,t,c} * M_Low_{w,m,t,c} \\
& - \sum_{(m,t) \in MT} MT_Penalty_{m,t} * [\widehat{M_Visit}_{m,t} |_{t \in Fix_T} + M_Visit_{m,t} |_{t \in Bin_T} + M_Visit^L_{m,t} |_{t \in Rel_T}] \\
& - \sum_{(v,t) \in VT} VT_Penalty_{v,t} * [\widehat{V_Visit}_{v,t} |_{t \in Fix_T} + V_Visit_{v,t} |_{t \in Bin_T} + V_Visit^L_{v,t} |_{t \in Rel_T}] \\
& - \sum_{(s,i,j) \in Arc} Arc_Cost_{s,i,j} * [\widehat{S_Leg}_{s,i,j} |_{(s,i,j) \in Fix_Arc} + S_Leg_{s,i,j} |_{(s,i,j) \in Bin_Arc} + S_Leg^L_{s,i,j} |_{(s,i,j) \in Rel_Arc}] \\
& - \sum_{(m,t,s) \in MTS} SM_Cost_{s,m} * [\widehat{M_RAS}_{m,t,s} |_{t \in Fix_T} + M_RAS_{m,t,s} |_{t \in Bin_T} + M_RAS^L_{m,t,s} |_{t \in Rel_T}] \\
& - \sum_{(s,t,v) \in STV} VS_Cost_{v,s} * [\widehat{S_Load}_{s,t,v} |_{t \in Fix_T} + S_Load_{s,t,v} |_{t \in Bin_T} + S_Load^L_{s,t,v} |_{t \in Rel_T}] \\
& - \sum_{(m,t,v) \in MTV} VM_Cost_{v,m} * [\widehat{M_Rep}_{m,t,v} |_{t \in Fix_T} + M_Rep_{m,t,v} |_{t \in Bin_T} + M_Rep^L_{m,t,v} |_{t \in Rel_T}]
\end{aligned} \tag{3.1}$$

Inventory and Scheduling Constraints

$$\begin{aligned}
M_Has_C_{m,t,c} = & M_Has_C_{m,t-1,c} + \sum_{s \in S} S_To_M_{m,t,s,c} |_{(m,t,s,c) \in MTSC} \\
& + \sum_{v \in V} V_To_M_{m,t,v,c} |_{(m,t,v,c) \in MTVC} - M_Uses_C_{m,t,c} \quad \forall (m,t > 1, c) \in MTC \tag{3.2a}
\end{aligned}$$

$$\begin{aligned}
M_Has_C_{m,1,c} &= M_Init_C_{m,1,c} + \sum_{s \in S} S_To_M_{s,1,m,c} \mid_{(m,1,s,c) \in MTSC} \\
&+ \sum_{v \in V} V_To_M_{v,1,m,c} \mid_{(m,1,v,c) \in MTV C} - M_Uses_C_{m,1,c} \quad \forall (m, 1, c) \in MTC \quad (3.2b)
\end{aligned}$$

$$\begin{aligned}
S_Has_C_{s,t,c} &= S_Has_C_{s,t-1,c} + \sum_{v \in V} V_To_S_{s,t,v,c} \mid_{(s,t,v,c) \in STVC} \\
&- \sum_{m \in M} S_To_M_{m,t,s,c} \mid_{(m,t,s,c) \in MTSC} \quad \forall (s, t > 1, c) \in STC \quad (3.3a)
\end{aligned}$$

$$\begin{aligned}
S_Has_C_{s,1,c} &= S_Init_C_{s,1,c} + \sum_{v \in V} V_To_S_{v,1,s,c} \mid_{(s,1,v,c) \in STVC} \\
&- \sum_{m \in M} S_To_M_{s,1,m,c} \mid_{(m,1,s,c) \in MTSC} \quad \forall (s, 1, c) \in STC \quad (3.3b)
\end{aligned}$$

$$\begin{aligned}
V_Has_C_{v,t,c} &= V_Has_C_{v,t-1,c} - \sum_{s \in S} V_To_S_{s,t,v,c} \mid_{(s,t,v,c) \in STVC} \\
&- \sum_{m \in M} V_To_M_{m,t,v,c} \mid_{(m,t,v,c) \in MTV C} + V_Gets_C_{v,t,c} \quad \forall (v, t > 1, c) \in VTC \quad (3.4a)
\end{aligned}$$

$$\begin{aligned}
V_Has_C_{v,1,c} &= V_Init_C_{v,1,c} - \sum_{s \in S} V_To_S_{v,1,s,c} \mid_{(s,1,v,c) \in STVC} \\
&- \sum_{m \in M} V_To_M_{v,1,m,c} \mid_{(m,1,v,c) \in MTV C} + V_Gets_C_{v,1,c} \quad \forall (v, 1, c) \in VTC \quad (3.4b)
\end{aligned}$$

$$M_Has_C_{m,t,c} \leq MTC_Max_{m,t,c} \quad \forall (m, t, c) \in MTC \quad (3.5)$$

$$MWC_Limit_{m,w,c} - M_Has_C_{m,t,c} \leq M_Low_{w,m,t,c} \quad \forall (w, m, t, c) \in WMTC \quad (3.6)$$

$$\sum_{t' \in T} [\widehat{M_Visit}_{m,t'} \mid_{t' \in Fix_T} + M_Visit_{m,t'} \mid_{t' \in Bin_T} + M_Visit_{m,t'}^L \mid_{t' \in Rel_T}] \mid_{(m,t,t') \in Cycle_MTT} \leq 1 \quad \forall (m, t) \in Cycle_MT \quad (3.7)$$

$$M_Visit_{m,t} \mid_{t \in Bin_T} + M_Visit_{m,t}^L \mid_{t \in Rel_T} = N_Visit_{n,i} \mid_{t \in Bin_T} + N_Visit_{n,i}^L \mid_{t \in Rel_T} \\ \forall m, t, n, i \mid (n, i, m) \in NIM, t = GTP_To_T_i, t \notin Fix_T \quad (3.8)$$

$$\sum_{c \in C} S_Has_C_{s,t,c} \mid_{(c,a) \in Store_CA} \leq S_Cap_A_{s,a} \quad \forall (s, t, a) \in STA \quad (3.9)$$

$$V_Has_C_{v,t,c} \geq 0 \quad \forall (v, t, c) \in Limit_VTC \quad (3.10)$$

$$S_To_M_{m,t,s,c} \leq \\ MTC_Max_{m,t,c} * [\widehat{M_RAS}_{m,t,s} \mid_{t \in Fix_T} + M_RAS_{m,t,s} \mid_{t \in Bin_T} + M_RAS_{m,t,s}^L \mid_{t \in Rel_T}] \\ \forall (m, t, s, c) \in MTSC \quad (3.11)$$

$$\begin{aligned}
V_To_S_{s,t,v,c} \leq \\
STC_Max_{s,t,c} * [\widehat{S_Load}_{s,t,v} \mid_{t \in Fix_T} + S_Load_{s,t,v} \mid_{t \in Bin_T} + S_Load^L_{s,t,v} \mid_{t \in Rel_T}] \\
\forall (s, t, v, c) \in STVC \quad (3.12)
\end{aligned}$$

$$\begin{aligned}
V_To_M_{m,t,v,c} \leq \\
MTC_Max_{m,t,c} * [\widehat{M_Rep}_{m,t,v} \mid_{t \in Fix_T} + M_Rep_{m,t,v} \mid_{t \in Bin_T} + M_Rep^L_{m,t,v} \mid_{t \in Rel_T}] \\
\forall (m, t, v, c) \in MTVC \quad (3.13)
\end{aligned}$$

$$\begin{aligned}
& \sum_{s \in S} M_RAS_Hrs_m * [M_RAS_{m,t,s} \mid_{t \in Bin_T} + M_RAS^L_{m,t,s} \mid_{t \in Rel_T}] \mid_{(m,t,s) \in MTS} \\
& + \sum_{v \in V} V_Load_Hrs_v * [M_Rep_{m,t,v} \mid_{t \in Bin_T} + M_Rep^L_{m,t,v} \mid_{t \in Rel_T}] \mid_{(m,t,v) \in MTV} \\
& \leq M_Avail_Hrs_{m,t} * [M_Visit_{m,t} \mid_{t \in Bin_T} + M_Visit^L_{m,t} \mid_{t \in Rel_T}] \quad \forall (m, t) \in MT, t \notin Fix_T \\
& \quad (3.14)
\end{aligned}$$

$$\begin{aligned}
& \sum_{m \in M} M_RAS_Hrs_m * [M_RAS_{m,t,s} \mid_{t \in Bin_T} + M_RAS^L_{m,t,s} \mid_{t \in Rel_T}] \mid_{(m,t,s) \in MTS} \\
& + \sum_{v \in V} V_Load_Hrs_v * [S_Load_{s,t,v} \mid_{t \in Bin_T} + S_Load^L_{s,t,v} \mid_{t \in Rel_T}] \mid_{(s,t,v) \in STV} \\
& \leq S_Avail_Hrs_{s,t} \quad \forall (s, t) \in ST, t \notin Fix_T \quad (3.15)
\end{aligned}$$

$$\begin{aligned}
& \sum_{s \in S} V_Load_Hrs_v * [S_Load_{s,t,v} \mid_{t \in Bin_T} + S_Load_{s,t,v}^L \mid_{t \in Rel_T}] \mid_{(s,t,v) \in STV} \\
& + \sum_{m \in M} V_Load_Hrs_v * [M_Rep_{m,t,v} \mid_{t \in Bin_T} + M_Rep_{m,t,v}^L \mid_{t \in Rel_T}] \mid_{(m,t,v) \in MTV} \\
& \leq V_Avail_Hrs_{v,t} \quad \forall (v,t) \in VT, t \notin Fix_T \quad (3.16)
\end{aligned}$$

Arc Constraints

$$\begin{aligned}
& M_RAS_{m,t,s} \mid_{t \in Bin_T} + M_RAS_{m,t,s}^L \mid_{t \in Rel_T} \\
& \leq \sum_{(i,j) \in MTS_To_IJ_{m,t,s}} [\widehat{S_Leg}_{s,i,j} \mid_{(s,i,j) \in Fix_Arc} + S_Leg_{s,i,j} \mid_{(s,i,j) \in Bin_Arc} + S_Leg_{s,i,j}^L \mid_{(s,i,j) \in Rel_Arc}] \\
& \quad \forall (m,t,s) \in MTS, t \notin Fix_T \quad (3.17)
\end{aligned}$$

$$\begin{aligned}
& S_Load_{s,t,v} \mid_{t \in Bin_T} + S_Load_{s,t,v}^L \mid_{t \in Rel_T} \\
& \leq \sum_{(i,j) \in STV_To_IJ_{s,t,v}} [\widehat{S_Leg}_{s,i,j} \mid_{(s,i,j) \in Fix_Arc} + S_Leg_{s,i,j} \mid_{(s,i,j) \in Bin_Arc} + S_Leg_{s,i,j}^L \mid_{(s,i,j) \in Rel_Arc}] \\
& \quad \forall (s,t,v) \in STV, t \notin Fix_T \quad (3.18)
\end{aligned}$$

$$\begin{aligned}
& \sum_{j \in First_SI_To_J_{s,i}} [S_Leg_{s,i,j} \mid_{(s,i,j) \in Bin_Arc} + S_Leg_{s,i,j}^L \mid_{(s,i,j) \in Rel_Arc}] \\
& = 1 \quad \forall (s,i) \in First_SI, GTP_To_T_i \notin Fix_T \quad (3.19)
\end{aligned}$$

$$\begin{aligned}
& \sum_{j \in SK_To_I_{s,k}} [\widehat{S_Leg}_{s,j,k} \mid_{(s,j,k) \in Fix_Arc} + S_Leg_{s,j,k} \mid_{(s,j,k) \in Bin_Arc} + S_Leg^L_{s,j,k} \mid_{(s,j,k) \in Rel_Arc}] \\
& = 1 \quad \forall (s, k) \in Last_SI \quad (3.20)
\end{aligned}$$

$$\begin{aligned}
& \sum_{i \in Flow_SI_To_I_{s,j}} [\widehat{S_Leg}_{s,i,j} \mid_{(s,i,j) \in Fix_Arc} + S_Leg_{s,i,j} \mid_{(s,i,j) \in Bin_Arc} + S_Leg^L_{s,i,j} \mid_{(s,i,j) \in Rel_Arc}] \\
& - \sum_{k \in Flow_SI_To_K_{s,j}} [\widehat{S_Leg}_{s,j,k} \mid_{(s,j,k) \in Fix_Arc} + S_Leg_{s,j,k} \mid_{(s,j,k) \in Bin_Arc} + S_Leg^L_{s,j,k} \mid_{(s,j,k) \in Rel_Arc}] \\
& = 0 \quad \forall (s, j) \in Flow_SI \quad (3.21)
\end{aligned}$$

Decision Variable Constraints

$$\begin{aligned}
M_Has_C_{m,t,c} &\in \mathbb{R} \quad \forall m, t, c \\
V_Has_C_{v,t,c} &\in \mathbb{R} \quad \forall v, t, c \\
S_Has_C_{s,t,c} &\in \mathbb{R}^+ \quad \forall s, t, c \\
M_Low_{w,m,t,c} &\in \mathbb{R}^+ \quad \forall w, m, t, c \\
V_To_S_{s,t,v,c} &\in \mathbb{R}^+ \quad \forall (s, t, v, c) \in STVC \\
S_To_M_{m,t,s,c} &\in \mathbb{R}^+ \quad \forall (m, t, s, c) \in MTSC \\
V_To_M_{m,t,v,c} &\in \mathbb{R}^+ \quad \forall (m, t, v, c) \in MTVC \\
M_RAS_{m,t,s} &\in \{0, 1\} \quad \forall (m, t, s) \in MTS \mid t \in Bin_T \\
0 \leq M_RAS_{m,t,s}^L &\leq 1 \quad \forall (m, t, s) \in MTS \mid t \in Rel_T \\
S_Load_{s,t,v} &\in \{0, 1\} \quad \forall (s, t, v) \in STV \mid t \in Bin_T \\
0 \leq S_Load_{s,t,v}^L &\leq 1 \quad \forall (s, t, v) \in STV \mid t \in Rel_T \\
M_Rep_{m,t,v} &\in \{0, 1\} \quad \forall (m, t, v) \in MTV \mid t \in Bin_T \\
0 \leq M_Rep_{m,t,v}^L &\leq 1 \quad \forall (m, t, v) \in MTV \mid t \in Rel_T \\
M_Visit_{m,t} &\in \{0, 1\} \quad \forall (m, t) \in MT \mid t \in Bin_T \\
0 \leq M_Visit_{m,t}^L &\leq 1 \quad \forall (m, t) \in MT \mid t \in Rel_T \\
N_Visit_{n,i} &\in \{0, 1\} \quad \forall (n, i) \in NI \mid i \in Bin_I \\
0 \leq N_Visit_{n,i}^L &\leq 1 \quad \forall (n, i) \in NI \mid i \in Rel_I \\
V_Visit_{v,t} &\in \{0, 1\} \quad \forall (v, t) \in VT \mid t \in Bin_T \\
0 \leq V_Visit_{v,t}^L &\leq 1 \quad \forall (v, t) \in VT \mid t \in Rel_T \\
S_Leg_{s,i,j} &\in \{0, 1\} \quad \forall (s, i, j) \in Bin_Arc \\
0 \leq S_Leg_{s,i,j}^L &\leq 1 \quad \forall (s, i, j) \in Rel_Arc
\end{aligned}$$

3.2.6 Discussion

Objective Function

The objective function 3.1 expresses the reward of customer replenishment from a shuttle, plus the reward of customer replenishment from a vendor, less the commodity expense for a shuttle load event, less the commodity expense for a customer INREP, less the penalty incurred for a customer falling below a commodity's threshold, less the scheduling penalty incurred for a customer's replenishment, less the scheduling penalty incurred for a vendor's service, less the fuel cost associated with a selected shuttle leg, less the cost of completing an UNREP, less the cost of completing a load event, and less the cost of completing an

INREP. The units of objective function 3.1 are thousands of dollars (\$K).

Inventory and Scheduling Constraints

Each equality 3.2a accounts for customer inventory from one period to the next for each customer and commodity.

Each equality 3.2b initializes an inventory for each customer and commodity.

Each equality 3.3a accounts for shuttle inventory from one period to the next for each shuttle and commodity.

Each equality 3.3b initializes an inventory for each shuttle and commodity.

Each equality 3.4a accounts for vendor inventory from one period to the next for each vendor and commodity.

Each equality 3.4b initializes an inventory for each vendor and commodity.

Each inequality 3.5 limits customer inventory to not exceed the max volume for each customer, day, and commodity.

Each inequality 3.6 requires each inventory deficit to exceed the difference between the commodity threshold and customer inventory for each threshold type, customer, date, and commodity.

Each inequality 3.7 limits the number of customer replenishments to not exceed 1 during a replenishment cycle for each customer and date.

Each equality 3.8 requires concurrent replenishments for co-located customers and strike groups for each customer, date, strike group, and geo-time posit.

Each inequality 3.9 limits total shuttle inventory to not exceed storage capacity for each shuttle, date, and storage area.

Each inequality 3.10 requires vendor inventory to be non-negative for each vendor, date, and commodity.

Each inequality 3.11 limits the quantity of commodity transferred from shuttle to customer to not exceed the max commodity volume for each customer and only occur during an UNREP event for each customer, date, shuttle, and commodity.

Each inequality 3.12 limits the quantity of commodity transferred from vendor to shuttle to not exceed the max commodity volume for each shuttle and only occur during a load event for each shuttle, date, vendor, and commodity.

Each inequality 3.13 limits the quantity of commodity transferred from vendor to customer to not exceed the max commodity volume for each customer and only occur during an

INREP event for each customer, date, vendor, and commodity.

Each inequality 3.14 limits the customer hours spent completing UNREPs with shuttles and INREPs with vendors to not exceed each customer's available hours for each customer and date.

Each inequality 3.15 limits the shuttle hours spent completing UNREPs with customers and load events with vendors to not exceed each shuttle's available hours for each shuttle and date.

Each inequality 3.16 limits the vendor hours spent completing load events with shuttles and INREPs with customers to not exceed each vendor's available hours for each vendor and date.

Arc Constraints

Each inequality 3.17 requires an UNREP to occur on a shuttle leg selection for each customer, date, and shuttle.

Each inequality 3.18 requires a load event to occur on a shuttle leg selection for each shuttle, date, and vendor.

Each equality 3.19 requires each shuttle to start from a known start location for each shuttle and initial GTP.

Each equality 3.20 requires each shuttle to arrive at a known terminal location for each shuttle and terminal GTP.

Each equality 3.21 requires each shuttle to arrive and leave through flow locations for each shuttle and adjacent GTPs.

Note: The V_Visit variable is not triggered in this formulation but is maintained for future capabilities. If desired, inequality 3.16 can be modified to trigger the V_Visit variable similar to how inequality 3.14 triggers the M_Visit variable.

3.3 RASP File Flow

Users access RASP in a Microsoft Excel Workbook where they can update parameters involving units, schedules, and solver settings. When the user initiates the solve, the workbook generates an input file containing the data necessary for the MILP. For this thesis, the input file is read in by a Python script, building a Pyomo model reflecting the formulation from

Section 3.2. After the script builds the Pyomo model, the script accesses a commercial solver such as COIN CBC, Gurobi, or IBM CPLEX to generate a solution and create an output file. The output file then populates the Excel Workbook with report and scheduling information along with a Keyhole Markup Language (KML) file for an interactive Google Earth visualization. Figure 3.4 illustrates the file flow for RASP.

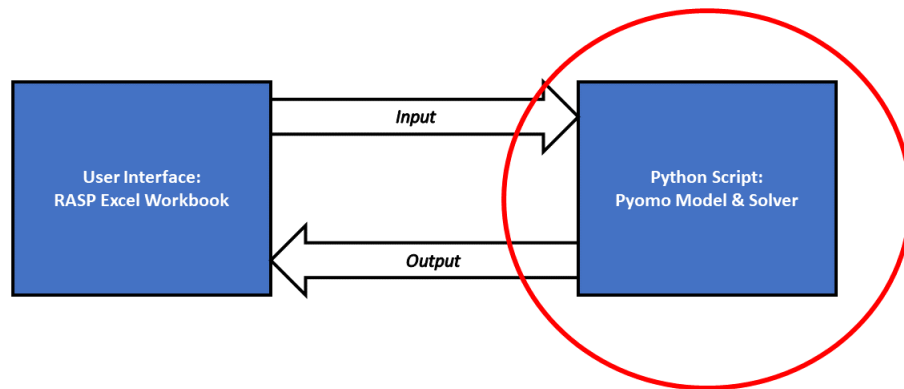


Figure 3.4. The file flow for solving a scenario with RASP begins with the Excel user interface. Once the user sets parameters and solver settings, the workbook generates an input file for the Python optimization script. A commercial solver then generates a solution for the Pyomo model based on the formulation from section 3.2 where the output file populates the RASP workbook with the results.

3.3.1 Input & Output Files

The input file provides the three requirements for the Python optimization script to execute. First, the input file provides the rolling horizon configuration by specifying the window and advance sizes. Second, the input file indicates the commercial solver and optimality settings such as the relative optimality gap and time limits. Lastly, the input file provides the information to populate the necessary sets, derived sets, dictionaries, and parameters to build the model.

The output text file provides the detailed replenishment plan following the execution of the Python optimization script. Table 3.1 lists the sections of the output file with corresponding descriptions.

Table 3.1. Sections of the RASP Output File

Section	Description
SUMMARY	Summary statistics for model outputs and replenishment plan
LOAD	Load events by provider with commodity quantities
RAS	UNREP events by customer with shuttle providers and commodity quantities
INREP	INREP events by customer with vendor providers and commodity quantities
UNABLE	Missed customer events with requested commodity quantities
SAWTOOTH	Customer day-by-day commodity inventory levels
STORAGE	Shuttle day-by-day commodity inventory levels

3.3.2 Batch Output

To test multiple rolling horizon configurations of windows and advances for a single scenario, we use a “batch runner” script to generate individual output files and a consolidated summary file. The batch runner accepts a comma-separated values (CSV) file listing window configurations and then sequentially executes the Python optimization script for each rolling horizon configuration. The batch runner generates individual output files for each

configuration as described in section 3.3.1 and consolidates all the results by appending the summary statistics to a single CSV file. The summary CSV file then provides the necessary data for further analysis and plot generation concerning objective function values, computation time, threshold violation rates, and provider plan costs. Figure 3.5 provides a screenshot of a cleaned batch output used for further analysis.

	A	B	C	D	E	F	G	K	L	M	O
1	Window	Advance	Obj Value	Objective Gap %	NumSolves	Time (s)	Time (m)	Below Threshold Violat	Total Possi	Below Threshold Violatio	Provider LOAD E
2	10	10	-83862.13654	0	6	87.339	1.45565	294	840	35	
3	10	7	-55457.59857	0	9	125.724	2.0954	290	840	34.52380952	
4	10	5	-68018.42126	0	11	155.178	2.5863	270	840	32.14285714	
5	14	14	-65185.59981	0	5	75.151	1.252516667	272	840	32.38095238	
6	14	12	-68886.95639	0	5	76.465	1.274416667	279	840	33.21428571	
7	14	10	-65185.59981	0.000437173	6	94.78	1.579666667	272	840	32.38095238	
8	14	7	-45504.69954	0	8	125.331	2.08885	224	840	26.66666667	
9	14	5	-78529.0403	0	11	176.436	2.9406	258	840	30.71428571	
10	15	15	-98304.6031	0.009493437	4	77.16	1.286	287	840	34.16666667	
11	15	14	-100272.1191	0	5	81.072	1.3512	294	840	35	
12	15	12	-68886.95639	0	5	77.494	1.291566667	279	840	33.21428571	
13	15	10	-98304.49476	0.001453968	6	98.217	1.63695	287	840	34.16666667	
14	15	7	-79163.58587	0	8	131.716	2.195266667	245	840	29.16666667	
15	15	5	-67550.37187	0.004857472	10	169.741	2.829016667	239	840	28.45238095	
16	20	20	-49459.045	0.009303808	3	91.755	1.52925	231	840	27.5	
17	20	15	-67706.3873	0	4	116.675	1.944583333	249	840	29.64285714	
18	20	14	-67550.37187	0	4	115.616	1.926933333	239	840	28.45238095	
19	20	12	-87128.04736	0.001246272	5	125.632	2.093866667	286	840	34.04761905	
20	20	10	-67550.37187	0.009319255	5	139.344	2.3224	242	840	28.80952381	
21	20	7	-67550.37187	0	7	209.362	3.489366667	239	840	28.45238095	
22	20	5	-67550.37187	0.009319255	9	237.017	3.950283333	242	840	28.80952381	

Figure 3.5. The batch output consolidates the summary statistics of multiple rolling horizon configurations for a given scenario. With this data, the analysis examines how various configurations affect optimality and computation time.

3.4 Warm Start

This thesis explores the use of a warm start approach wherein the monolith model is initialized using the rolling horizon's solution. Warm starts for a MILP “accelerate solution of a problem instance by taking advantage of information obtained during solution of a related instance” (Ralphs and Güzelsoy 2006). Regarding the warm start application for RASP, the rolling horizon solution provides a jumpstart for the monolith model, which then provides a final polish in the time remaining. Figure 3.6 illustrates the process flow for the rolling horizon implementation of RASP with an optional warm start.

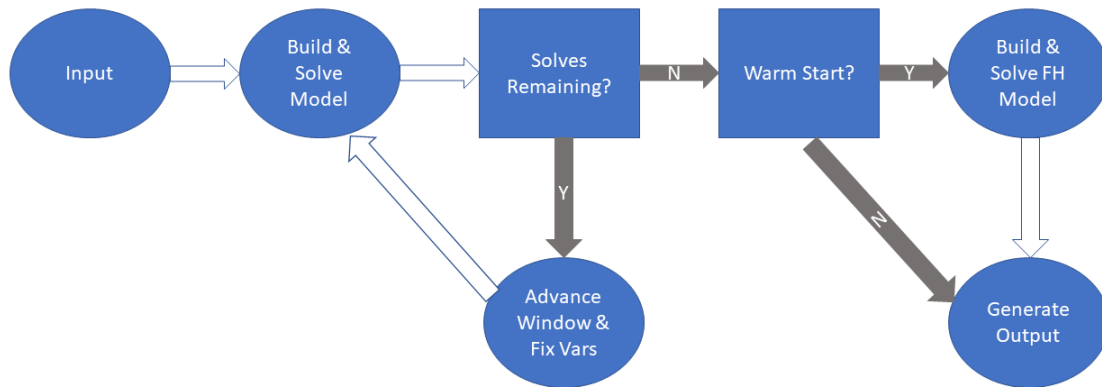


Figure 3.6. The process flow for this implementation begins with the iterations of the rolling horizon. If the warm start option is active, the final solution of the rolling horizon is used as an initial solution for the monolith model across the full horizon.

CHAPTER 4:

Experimentation and Results

This chapter describes our computational experiments. We first discuss the methodology, measures of effectiveness, hardware, and software used to execute the experiments and follow-on analysis. We then examine four different scenarios with varying complexity to compare rolling horizon configurations against the monolith model. In addition, the final two experiments employ rolling horizon solutions to warm start the monolith model with stricter time limits. The appendix contains the tabulated results from this chapter.

4.1 Methodology

In our analysis, the primary measures of effectiveness are the costs expressed in the objective function (in \$K) and computation times. Given four different scenarios with varying complexity regarding planning horizon, geographic restrictions, and unit composition, we solve these instances using multiple rolling horizon configurations then compare the results to those of the monolith model. In addition to this base analysis, we also examine solution quality via sawtooth graphs, solution treemaps, and trade-off plots. Lastly, we evaluate the efficacy of using a rolling horizon configuration to warm start the monolith model by comparing absolute optimality gaps under a strict computation time limit.

For this thesis, we make three assumptions regarding our model. First, both costs and consumption rates are accurate. Next, penalty and reward rates accurately represent the desires of the stakeholder or decision maker. Lastly, we assume the RASP workbook is generating the necessary arcs for this network problem. Regarding limitations, we constrain our analysis to these four scenarios, focusing on the efficacy of the optimization script and not the other features involved with the RASP workbook. Furthermore, although the optimization script can use multiple commercial packages to solve the MILP, we utilize only one solver.

4.2 Hardware and Software

We solve these models using a computer located within NPS’s optimization laboratory, informally known as the “Voodoo Lounge.” The computer operates in a 64-bit, Windows 10 operating environment using a 2.10 GHz Intel Xeon Gold 6230 processor with 128 GB random access memory (RAM) available.

For software, the Python 3.7 environment pairs with the Pyomo 5.7.1 optimization modeling language (Bynum et al. 2021). This thesis employs IBM’s CPLEX 12.10 as its solver. In addition to this optimization software, we use Plotly (Plotly Technologies Inc. 2015) and Microsoft Excel to create the visualizations in this chapter.

4.3 Scenario 1: Pacific Transit with Zone Restriction

Our first scenario covers a 60-day period with two carrier strike groups (CSGs) transiting the Pacific. Specifically, this scenario has eight customers, three shuttles, and seven vendors with two commodities. To simplify the complexity of the MILP and facilitate faster solve times, we first set geographic zone restrictions or “tethers” on the shuttles. This type of restriction has been made in practice in order to reduce solve times for large or complex instances using the legacy RASP tool. The resulting MILP associated with the monolith model contains 12,569 variables (of which 7,196 are binary), 5,376 arcs, and 9,785 linear constraints. In addition to the monolith, we test 59 rolling horizon configurations as shown in Table 4.1 and do not impose a computational time-limit.

Table 4.1. Scenario 1 configurations

Window (Days)	Advances (Days)	Subcount
10	5, 7, 10	3
14	5, 7, 10, 12, 14	5
15	5, 7, 10, 12, 14, 15	6
20	5, 7, 10, 12, 14, 15, 20	7
21	5, 7, 10, 12, 14, 15, 20, 21	8
24	5, 7, 10, 12, 14, 15, 20, 21, 24	9
28	5, 7, 10, 12, 14, 15, 20, 21, 24, 28	10
30	5, 7, 10, 12, 14, 15, 20, 21, 24, 28, 30	11

Our analysis begins with comparing objective value against time as captured in Figure 4.1. With the monolith model solving in 12.9 minutes and achieving an objective value of –\$45,505K, 25 rolling horizon configurations match the monolith’s optimal value in less time. The fastest rolling horizon configuration that also matches the monolith’s solution has a 21-day window and advance, solving in 1.1 minutes. The remaining 34 rolling horizon configurations do not achieve as high a quality of solution; however, these configurations solve in less than half the time of the monolith.

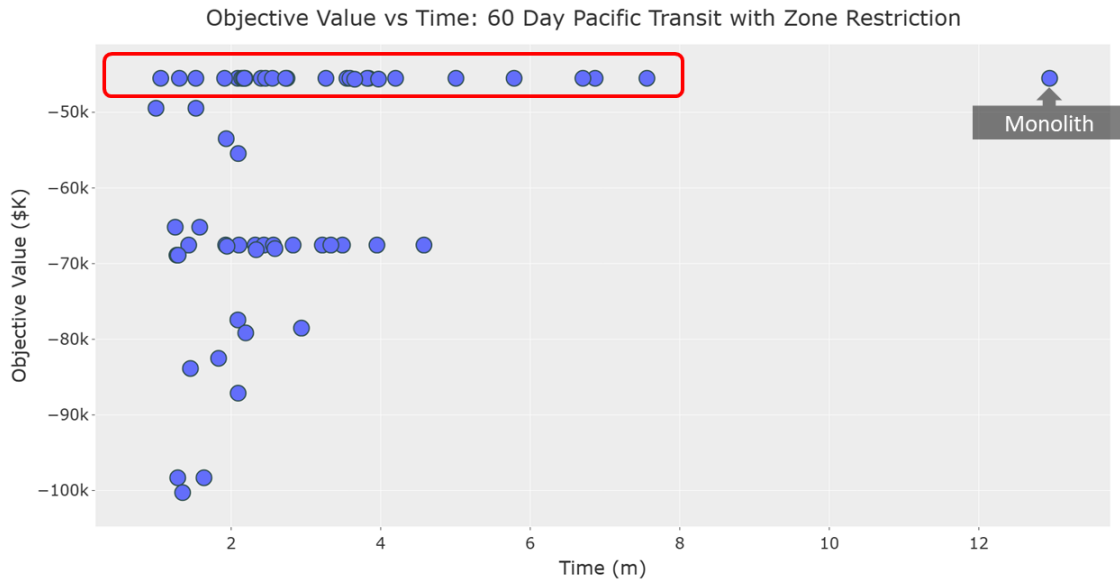


Figure 4.1. As captured by the points within the red boundary, 25 configurations achieve the same optimal solution as the monolith in less time. Although the remaining 34 configurations have worse objective values, they solve in less than half the time of the monolith model.

Figure 4.2 visualizes how various configurations perform in this scenario. The coordinates of the bubbles represent the applicable window (vertical axis) and advance (horizontal axis) for the rolling horizon configuration, while the bubble size represents the computation time (higher computation times result in larger bubbles). With the shade of the bubbles representing the objective values, small, purple configurations are ideal, representing high-quality solutions achieved in minimal time, while large, yellow configurations are undesirable signifying low-quality solutions with large computation times. In general, we observe that larger windows with smaller advances achieve higher-quality solutions while smaller windows with larger advances attain lower-quality solutions. Increasing window size increases computation time since the MILP is larger, while decreasing the advance increases computation time as the model requires more iterations to solve the scenario.

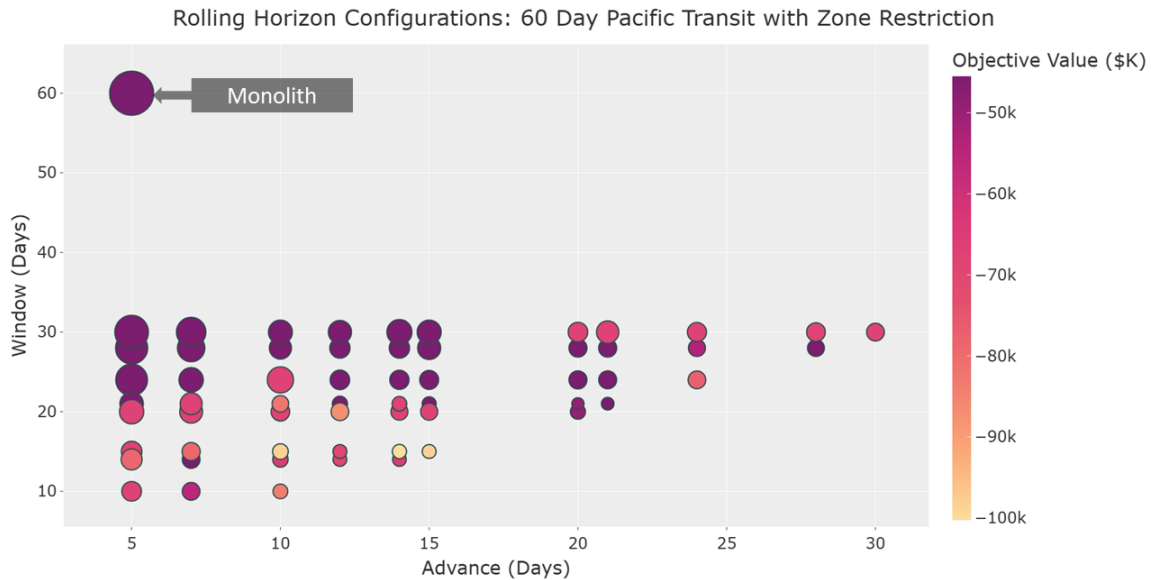


Figure 4.2. With bubble size indicating computation time and bubble shade representing objective values, this plot visualizes how different configurations perform in terms of solution quality and computation time. Small, purple configurations are ideal, while large, yellow configurations are undesirable.

To examine the effectiveness of a replenishment plan, we develop a sawtooth graph plotting a commodity's inventory percentage as the scenario progresses. The model invokes penalties whenever a customer's commodity inventory drops below a safety threshold of 60% and an extremis threshold of 20%. Using the optimal solution obtained from the monolith, Figure 4.3 plots the diesel fuel marine (DFM) inventory level for both CSGs. Although the geographic zone restriction enables faster solution times, the optimal solution fails to provide a practical solution. In this 60-day scenario, the escorts of the CSGs fall into extremis four times, reaching a minimum of 4% DFM. With such low quantities of fuel on board, schedulers cannot expect to use this plan at the operational level.

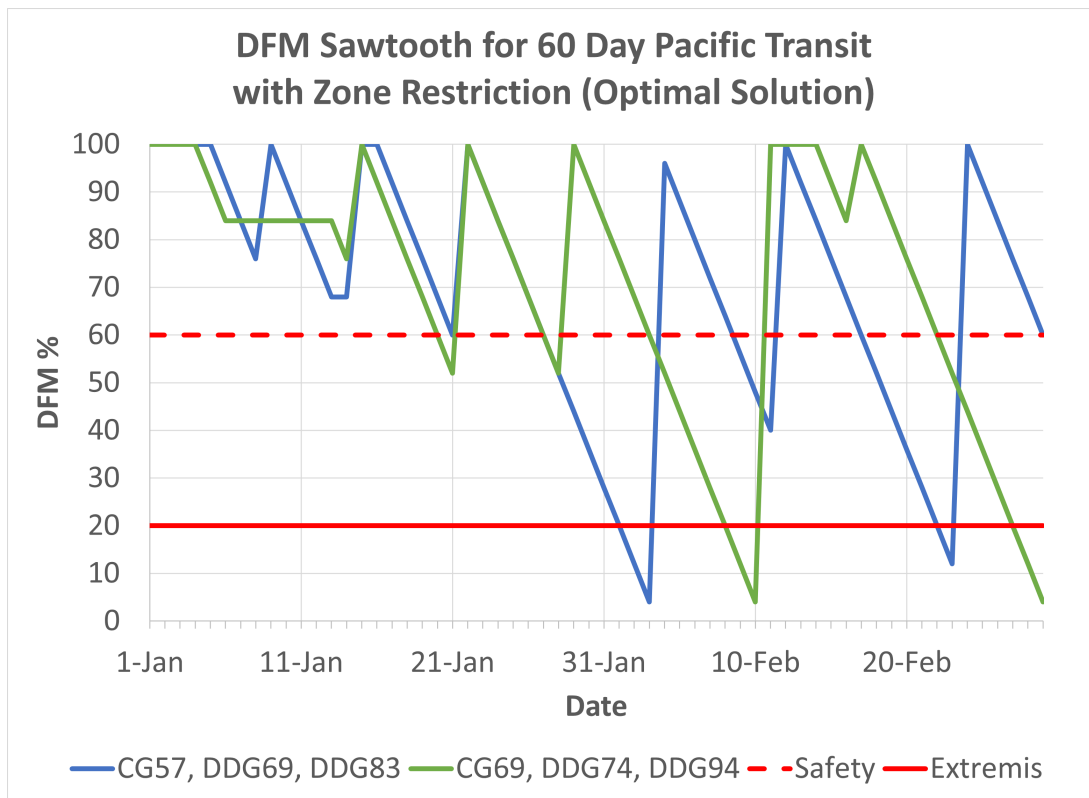


Figure 4.3. A sawtooth graph showing DFM inventory levels for the CSG escorts. Although RASP achieves optimality, the zone restrictions result in poor management of customer fuel levels and prevent practical use at the operational level.

To further investigate the quality of the replenishment plan, we develop solution treemaps. We create three groups based on the objective function (3.1) terms consisting of the rewards, effectiveness, and efficiency. Effectiveness includes the penalties associated with threshold violations and scheduling, while efficiency contains the costs and expenses associated with commodities, routing and events. The remaining terms of the objective function 3.1 are the reward terms associated with UNREPs and INREPs. For an ideal solution, we would observe large rewards with small effectiveness penalties and efficiency costs. As shown in Figure 4.4, the optimal solution for this scenario is not effective as the large threshold penalties dominate the solution.

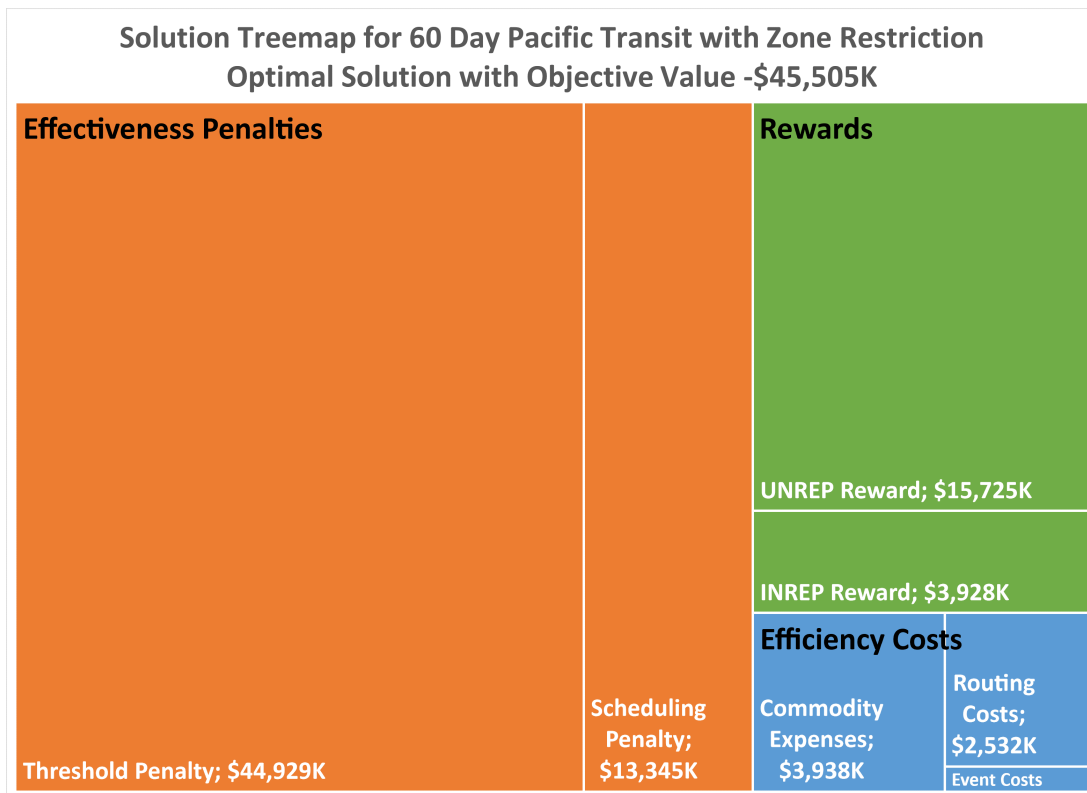


Figure 4.4. By grouping together terms in the objective function, we see the threshold violations dominate the optimal solution in this scenario. Although this replenishment plan gains efficiency by managing costs and expenses, a quality solution requires managing penalties associated with effectiveness.

4.4 Scenario 2: Pacific Transit without Zone Restriction

Our second scenario is nearly identical to the one described in Section 4.3 with eight customers, three shuttles, seven vendors, and two commodities covering a 60-day period. Whereas the first scenario restricts shuttles to geographic zones, the second scenario removes this restriction. This update increases both the number of arcs available for shuttles and the complexity of the problem. Considering the monolith, the amended MILP contains 36,207 variables (of which 28,155 are binary), 24,899 arcs, and 14,743 linear constraints. We compare the monolith to the same 59 rolling horizon configurations as in the previous experiment and impose a 48-hour computational time limit.

Once again, we first visualize the results by plotting the objective value against time as shown in Figure 4.5. Using the entire 48-hour time limit, the monolith model achieves a best integer solution of $-\$989.8\text{K}$ with a best bound of $-\$6,164.6\text{K}$, an absolute optimality gap of $-\$7,154.4\text{K}$. Of the 59 rolling horizon configurations tested, 34 configurations provide a better objective value in less time, ranging from 5.4 to 71.2 minutes. Although the remaining 25 rolling horizon configurations do not achieve as high solution quality, these configurations solve faster than the monolith model, ranging from 2.5 to 22.1 minutes.

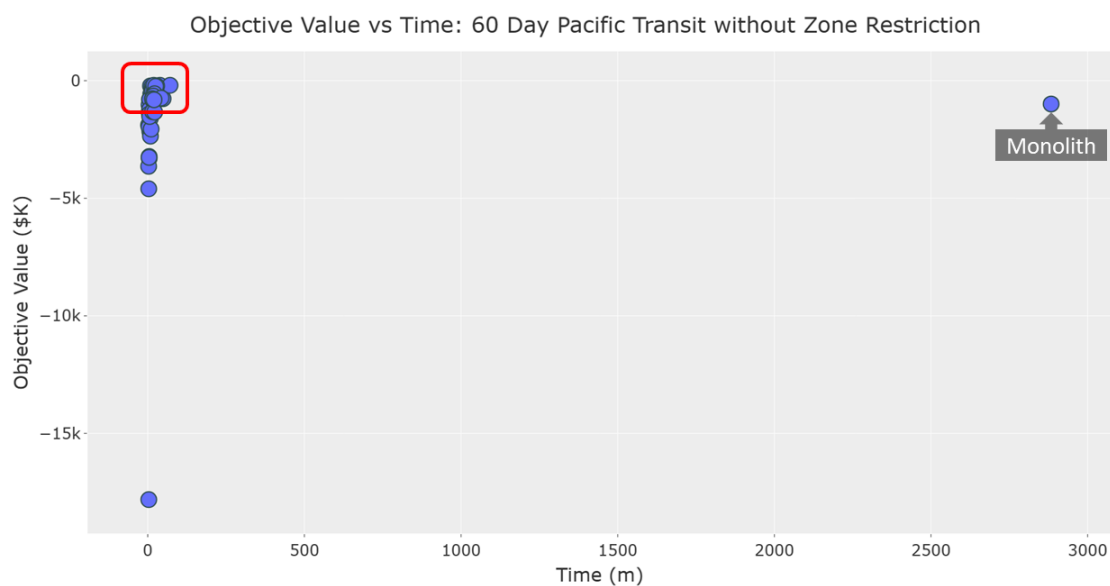


Figure 4.5. As captured by the points within the red boundary, we observe that 34 of the 59 rolling configurations achieve a better solution in less time than the monolith, which terminates after reaching the 48-hour time limit. The remaining 24 configurations also solve in less time, with one visible outlier in terms of solution quality.

For this scenario, configurations with larger windows provide higher quality solutions than their counterparts with smaller windows. Although larger windows increase computation time, these configurations are more robust to larger advances. Completing in 38.9 minutes, the configuration with a window of 30 days and advance of 7 days achieves the best objective value of $-\$194.8\text{K}$. Conversely, the configuration with a window and advance of 15 days achieves the worst objective value of $-\$17,815.2\text{K}$ in 3.3 minutes.

Figures 4.6 and 4.7 capture configuration performance with respect to computation time (bubble size) and objective value (bubble shade).

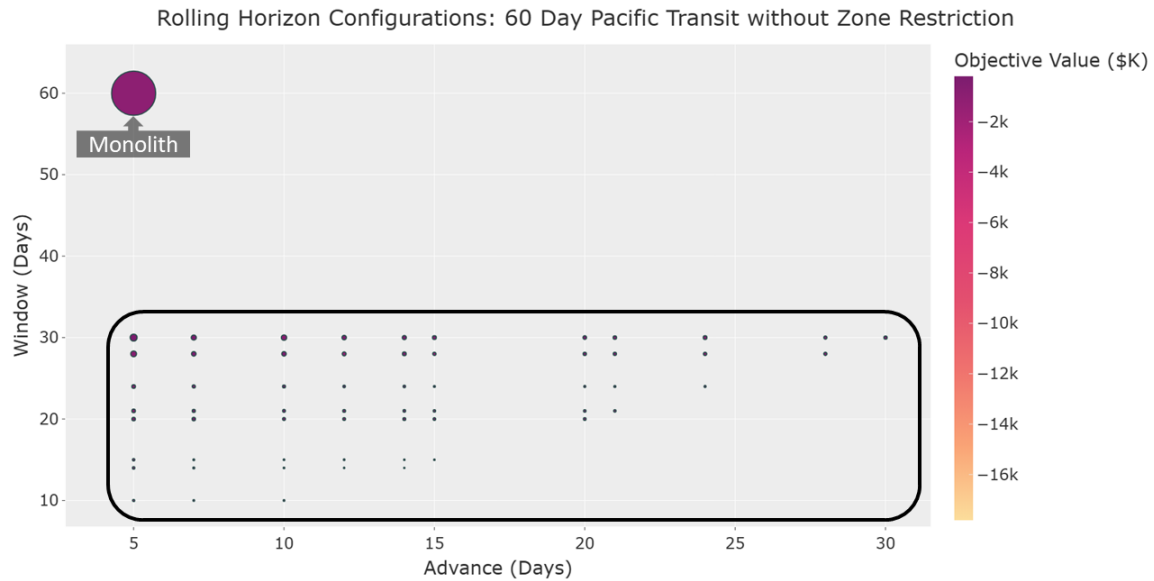


Figure 4.6. Although the monolith achieves a high-quality solution, its 48-hour computation time dwarfs that of the rolling horizon configurations. To further examine solution quality of the rolling horizon configurations, we remove the monolith model in Figure 4.7 while maintaining the same objective value scale.

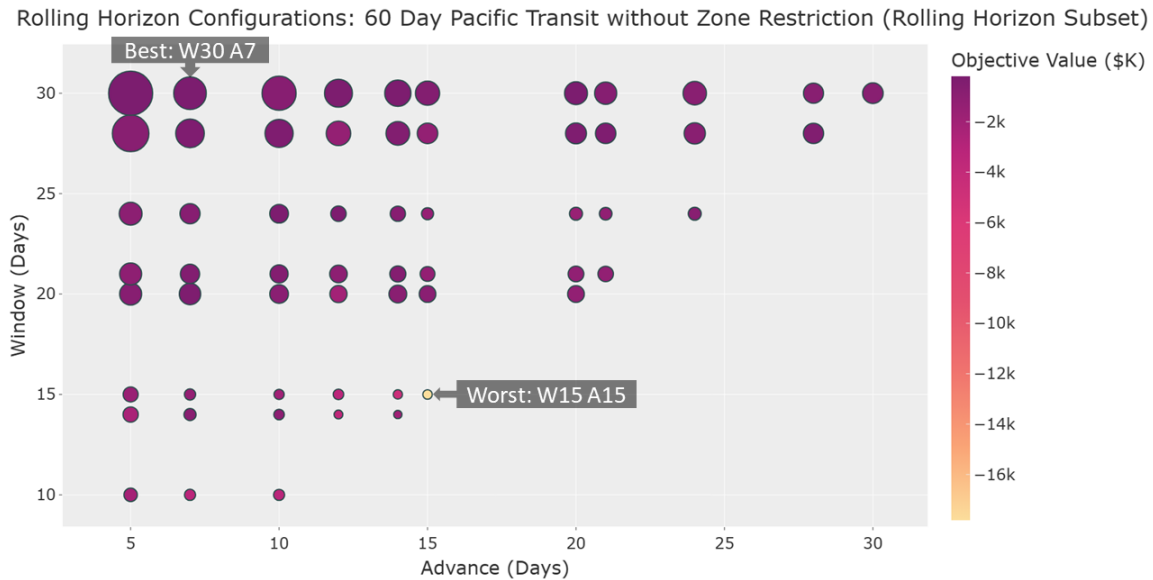


Figure 4.7. With bubble size re-scaled due to the monolith model's removal, we observe that all but one rolling horizon configuration achieve high quality solutions. The low-performing configuration lacks an overlap, meaning the advance is equal to the window.

To assess the effectiveness of the replenishment plan, we again create a sawtooth graph plotting the DFM inventory level for the CSG escorts in this scenario as shown in figure 4.8. Recall that for the zone-restricted scenario described in Section 4.3, the replenishment plan is not practical since the escorts fall to 4% fuel levels as shown in Figure 4.3. By removing the zone restrictions and implementing a rolling horizon, RASP not only prevents threshold violations for this scenario, but it also achieves these solutions in less than an hour. With its high solution quality and low computation time, the rolling horizon implementation of RASP provides plans that could be used at the operational level to schedule replenishments.

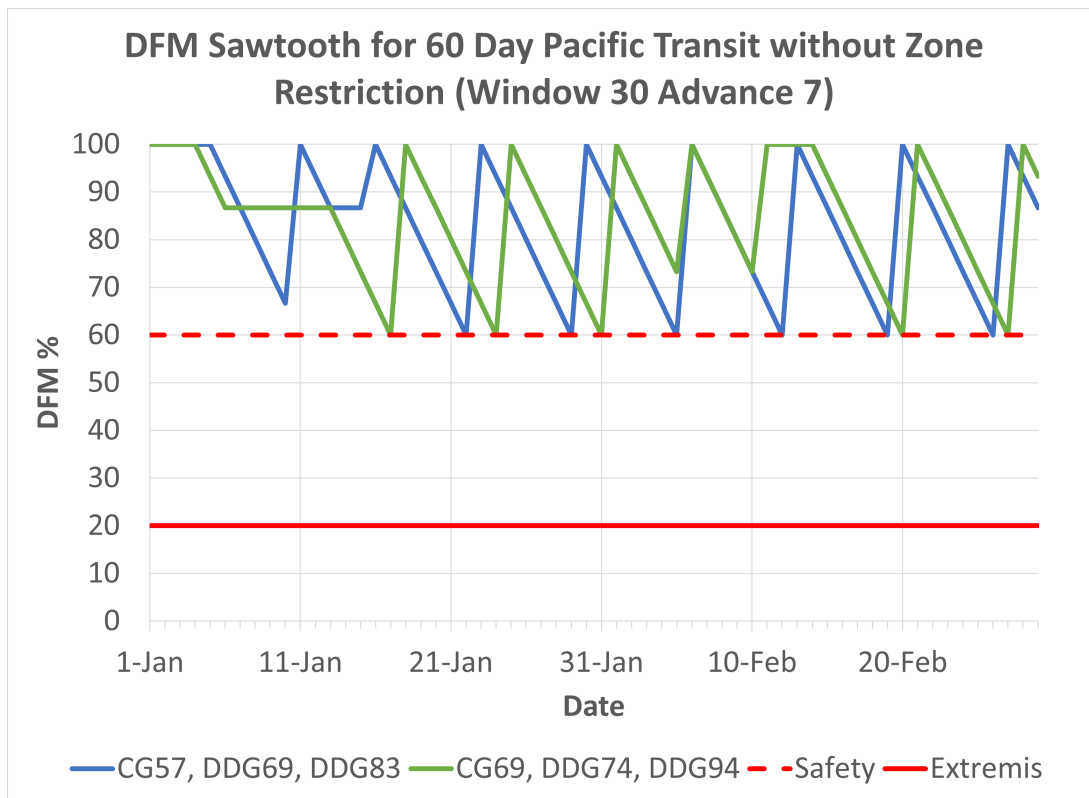


Figure 4.8. Using a window of 30 days and advance of 7 days, RASP achieves a more effective replenishment plan compared to that shown in Figure 4.3. With the removal of zone restrictions for shuttles, we observe the CSG escorts avoid violating the safety threshold for DFM.

To further examine the quality of the solution, we construct a solution treemap. As discussed in Section 4.3, imposing zone restrictions on shuttles results in the inability to adequately resupply customers, causing significant threshold violations. Figure 4.9 shows that threshold penalties are no longer present when zone restrictions are removed. With the rewards nearly equal in size to the efficiency costs and effectiveness penalties. After removing the zone restrictions, RASP develops an effective and efficient plan, resulting in an objective value improvement of \$45 million compared to the zone-restricted scenario.

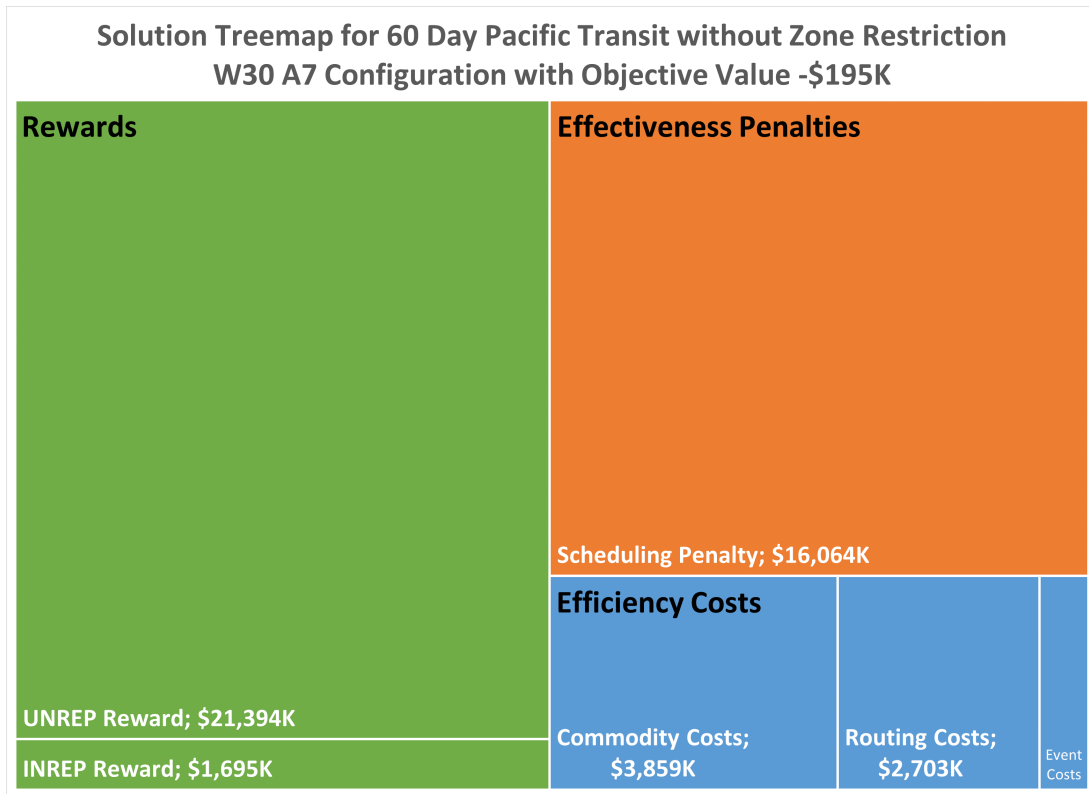


Figure 4.9. Whereas the penalties dominate the solution to the previous scenario as shown in Figure 4.4, this solution using a window of 30 days and advance of 7 days results in rewards nearly matching the effectiveness penalties and efficiency costs. By removing the zone restrictions for the shuttles, the objective value improves by \$45 million in this 60-day scenario.

4.5 Scenario 3: 7th Fleet Operations

Our third scenario covers a 60-day period simulating operations in the US Navy's 7th Fleet operating area in the western Pacific, spanning 124 million square kilometers and 36 maritime countries (United States 7th Fleet 2023). Specifically, this scenario has eighteen customers, three shuttles, and twelve vendors with four commodities in use. We do not impose zone restrictions in this scenario. This lack of zone restriction, coupled with the increased numbers of ships, vendors, and commodities, creates a more complex model than the previous scenarios. The resulting MILP contains 86,381 variables (of which 60,029 are binary), 54,283 arcs, and 39,974 linear constraints.

Including the monolith model, this thesis tests 39 configurations as shown in Table 4.2 and enforces a 24-hour computational time limit.

Table 4.2. Scenario 2 configurations

Window (Days)	Advances (Days)	Subcount
10	10, 7	2
14	14, 12, 10, 7	4
15	15, 14, 12, 10	4
20	20, 15, 14, 12, 10	5
21	21, 20, 15, 14, 12	5
24	24, 21, 20, 15, 14, 12	6
28	28, 24, 21, 20, 15, 14	6
30	30, 28, 24, 21, 20, 15	6

To begin our analysis for this scenario, we visualize the results by plotting the objective value against time as shown in Figure 4.10. Using the entire 24-hour time limit, the monolith model achieves a best integer solution of $-\$25,843\text{K}$ with a best bound of $-\$10,537\text{K}$, an absolute optimality gap of $-\$15,306\text{K}$. Of the 38 rolling horizon configurations tested, 28 configurations provide a better objective value in less than or equal time. Although the remaining 10 rolling horizon configurations do not achieve the monolith's solution quality, these configurations solve faster than the monolith model, ranging from 0.8 to 12.4 hours.

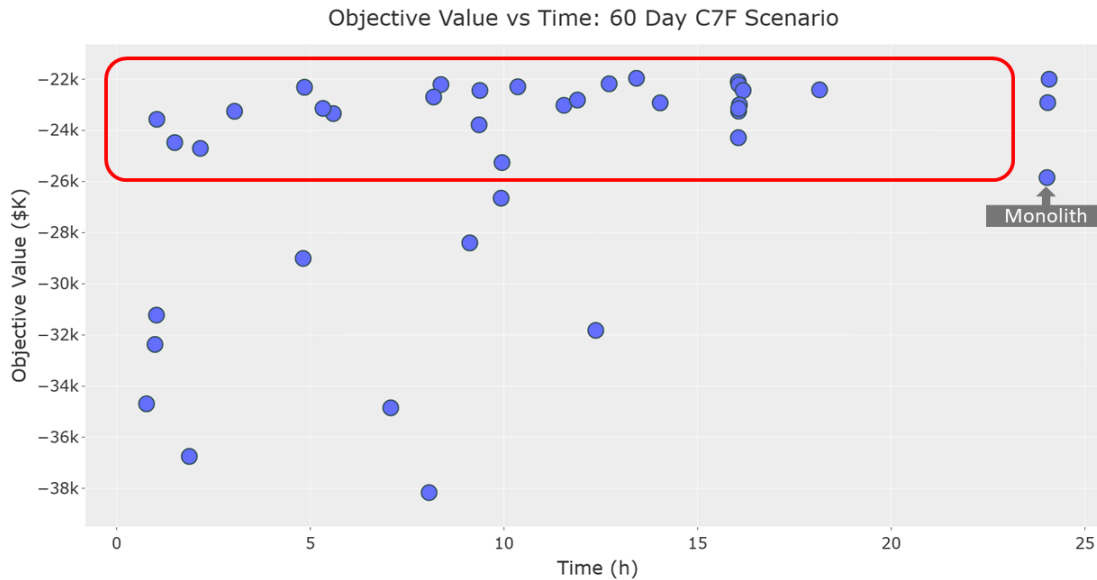


Figure 4.10. As captured by the points within the red boundary, we observe that 26 of the 38 rolling configurations achieve a higher-quality solution in less time than the monolith model achieves under a 24-hour time limit. Two additional rolling horizon configurations achieve higher objective values, but these configurations require 24 hours to execute. The remaining 10 configurations solve in less time than the monolith, but they do not achieve as high of an objective value.

Comparable to the last scenario, configurations with larger windows provide higher-quality solutions than their counterparts with smaller windows; nonetheless, these larger windows increase the computation time. All twelve rolling horizon configurations with windows of at least 28 days provide a better objective value than the monolith, but two of these configurations use the entirety of the 24-hour time allotment. Completing in 13.4 hours, the configuration with a window of 21 days and advance of 12 days achieves the best objective value of $-\$21,960\text{K}$. Conversely, the configuration with a window and advance of 21 days achieves the worst objective value of $-\$38,166.1\text{K}$ in 8.1 hours. Figures 4.11 and 4.12 capture configuration performance with respect to computation time (bubble size) and objective value (bubble shade).

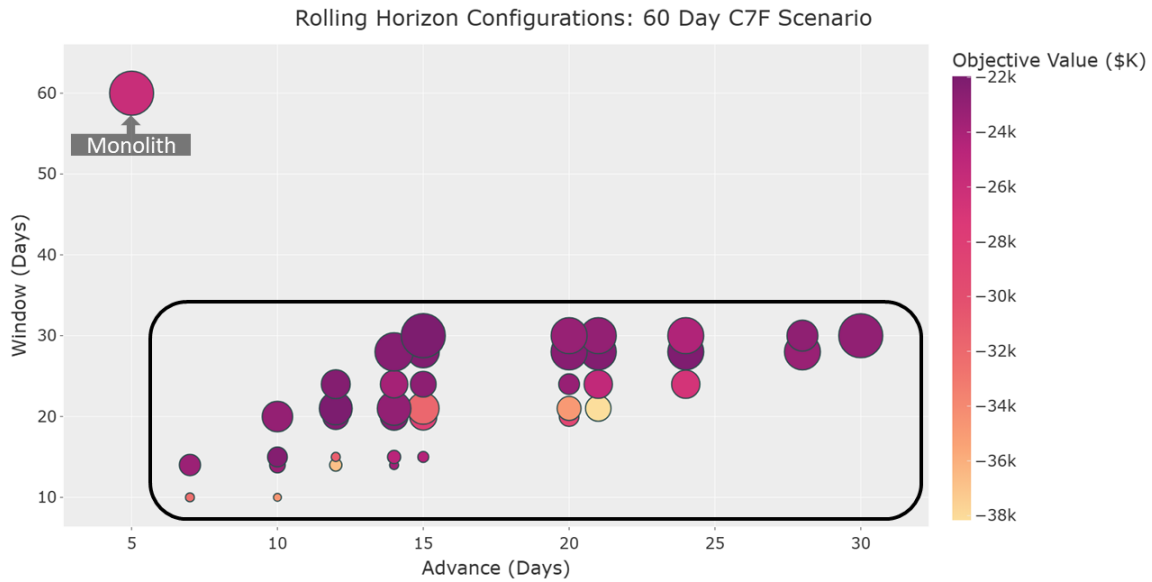


Figure 4.11. The monolith obtains an inferior objective value than most of the rolling horizon configurations, while utilizing the entire 24-hour computation time limit. To better examine individual configuration performance, we remove the monolith from Figure 4.12 while maintaining the same time and objective value scales.

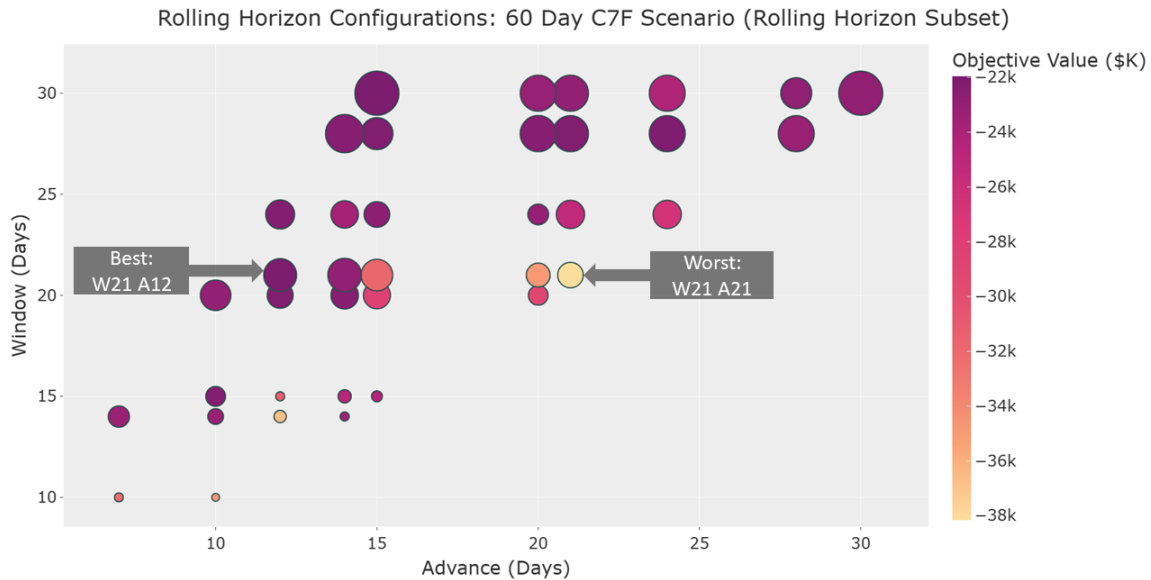


Figure 4.12. Compared to smaller windows, larger windows generally achieve higher objective values and are more robust to larger advances, but at the cost of increased computation time. Smaller windows can achieve higher objective values by decreasing the advance, which increases the window overlap between iterations.

To further explore the monolith model’s solution quality compared to the rolling horizon configurations, we construct an efficiency vs effectiveness trade-off plot in Figure 4.13. Although not all-encompassing, threshold violations and provider plan costs to route shuttles are strong indicators of effectiveness and efficiency. On the vertical axis illustrating effectiveness, we plot the threshold violation rate which is the percentage of total ship-commodity-days that fell below the safety threshold. On the horizontal axis illustrating efficiency, we plot the provider plan cost for the scenario. Lastly, the bubble shade indicates the maximum severity of any threshold violation in the solution. Ideally, a yellow bubble in the lower left of the plot would indicate a plan with a low rate and severity of threshold violations because of efficiently routed shuttles. Conversely, a purple bubble in the upper right of the plot would indicate a plan with a high rate and severity of threshold violations because of inefficiently routed shuttles.

Figure 4.13 shows the monolith model as the least efficient plan in terms of shuttle routing while having moderately severe threshold violations at a moderate rate.

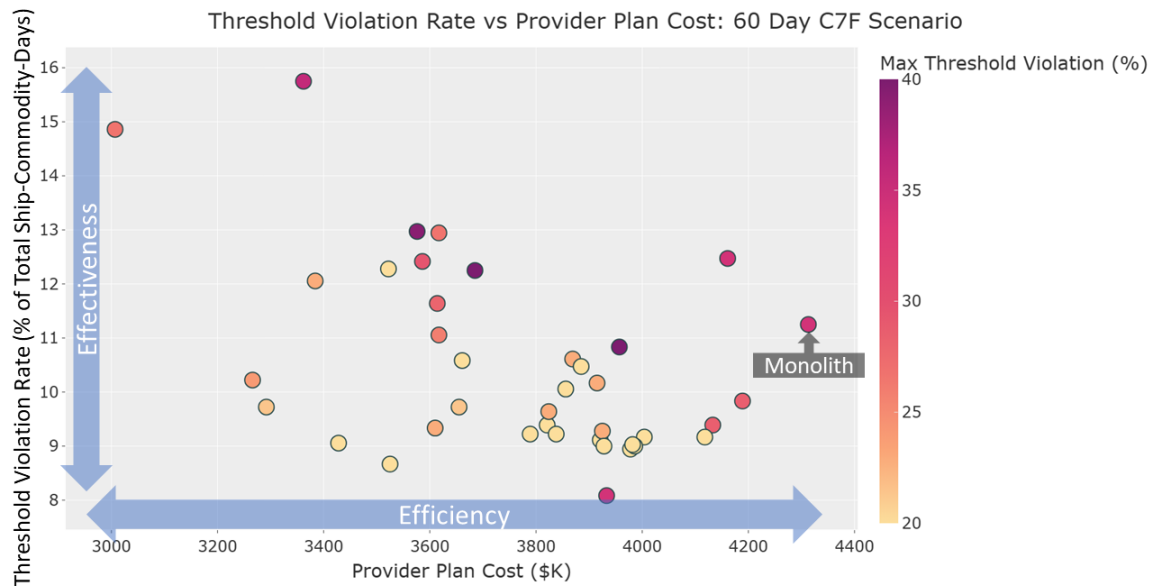


Figure 4.13. By plotting these indicators of effectiveness and efficiency, we observe the monolith has the least efficient shuttle routing of all the configurations tested. Since we observe many configurations achieve a lower rate of threshold violations at a lower cost to route shuttles, we understand how the 28 rolling horizon configurations achieve a higher objective value than the monolith.

4.5.1 Warm Starting the Monolith

We next transition to using a rolling horizon solution as a warm start for the monolith model. The goal of this experiment is to use the rolling horizon as a heuristic to provide a “good” initial solution, which the monolith can then polish. Ideally, this approach combines the computational efficiency of the rolling horizon approach with the guaranteed solution quality of the monolith. In addition to improving solution quality, the warm start approach provides a valid bound on the overall solution quality, while the rolling horizon process (alone) does not.

Our first warm start instance is that of Scenario 3, described in Section 4.5. We establish

a time limit of two hours for the overall process (rolling horizon and final monolith solve) and allocate no more than 80% of the total time limit to the rolling horizon algorithm. We divide this allocated time equally between each rolling horizon iteration. If a rolling horizon iteration reaches optimality before consuming its allocated time limit, we add its remaining time to the monolith model's portion. We test 10 rolling horizon configurations and utilize a 10% relative optimality gap for both the rolling horizon iterations and the monolith.

Using a window of 14 days and advance of 7 days, we first visualize how the objective value progresses during the solution process. With this configuration setting, RASP requires eight iterations to progress through the entire scenario as shown in Figure 4.14. After each iteration, we observe the objective value decreasing (worsening) as fewer and fewer integer decision variables are relaxed. In addition, we note that the second through fourth iterations consume their entire allocated iteration time while the other iterations do not. The rolling horizon portion terminates after 55 minutes with an objective value of $-\$24,633\text{K}$, which then allows RASP to dedicate the remaining 65 minutes to solution polishing using the monolith model. This yields a slight improvement in objective value with a best integer solution of $-\$24,103\text{K}$ and a best possible solution of $-\$9,544\text{K}$.

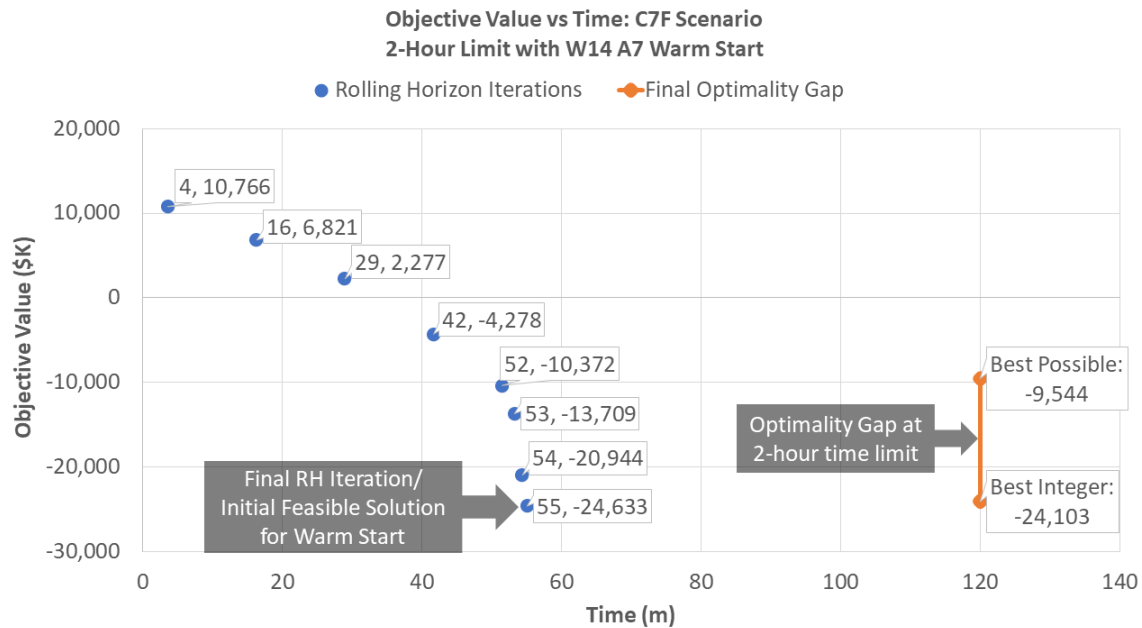


Figure 4.14. Objective value during the rolling horizon solution process. Since each successive rolling horizon iteration relaxes fewer integer decision variables, we observe the objective value decreasing as the rolling horizon algorithm progresses. Given an integer feasible solution obtained via rolling horizon, the monolith makes an objective value improvement of \$530K in the last 65 minutes of this experiment. Caption pairs indicate cumulative computation time and objective values following an iteration's termination.

By achieving significant improvements in quality, the warm start applications display better performance under tight time constraints. Comparing the optimality gaps for the monolith model and nine warm start applications as shown in Figure 4.15, we observe comparable best possible solutions with all configurations. However, the strict time limit of two hours has a significant effect on solution quality. The monolith (alone) achieves a $-\$46,663K$ objective value in this time span, while the warm start applications improve the best integer solution by \$15-22 million.

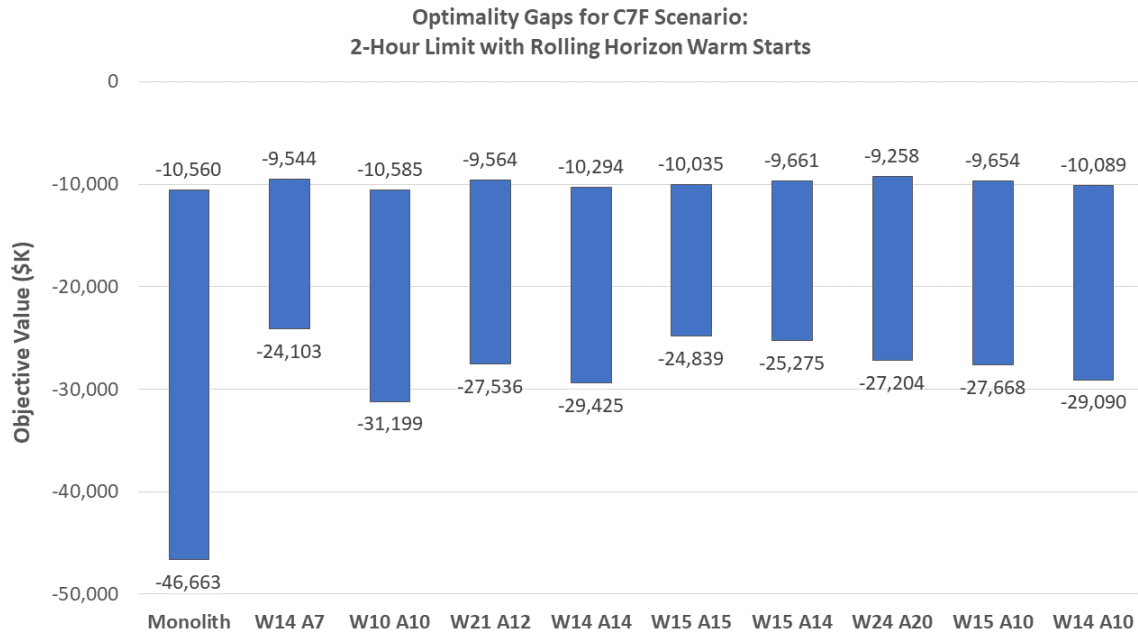


Figure 4.15. In this experiment, all configurations achieve similar best bounds (tops of bars); however, the warm start applications significantly improve the best integer solution (bottoms of bars) compared to the monolith model.

4.6 Scenario 4: 90-Day Replenishment Plan

To further evaluate the warm start approach, we construct a final scenario covering a 90-day period. This scenario utilizes three customers, six shuttles, two vendors, and four commodities. The resulting MILP contains 32,653 variables (of which 24,336 are binary), 21,006 arcs, and 17,221 linear constraints. Imposing time limits of one, two, three, and six hours, we solve this scenario using the warm start application with 174 different configurations of window and advance as shown in Table 4.3.

Table 4.3. Scenario 4 pivot table

Window (Days)	1 h Count	2 h Count	3 h Count	6 h Count	Window Subcount
10	3	3	3	2	11
12	3	3	3	0	9
14	4	4	4	4	16
15	4	4	4	3	15
20	5	5	5	5	20
21	5	5	5	4	19
24	6	6	6	5	23
28	6	6	6	6	24
30	6	6	6	3	21
45	3	3	3	3	12
Monolith	1	1	1	1	4
Time Subcount	46	46	46	36	174

We first visualize the results by plotting the objective value against time as shown in Figure 4.16. All configurations utilize the entire allotted time limit. As the time limit increases, we observe that the variance between objective values decreases. The monolith model shows no improvement between one- and two-hour time limits, resulting in an objective value of $-\$125,306\text{K}$ and customers with a negative inventory of 13%. For warm start applications with larger windows and constrained time, the rolling horizon heuristic may not provide a “good” feasible solution for the monolith polish, since iterations with larger windows risk carrying greater optimality gaps due to the higher count of binary variables. Conversely, configurations that solve quickly (smaller windows) are more consistent and better for the warm start application, in this experiment.

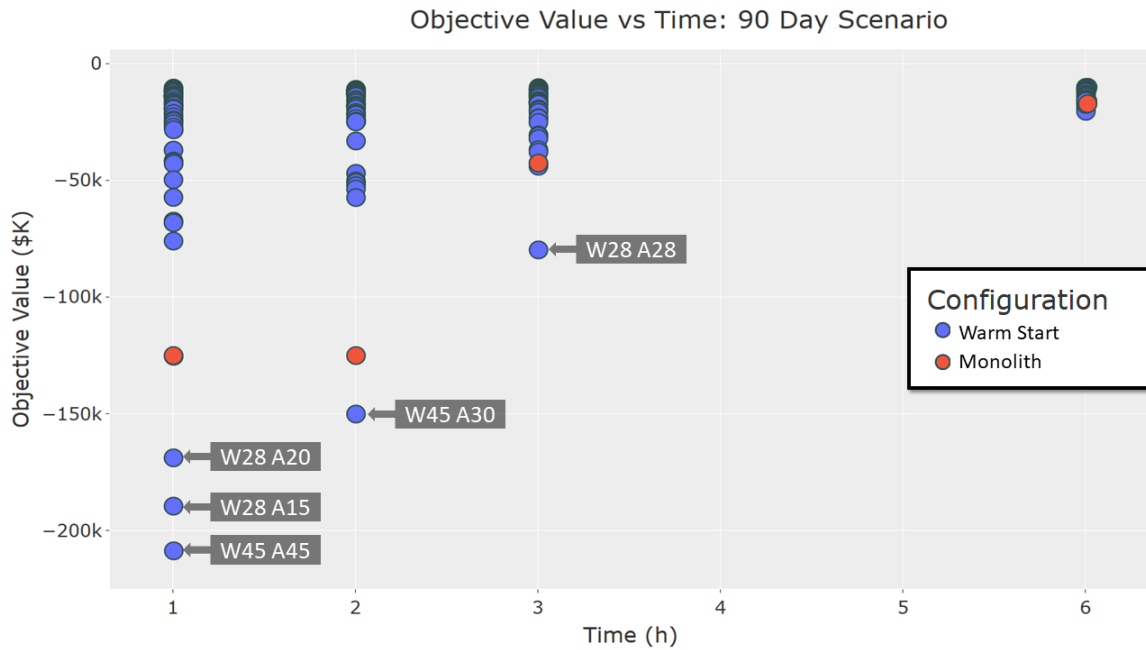


Figure 4.16. Although variance decreases as the computational time limit increases, the majority of warm start applications provide better solutions than the monolith model alone for all time limits.

Using the warm start, the majority of configurations outperform the monolith model as captured in Table 4.4. Given a six-hour limit, 89% of the configurations provide a higher objective value. With three hours or less, these percentages exceed 90%.

Table 4.4. Performance summary for 90-day scenario

Time Limit (h)	Rolling Horizon Configs	Rolling Horizon Configs Outperforming Monolith
6	35	31
3	45	42
2	45	44
1	45	41

As shown in Figure 4.17, the warm start applications provide a significant improvement over the monolith, especially with strictly constrained computation times. Whereas the monolith's objective value notably increases with a greater computation time limit, the majority of rolling horizon configurations achieve comparable or better objective values even under strict time limits, and minimally improve as the time limit increases. With a six-hour limit, the warm start provides a \$1.2 to \$7.0 million improvement over the monolith model. Decreasing to a three-hour limit, the warm start provides a \$5.7 million to \$32 million improvement. In two hours or less, the warm start provides a \$75 million to \$115 million improvement over for this 90-day replenishment plan.

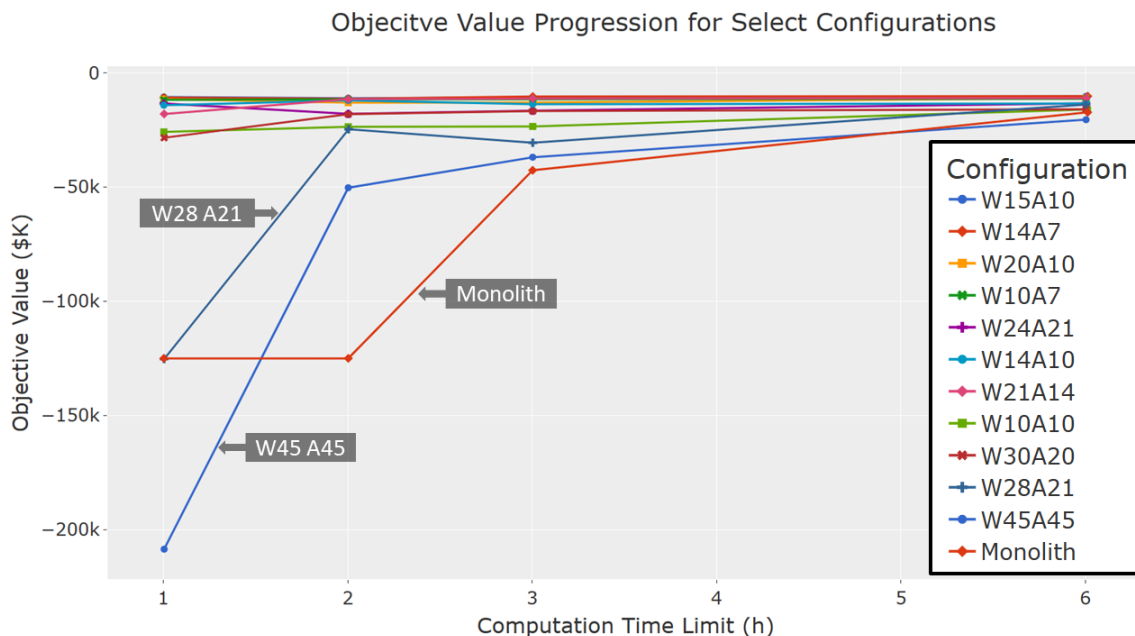


Figure 4.17. While the monolith model requires more time to progress toward a quality solution, warm starts using smaller windows achieve quality solutions faster.

4.6.1 Rolling Horizon vs Warm Start Application

To conclude our experiments, we compare the rolling horizon performance against the warm start application. Using the same 90-day scenario and solver settings, we create replenishment schedules with the rolling horizon approach under time limits of one, two, and three hours. With results from both the rolling horizon and warm start application,

we then compare the objective values of configurations grouped by computational time limits. Whereas the warm start applications use all the allotted time, the rolling horizon approaches do not. Without the warm start, the rolling horizon approach does not reallocate time saved for an iteration reaching optimality toward a monolith polish. The best warm start application given three hours is the W14A7 configuration, achieving an objective value of -\$10,289K with a best bound of \$700K. The best rolling horizon configuration given three hours or less is the W15A10 configuration, achieving an objective value of -\$10,470K in 1.3 hours. Table 4.5 summarizes the results across all three time limits.

Table 4.5. Rolling horizon vs warm start summary

Time Limit (h)	Configs Tested	RH Configs with Time Remaining	Obj Values WS > RH
1	45	37	27
2	45	37	27
3	45	38	29

Although the warm start application provides higher objective values compared to the rolling horizon approach for approximately 60% of the configurations in this scenario, we only observe two conclusive trends. First, with more time allocated, the objective value differences between the two methods decrease. Second, given a 3-hour computation limit, warm starts outperform rolling horizons for windows of 14 days or less. For two hours or less, neither heuristic consistently outperforms the other based on window size. Figures 4.18, 4.19, and 4.20 capture the objective value difference from performing a warm start compared to the rolling horizon approach.

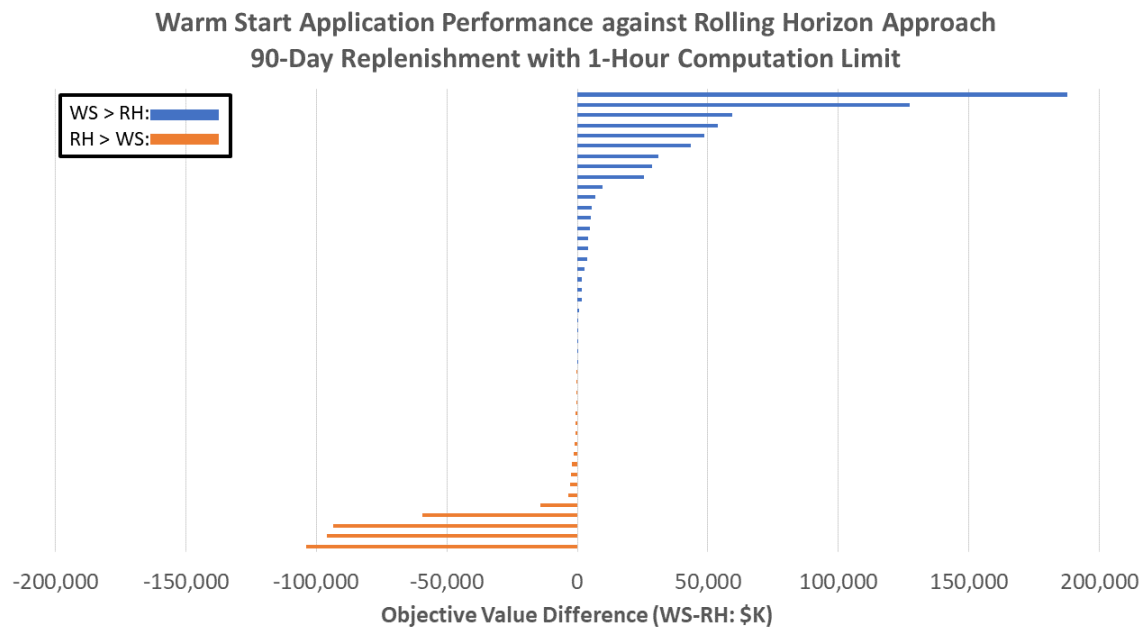


Figure 4.18. Tornado plot for 1-hour computation limit.

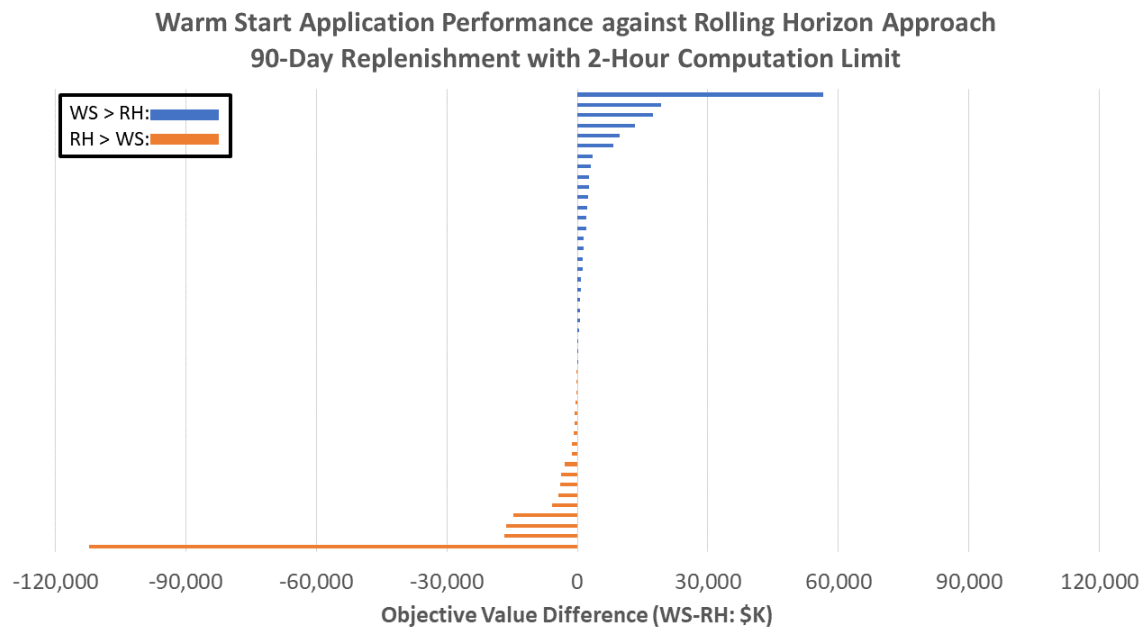


Figure 4.19. Tornado plot for 2-hour computation limit.

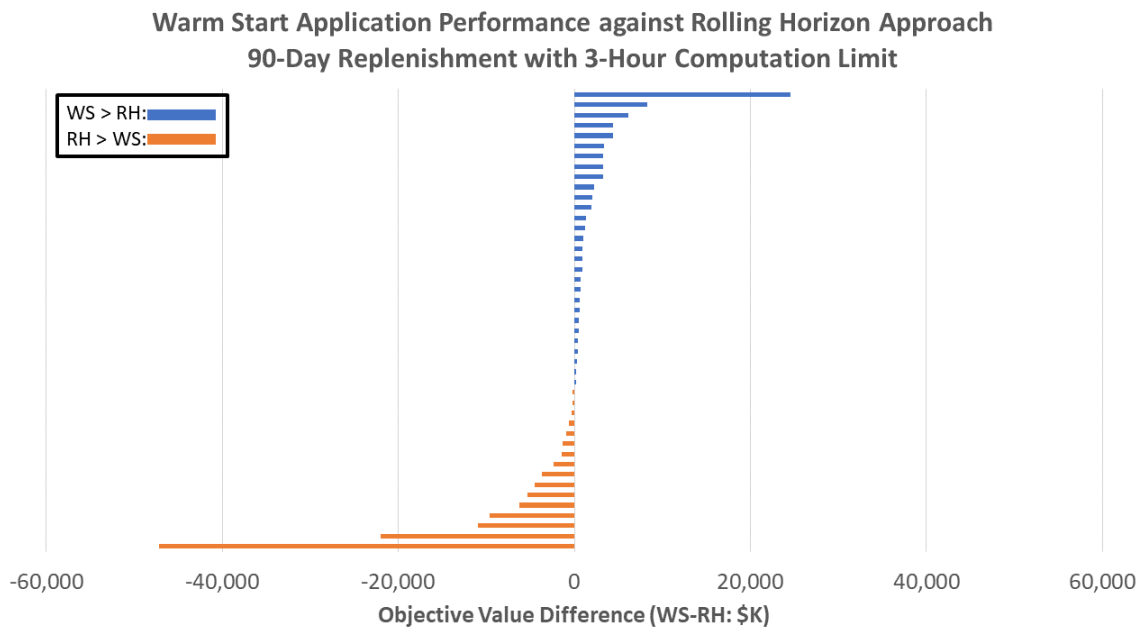


Figure 4.20. Tornado plot for 3-hour computation limit.

CHAPTER 5:

Conclusions and Future Work

This thesis explores the use of a rolling horizon structure with future relaxation in the RASP model in an attempt to obtain operational-quality solutions with reduced computation time. We evaluate multiple rolling horizon configurations across four scenarios varying in complexity based on time range and number of customers, shuttles, vendors, and commodities. Using objective value and computation time as the primary measures of effectiveness, we compare various rolling horizon configurations against the monolith model. Our experimental results support three primary conclusions:

1. The rolling horizon implementation is an effective tool for RASP, leading to operational-quality solutions in operationally-relevant time. Compared to the monolith model, we observe that not only do the rolling horizon configurations provide solutions faster, but most provide higher-quality solutions.
2. Imposing geographic restrictions on shuttles can prevent RASP from generating effective replenishment plans, leading to increased costs. This result echoes the findings of Doyle (2006), which found the global allocation of CLF units provides significant efficiency benefits over fleet ownership. While zone restrictions have previously been utilized in order to generate more computationally tractable instances for RASP, the computational efficiency of the rolling horizon approach renders these restrictions unnecessary.
3. The rolling horizon approach provides a high-quality solution that can be used to warm start the monolith model. The warm start approach can provide improved solution quality over the rolling horizon approach alone, as well as a valid bound on overall solution quality. Simulating time-constrained planning in the US 7th Fleet operating area, we estimate that the rolling horizon improves replenishment planning by \$15-22 million over the monolith model for a 60-day scenario.

With a faster model for use at the operational level, theaters can potentially increase combatant time-on-station, support more ships in-theater, and reduce operating costs for the CLF.

5.1 Future Work

The first opportunity for future work is to complete a more thorough analysis of rolling horizon performance based on scenario inputs, starting conditions, and rolling horizon configurations. This work could guide analysts in selecting the best rolling horizon configuration to create a replenishment plan. Our work does provide some initial guidance for using the rolling horizon tool. For instance, for the warm start application under strict time constraints, smaller windows with overlap outperform larger windows. With our findings, we recommend windows between 14 and 24 days with an advance smaller than the window. Specifically, the window equal to 14 days and an advance equal to 7 days is a robust configuration, consistently creating quality solutions for the scenarios we considered. However, a thorough analysis with a larger number of scenarios can provide more robust guidelines, making the rolling horizon implementation much more effective.

Another opportunity for additional work could implement a different cascade into the RASP model. Instead of focusing on the time horizon, a cascade could iterate over different sets. By iterating over geographic zones, commodities, and/or customers, the model could potentially achieve better quality solutions and/or reduce computation time. This new implementation can also develop solutions prioritizing specific customers, groups, or geographic areas.

An additional opportunity for future work could focus on force planning and tactics. RASP can provide quantifiable impacts on allocating more or fewer shuttles in a theater or the acquisition of new platforms, ranging from manned replenishment vessels to unmanned combatants. In the age of anti-access/area denial, additional research can also incorporate new tactics in the RASP model such as a consolidated cargo replenishment at sea (CONSOL) event where commercial tankers transfer fuel to a CLF shuttle. With the improved speed of RASP, research can test more scenarios with more variations in parameters.

Lastly, future work can incorporate persistence into the RASP model. Since RASP is an optimization model, small changes in input parameters can drastically change the optimal solution (Brown et al. 1997). In an operational setting, we expect RASP to be run repeatedly with updated input data, and with portions of the previous solution already promulgated to customers and vendors. While it may be possible to modify previously planned events, it is undesirable. For naval replenishments, cancelling a RAS after a vendor loads the supporting CLF unit has substantial impacts for the combatant. This schedule change delays

the receipt of critical repair parts and may result in the fiscal loss of perishable food stores. By penalizing near-term changes to the replenishment plan, the Navy could avoid these issues, saving additional costs and improving readiness.

THIS PAGE INTENTIONALLY LEFT BLANK

APPENDIX: Tabulated Results

This appendix contains the tabulated results for the tested scenarios. At a minimum, each table contains the objective value, computation time, violation rate, below threshold max percentage, and provider plan cost. Table observations are rolling horizon configurations (window and advance) or the monolith model.

Table A.1. Scenario 1 results (1-30)

Config	Objective Value (\$K)	Time (m)	Violation Rate %	Below Threshold Max %	LOAD Events	RAS Events	INREP Events	Provider Plan Cost (\$K)
W28A7	-45,505	5.0	27	56	6	36	8	2,532
W28A5	-45,505	6.9	27	56	6	36	8	2,532
W30A7	-45,505	5.8	27	56	6	36	8	2,532
W14A7	-45,505	2.1	27	56	6	36	8	2,532
W24A21	-45,505	2.1	27	56	6	36	8	2,532
W24A20	-45,505	2.2	27	56	6	36	8	2,532
W24A15	-45,505	2.4	27	56	6	36	8	2,532
W24A14	-45,505	2.5	27	56	6	36	8	2,532
W24A12	-45,505	2.6	27	56	6	36	8	2,532
W24A5	-45,505	6.7	27	56	6	36	8	2,532
W28A28	-45,505	1.9	27	56	6	36	8	2,532
W28A21	-45,505	2.2	27	56	6	36	8	2,532
W28A20	-45,505	2.2	27	56	6	36	8	2,532
W28A15	-45,505	3.5	27	56	6	36	8	2,532
W28A14	-45,505	2.7	27	56	6	36	8	2,532
W28A12	-45,505	2.7	27	56	6	36	8	2,532
W28A10	-45,505	3.3	27	56	6	36	8	2,532
W30A15	-45,505	3.8	27	56	6	36	8	2,532
W30A14	-45,505	4.2	27	56	6	36	8	2,532
W30A10	-45,505	3.8	27	56	6	36	8	2,532
W30A5	-45,505	7.6	27	56	6	36	8	2,532
Monolith	-45,505	12.9	27	56	6	36	8	2,532
W21A15	-45,505	1.3	27	56	6	36	8	2,532
W21A21	-45,505	1.1	27	56	6	36	8	2,532
W21A12	-45,505	1.5	27	56	6	36	8	2,532
W30A12	-45,505	3.6	27	56	6	36	8	2,532
W24A7	-45,614	4.0	27	56	6	36	8	2,641
W21A5	-45,627	3.7	27	56	6	36	8	2,654
W20A20	-49,459	1.5	28	56	6	36	8	2,533
W21A20	-49,459	1.0	28	56	6	36	8	2,533

Table A.2. Scenario 1 results (31-60)

Config	Objective Value (\$K)	Time (m)	Violation Rate %	Below Threshold Max %	LOAD Events	RAS Events	INREP Events	Provider Plan Cost (\$K)
W28A24	-53,489	1.9	29	64	6	40	8	2,684
W10A7	-55,458	2.1	35	64	5	32	8	2,652
W14A14	-65,186	1.3	32	72	5	32	8	2,288
W14A10	-65,186	1.6	32	72	5	32	8	2,288
W30A30	-67,550	2.1	29	80	6	36	8	2,366
W15A5	-67,550	2.8	28	80	6	36	8	2,366
W20A14	-67,550	1.9	28	80	6	36	8	2,366
W20A10	-67,550	2.3	29	80	6	36	8	2,366
W20A7	-67,550	3.5	28	80	6	36	8	2,366
W20A5	-67,550	4.0	29	80	6	36	8	2,366
W21A7	-67,550	3.2	28	80	6	36	8	2,366
W24A10	-67,550	4.6	29	80	6	36	8	2,366
W30A24	-67,550	2.4	28	80	6	36	8	2,366
W30A21	-67,550	3.3	28	80	6	36	8	2,366
W30A20	-67,550	2.6	29	80	6	36	8	2,366
W21A14	-67,550	1.4	28	80	6	36	8	2,366
W20A15	-67,706	1.9	30	80	6	36	8	2,329
W10A5	-68,018	2.6	32	80	6	32	8	2,380
W30A28	-68,167	2.3	31	80	7	36	8	2,287
W14A12	-68,887	1.3	33	72	5	32	8	2,289
W15A12	-68,887	1.3	33	72	5	32	8	2,289
W24A24	-77,441	2.1	30	80	6	36	8	2,309
W14A5	-78,529	2.9	31	80	6	36	8	2,280
W15A7	-79,164	2.2	29	80	7	36	8	2,378
W21A10	-82,520	1.8	31	80	7	36	4	2,343
W10A10	-83,862	1.5	35	80	7	32	8	2,371
W20A12	-87,128	2.1	34	80	6	32	8	2,252
W15A10	-98,304	1.6	34	80	6	32	8	2,134
W15A15	-98,305	1.3	34	80	6	32	8	2,134
W15A14	-100,272	1.4	35	80	6	32	8	2,073

Table A.3. Scenario 2 results (1-30)

Config	Objective Value (\$K)	Time (m)	Violation Rate %	Below Threshold Max %	LOAD Events	RAS Events	INREP Events	Provider Plan Cost (\$K)
W30A7	-195	38.9	0	0	7	52	8	2,703
W30A20	-195	19.3	0	0	7	52	8	2,703
W30A5	-196	71.2	0	0	7	52	8	2,704
W30A12	-220	28.6	0	0	7	52	8	2,728
W24A12	-220	8.7	0	0	7	52	8	2,728
W30A14	-241	25.2	0	0	6	52	8	2,756
W24A10	-262	12.8	0	0	6	52	8	2,777
W20A7	-298	16.9	0	0	7	52	8	2,804
W28A20	-333	16.0	0	0	7	52	4	2,677
W28A10	-333	29.0	0	0	7	52	4	2,677
W28A7	-335	30.0	0	0	7	52	4	2,679
W28A21	-335	15.3	0	0	7	52	4	2,679
W28A28	-436	15.0	0	0	7	52	4	2,778
W28A14	-475	20.6	0	0	7	52	4	2,817
W21A7	-482	13.6	0	0	7	52	4	2,824
W21A10	-529	12.0	0	0	7	52	4	2,871
W30A15	-555	21.6	0	0	7	52	8	2,857
W21A14	-568	9.5	0	0	7	52	4	2,910
W30A21	-653	18.0	1	7	7	52	8	2,740
W24A14	-718	8.5	1	7	7	52	8	2,810
W24A7	-718	14.8	1	7	7	52	8	2,810
W30A10	-737	42.7	3	7	7	48	8	2,669
W20A5	-742	17.8	0	0	7	52	8	3,052
W28A5	-764	49.1	1	7	7	52	4	2,694
W30A28	-766	14.7	1	7	7	52	8	2,862
W30A30	-777	15.5	1	7	7	52	8	2,873
W24A24	-787	6.2	0	0	7	52	8	2,682
W14A7	-809	5.4	0	7	7	52	4	2,999
W30A24	-811	20.1	1	7	7	52	8	2,701
W20A14	-834	11.6	0	0	6	52	8	2,742

Table A.4. Scenario 2 results (31-60)

Config	Objective Value (\$K)	Time (m)	Violation Rate %	Below Threshold Max %	LOAD Events	RAS Events	INREP Events	Provider Plan Cost (\$K)
W28A24	-839	16.6	1	7	7	52	4	2,769
W20A15	-882	10.2	1	7	7	52	8	2,770
W20A10	-883	12.7	1	7	7	52	8	2,771
W24A5	-977	19.2	0	0	7	52	4	3,117
Monolith	-990	2,883.5	0	0	7	52	8	2,661
W21A12	-1,016	11.3	1	7	7	52	4	2,951
W14A10	-1,033	4.1	0	0	7	52	4	2,960
W21A5	-1,048	17.8	1	7	7	52	4	2,772
W21A20	-1,213	9.2	1	7	7	52	4	3,148
W24A21	-1,226	5.9	2	7	7	48	8	2,369
W21A21	-1,233	8.8	2	7	7	48	4	2,415
W20A20	-1,240	10.3	1	7	7	52	8	3,135
W21A15	-1,297	8.0	2	7	8	52	4	2,806
W28A15	-1,314	15.5	2	7	8	52	4	2,819
W15A7	-1,328	4.7	2	13	6	48	8	3,258
W28A12	-1,335	22.1	3	7	7	48	4	2,689
W24A15	-1,408	5.3	1	7	7	52	4	2,935
W24A20	-1,520	6.2	3	7	6	48	8	2,461
W15A5	-1,601	8.3	3	7	8	48	4	2,946
W14A14	-1,888	2.5	4	7	6	44	4	2,510
W15A10	-1,964	3.9	2	7	7	52	4	3,273
W20A12	-2,055	10.8	3	7	6	48	8	3,377
W10A5	-2,163	6.6	2	9	7	52	4	3,171
W14A5	-2,357	8.5	3	7	7	52	4	3,047
W10A10	-3,225	4.6	3	9	7	52	4	3,615
W15A12	-3,262	4.1	5	13	6	44	4	2,245
W10A7	-3,305	4.6	5	13	6	44	4	2,491
W14A12	-3,640	2.9	5	13	7	48	4	3,364
W15A14	-4,599	3.2	7	20	6	44	4	2,527
W15A15	-17,815	3.3	9	47	7	44	4	3,354

Table A.5. Scenario 3 results (1-30)

Config	Objective Value (\$K)	Time (h)	Violation Rate %	Below Threshold Max %	LOAD Events	RAS Events	INREP Events	Provider Plan Cost (\$K)
W21A12	-21,960	13.4	9	20	17	89	0	3,821
W30A15	-21,996	24.1	9	20	18	87	0	3,928
W28A24	-22,099	16.0	9	20	18	91	0	3,789
W28A15	-22,177	12.7	9	20	18	89	0	3,982
W20A12	-22,207	8.4	9	20	18	88	0	3,921
W28A21	-22,213	16.1	9	20	18	88	0	3,986
W24A12	-22,293	10.4	9	20	16	92	0	3,838
W15A10	-22,311	4.9	9	20	17	90	0	3,978
W28A14	-22,413	18.2	9	20	17	92	0	4,118
W20A14	-22,435	9.4	9	20	18	91	0	4,004
W28A20	-22,437	16.2	9	23	18	90	0	3,925
W24A15	-22,693	8.2	10	23	17	89	0	3,824
W30A28	-22,812	11.9	9	20	19	90	0	3,525
W30A30	-22,911	24.0	9	20	17	85	0	3,428
W21A14	-22,921	14.0	10	20	16	84	0	3,885
W30A21	-22,997	16.1	10	21	18	84	0	3,292
W20A10	-23,019	11.5	10	21	17	81	0	3,655
W24A20	-23,137	5.3	9	29	17	91	0	4,133
W30A20	-23,147	16.1	9	23	18	86	0	3,610
W28A28	-23,245	16.1	10	23	18	94	0	3,915
W14A10	-23,251	3.0	10	29	16	89	0	4,189
W14A7	-23,339	5.6	11	20	16	87	0	3,661
W14A14	-23,565	1.0	11	23	15	89	0	3,869
W24A14	-23,781	9.4	10	20	18	84	0	3,856
W30A24	-24,284	16.1	10	24	16	85	0	3,266
W15A15	-24,475	1.5	12	20	17	84	0	3,522
W15A14	-24,704	2.2	12	23	17	80	0	3,384
W24A21	-25,259	10.0	8	34	16	84	0	3,933
Monolith	-25,843	24.0	11	34	16	84	0	4,313
W24A24	-26,648	9.9	13	27	14	89	0	3,617

Table A.6. Scenario 3 results (31-39)

Config	Objective Value (\$K)	Time (h)	Violation Rate %	Below Threshold Max %	LOAD Events	RAS Events	INREP Events	Provider Plan Cost (\$K)
W20A15	-28,399	9.1	12	34	15	93	0	4,161
W20A20	-29,007	4.8	11	26	16	83	0	3,617
W15A12	-31,224	1.0	15	27	13	77	0	3,007
W21A15	-31,820	12.4	13	39	13	83	0	3,576
W10A7	-32,371	1.0	11	40	15	94	0	3,957
W10A10	-34,693	0.8	12	28	12	83	0	3,614
W21A20	-34,849	7.1	12	30	14	83	0	3,586
W14A12	-36,751	1.9	16	36	15	73	0	3,362
W21A21	-38,166	8.1	12	40	11	84	0	3,685

Table A.7. Scenario 4 warm start results (1-30)

Config	Objective Value (\$K)	Time (h)	Violation Rate %	Below Threshold Max %	LOAD Events	RAS Events	INREP Events	Provider Plan Cost (\$K)
W14A7	-10,133	6.0	3	20	87	103	0	5,952
W15A10	-10,172	6.0	6	13	88	96	0	5,671
W14A7	-10,289	3.0	4	20	88	101	0	5,932
W20A14	-10,379	6.0	6	13	95	98	0	5,737
W15A10	-10,482	1.0	7	13	89	95	0	5,674
W20A15	-10,592	6.0	8	13	90	96	0	5,695
W10A5	-10,618	3.0	6	13	90	95	0	5,694
W24A12	-10,687	6.0	7	13	86	94	0	5,582
W15A10	-10,689	3.0	6	13	88	95	0	5,687
W14A7	-10,728	1.0	7	13	89	98	0	5,849
W21A14	-10,936	6.0	9	20	91	98	0	5,705
W20A10	-10,961	6.0	10	13	92	96	0	5,707
W15A10	-11,019	2.0	8	13	87	92	0	5,697
W21A15	-11,079	3.0	9	20	88	97	0	5,859
W20A12	-11,111	6.0	7	13	88	93	0	5,660
W21A14	-11,170	3.0	9	20	87	95	0	5,743
W10A7	-11,218	6.0	6	13	89	96	0	5,736
W14A7	-11,220	2.0	7	20	89	99	0	5,944
W20A12	-11,224	3.0	9	13	88	94	0	5,651
W20A12	-11,300	1.0	9	20	88	96	0	5,667
W20A14	-11,309	3.0	9	13	90	93	0	5,699
W20A10	-11,379	1.0	7	13	87	94	0	5,766
W21A14	-11,424	2.0	11	20	89	94	0	5,613
W10A7	-11,447	2.0	6	13	88	95	0	5,743
W10A5	-11,482	2.0	6	13	88	90	0	5,637
W10A7	-11,509	3.0	6	13	89	94	0	5,695
W21A15	-11,533	6.0	8	20	88	91	0	5,480
W12A7	-11,534	3.0	9	20	87	93	0	5,584
W20A12	-11,556	2.0	9	20	89	95	0	5,697
W21A12	-11,679	6.0	8	33	89	100	0	5,841

Table A.8. Scenario 4 warm start results (31-60)

Config	Objective Value (\$K)	Time (h)	Violation Rate %	Below Threshold Max %	LOAD Events	RAS Events	INREP Events	Provider Plan Cost (\$K)
W20A15	-11,731	2.0	7	20	85	92	0	5,669
W10A7	-11,732	1.0	7	13	87	94	0	5,803
W28A14	-11,758	6.0	12	20	87	95	0	5,780
W20A15	-11,805	3.0	10	13	88	91	0	5,477
W14A10	-11,976	2.0	7	27	86	95	0	5,694
W21A15	-12,024	2.0	11	20	90	94	0	5,589
W24A20	-12,067	3.0	10	13	88	93	0	5,571
W20A14	-12,101	1.0	9	20	88	98	0	5,936
W24A15	-12,115	6.0	9	20	87	92	0	5,588
W10A5	-12,139	1.0	8	13	86	90	0	5,555
W21A21	-12,140	6.0	11	20	90	94	0	5,589
W15A12	-12,145	3.0	10	13	89	95	0	5,738
W14A12	-12,198	6.0	10	27	90	95	0	5,718
W24A15	-12,221	2.0	9	20	88	92	0	5,537
W12A7	-12,305	2.0	9	27	91	96	0	5,731
W12A10	-12,407	3.0	13	20	88	92	0	5,524
W12A7	-12,532	1.0	10	27	91	96	0	5,846
W20A14	-12,678	2.0	10	13	87	91	0	5,845
W15A14	-12,683	3.0	13	13	87	89	0	5,475
W20A10	-12,943	3.0	11	20	85	94	0	5,770
W14A12	-12,951	3.0	12	27	89	94	0	5,666
W24A12	-12,955	2.0	10	27	88	96	0	5,843
W21A12	-12,975	2.0	10	33	90	97	0	5,964
W20A10	-12,981	2.0	12	27	91	99	0	5,753
W15A12	-13,042	6.0	7	7	87	91	0	5,472
W24A20	-13,050	2.0	11	13	86	90	0	5,418
W15A12	-13,065	2.0	8	13	86	90	0	5,582
W12A10	-13,199	2.0	14	20	87	90	0	5,440
W24A21	-13,215	6.0	11	33	87	88	0	5,440
W14A10	-13,298	6.0	9	20	86	88	0	5,332

Table A.9. Scenario 4 warm start results (61-90)

Config	Objective Value (\$K)	Time (h)	Violation Rate %	Below Threshold Max %	LOAD Events	RAS Events	INREP Events	Provider Plan Cost (\$K)
W24A21	-13,321	1.0	11	20	88	91	0	5,554
W24A14	-13,338	3.0	11	27	90	93	0	5,705
W15A12	-13,467	1.0	10	13	88	92	0	5,486
W14A12	-13,519	2.0	12	40	88	91	0	5,638
W14A10	-13,630	3.0	10	20	85	87	0	5,364
W24A12	-13,638	3.0	11	27	87	94	0	5,615
W21A12	-13,690	3.0	14	33	87	91	0	5,579
W28A21	-13,767	6.0	12	20	88	92	0	5,566
W14A12	-13,858	1.0	13	40	88	93	0	5,644
W24A15	-13,885	1.0	14	27	87	92	0	5,683
W21A15	-13,982	1.0	14	27	88	90	0	5,642
W24A14	-14,067	2.0	13	20	90	92	0	5,666
W14A10	-14,103	1.0	12	20	88	90	0	5,564
W24A14	-14,150	1.0	11	27	91	96	0	5,864
W20A15	-14,172	1.0	13	27	88	93	0	5,641
W24A14	-14,389	6.0	10	40	87	91	0	5,458
W24A15	-14,419	3.0	14	27	89	92	0	5,594
W30A15	-14,665	3.0	16	27	89	91	0	5,538
W12A10	-14,767	1.0	16	27	89	93	0	5,617
W30A15	-15,166	6.0	15	27	86	87	0	5,461
W30A21	-15,415	2.0	16	33	91	94	0	5,671
W28A28	-15,587	6.0	16	20	85	91	0	5,725
W45A15	-15,801	3.0	16	27	89	91	0	5,545
W21A12	-15,817	1.0	14	33	87	94	0	5,646
W30A20	-15,895	6.0	16	27	87	91	0	5,449
W10A10	-15,966	6.0	17	27	88	92	0	5,508
W15A14	-16,001	2.0	18	33	87	90	0	5,904
W20A20	-16,066	6.0	18	20	89	91	0	5,538
W45A15	-16,090	6.0	14	27	87	88	0	5,397
W30A21	-16,125	3.0	12	27	85	92	0	5,627

Table A.10. Scenario 4 warm start results (91-120)

Config	Objective Value (\$K)	Time (h)	Violation Rate %	Below Threshold Max %	LOAD Events	RAS Events	INREP Events	Provider Plan Cost (\$K)
W30A30	-16,287	6.0	18	40	85	89	0	5,435
W28A20	-16,444	3.0	17	27	87	91	0	5,641
W24A21	-16,450	3.0	15	33	86	89	0	5,516
W14A14	-16,456	6.0	19	27	88	90	0	5,514
W15A14	-16,586	1.0	17	27	87	89	0	5,731
W15A15	-16,633	3.0	19	27	88	92	0	5,627
W30A20	-16,648	3.0	16	27	89	90	0	5,502
W24A24	-16,812	2.0	19	27	88	90	0	5,555
W28A15	-16,937	6.0	16	33	86	89	0	5,454
W15A15	-16,955	6.0	20	27	87	89	0	5,354
W28A24	-17,040	6.0	17	33	87	91	0	5,536
W15A15	-17,073	1.0	19	27	87	89	0	5,630
W30A24	-17,142	3.0	17	33	88	88	0	5,361
Monolith	-17,183	6.0	14	27	88	91	0	5,956
W24A24	-17,186	6.0	16	33	87	91	0	5,567
W12A12	-17,408	3.0	17	27	86	87	0	5,365
W45A30	-17,469	6.0	15	27	86	89	0	5,519
W24A20	-17,671	1.0	17	27	86	88	0	5,385
W28A24	-17,866	2.0	18	27	87	89	0	5,643
W24A21	-17,894	2.0	16	33	87	90	0	5,544
W21A14	-17,912	1.0	17	20	86	89	0	5,769
W15A15	-18,028	2.0	19	27	85	88	0	5,735
W30A20	-18,038	2.0	18	33	89	92	0	5,621
W30A15	-18,643	2.0	17	33	86	88	0	5,693
W28A20	-18,763	6.0	16	40	87	91	0	5,629
W20A20	-18,941	1.0	19	27	84	86	0	5,548
W45A30	-19,180	1.0	18	27	85	88	0	5,533
W28A14	-19,381	2.0	19	27	87	92	0	5,922
W14A14	-19,435	3.0	22	40	86	88	0	5,399
W28A14	-19,469	1.0	17	40	87	92	0	5,630

Table A.11. Scenario 4 warm start results (121-150)

Config	Objective Value (\$K)	Time (h)	Violation Rate %	Below Threshold Max %	LOAD Events	RAS Events	INREP Events	Provider Plan Cost (\$K)
W28A14	-19,783	3.0	20	33	88	90	0	5,561
W21A21	-19,999	3.0	21	33	85	87	0	5,498
W45A45	-20,423	6.0	23	33	89	90	0	5,663
W20A20	-20,880	3.0	22	33	85	87	0	5,457
W12A12	-20,909	2.0	22	40	84	85	0	5,420
W14A14	-21,318	1.0	21	40	86	86	0	5,513
W14A14	-21,581	2.0	23	40	85	86	0	5,659
W21A21	-21,786	2.0	23	33	85	87	0	5,696
W30A24	-22,001	2.0	21	27	88	90	0	5,721
W12A12	-22,784	1.0	26	40	83	85	0	5,399
W21A20	-23,147	3.0	26	40	85	87	0	5,541
W10A10	-23,388	3.0	27	27	86	89	0	5,565
W10A10	-23,515	2.0	27	27	85	88	0	5,548
W30A21	-24,116	1.0	25	40	87	91	0	5,787
W28A21	-24,647	2.0	23	33	87	90	0	5,647
W24A12	-24,700	1.0	23	40	86	90	0	5,933
W21A20	-24,756	2.0	26	40	87	89	0	5,682
W20A20	-24,953	2.0	26	40	87	89	0	5,690
W24A24	-25,186	3.0	24	33	86	87	0	5,441
W10A10	-25,773	1.0	29	33	85	89	0	5,699
W21A21	-27,228	1.0	27	47	84	87	0	5,732
W30A20	-28,342	1.0	28	47	87	87	0	5,711
W28A21	-30,564	3.0	30	47	85	88	0	5,481
W30A28	-30,997	3.0	30	40	84	86	0	5,665
W28A15	-32,133	3.0	31	40	84	85	0	5,602
W30A28	-33,121	2.0	29	47	85	87	0	5,984
W45A45	-36,869	3.0	31	47	85	88	0	5,794
W24A24	-37,031	1.0	36	40	86	88	0	5,837
W28A24	-37,849	3.0	36	53	87	87	0	5,490
W21A20	-41,759	1.0	40	47	86	87	0	5,321

Table A.12. Scenario 4 warm start results (151-174)

Config	Objective Value (\$K)	Time (h)	Violation Rate %	Below Threshold Max %	LOAD Events	RAS Events	INREP Events	Provider Plan Cost (\$K)
W30A24	-42,170	1.0	39	47	86	87	0	5,770
Monolith	-42,592	3.0	38	47	85	86	0	5,630
W45A30	-42,827	3.0	31	60	90	92	0	5,793
W30A15	-42,904	1.0	41	40	84	85	0	5,363
W30A30	-43,943	3.0	40	53	85	85	0	5,468
W30A30	-46,988	2.0	44	47	85	88	0	5,647
W30A30	-49,730	1.0	45	53	85	86	0	5,487
W45A45	-50,235	2.0	43	53	86	88	0	5,547
W45A15	-50,909	2.0	42	60	84	85	0	5,593
W28A15	-52,268	2.0	39	60	86	86	0	5,613
W28A20	-53,683	2.0	46	53	84	84	0	5,417
W28A24	-57,289	1.0	52	53	83	86	0	5,817
W28A28	-57,348	2.0	53	47	83	84	0	5,539
W45A15	-67,475	1.0	50	53	80	81	0	6,071
W30A28	-68,158	1.0	53	67	81	81	0	5,506
W28A28	-75,980	1.0	53	67	79	80	0	5,442
W28A28	-79,774	3.0	61	60	82	83	0	5,575
Monolith	-125,024	1.0	62	73	79	79	0	5,425
Monolith	-125,024	2.0	62	73	79	79	0	5,425
W28A21	-125,306	1.0	71	80	82	83	0	6,072
W45A30	-150,066	2.0	70	80	80	80	0	5,556
W28A20	-168,840	1.0	67	80	80	80	0	5,663
W28A15	-189,506	1.0	75	80	84	84	0	5,505
W45A45	-208,644	1.0	72	87	80	81	0	5,713

Table A.13. Scenario 4 rolling horizon results (1-30)

Config	Objective Value (\$K)	Time Limit (h)	Time Used (h)	Violation Rate %	Below Threshold Max %	Provider Plan Cost (\$K)
W15A10	-10,470	3	1.3	6	13	5,824
W14A7	-10,703	2	1.0	4	20	6,004
W14A7	-10,703	3	1.1	4	20	6,004
W15A10	-10,828	1	0.7	8	13	5,831
W20A12	-10,902	3	2.6	9	13	5,715
W15A10	-10,916	2	1.1	7	13	5,793
W21A14	-10,970	3	2.6	9	20	5,833
W14A7	-11,208	1	0.7	8	13	5,927
W20A12	-11,272	2	1.6	9	13	5,701
W21A15	-11,461	3	2.5	9	20	5,893
W20A14	-11,682	1	0.9	7	13	5,710
W24A14	-11,699	1	0.9	10	13	5,619
W20A10	-11,713	2	1.9	9	13	5,865
W20A15	-11,763	2	1.5	10	20	5,871
W21A14	-11,803	2	1.8	10	20	5,944
W24A14	-11,904	3	2.6	9	13	5,800
W15A12	-11,920	1	0.6	9	13	5,726
W10A7	-12,010	1	0.2	7	13	5,835
W10A7	-12,010	2	0.2	7	13	5,835
W10A7	-12,010	3	0.2	7	13	5,835
W20A14	-12,071	3	2.6	13	20	5,675
W20A14	-12,082	2	1.8	10	13	5,881
W12A7	-12,185	3	0.5	9	27	5,771
W24A20	-12,262	3	2.4	9	20	5,543
W24A12	-12,299	3	2.6	10	20	5,887
W20A15	-12,310	3	2.2	10	13	5,684
W20A10	-12,316	3	2.7	11	13	5,795
W15A12	-12,367	2	1.0	9	20	5,913
W12A7	-12,672	1	0.4	10	27	5,908
W24A21	-12,748	3	2.4	13	20	5,397

Table A.14. Scenario 4 rolling horizons results (31-60)

Config	Objective Value (\$K)	Time Limit (h)	Time Used (h)	Violation Rate %	Below Threshold Max %	Provider Plan Cost (\$K)
W21A12	-12,756	3	2.5	12	27	6,089
W24A12	-12,779	2	1.8	11	20	5,870
W10A5	-12,859	1	0.3	8	13	6,022
W10A5	-12,859	2	0.3	8	13	6,022
W10A5	-12,859	3	0.4	8	13	6,022
W15A12	-13,100	3	1.4	8	13	5,602
W12A7	-13,128	2	0.7	10	27	5,704
W21A12	-13,189	2	1.7	11	33	5,928
W20A15	-13,526	1	0.9	11	27	5,824
W12A10	-13,751	2	0.6	16	20	5,433
W12A10	-13,751	3	0.7	16	20	5,433
W14A12	-13,764	1	0.5	13	40	5,759
W14A10	-13,782	1	0.6	11	27	5,584
W12A10	-13,849	1	0.5	14	27	5,798
W24A20	-13,954	2	1.6	11	20	5,647
W14A12	-13,999	3	1.2	11	40	5,752
W21A15	-14,073	2	1.7	13	27	5,990
W24A21	-14,135	2	1.6	15	20	5,565
W14A12	-14,138	2	1.0	12	40	5,768
W14A10	-14,240	2	1.1	10	27	5,454
W14A10	-14,240	3	1.4	10	27	5,454
W30A15	-14,340	2	2.0	13	27	5,814
W21A14	-14,544	1	0.9	13	20	5,951
W15A14	-14,815	2	1.0	16	27	5,658
W24A20	-14,896	1	0.8	14	20	5,711
W24A15	-15,270	2	1.7	13	33	5,906
W24A15	-15,299	3	2.5	13	27	5,578
W20A20	-15,604	3	2.4	17	27	5,753
W28A24	-15,820	3	2.3	16	27	5,737
W15A14	-15,921	3	1.4	17	27	5,754

Table A.15. Scenario 4 rolling horizon results (61-90)

Config	Objective Value (\$K)	Time Limit (h)	Time Used (h)	Violation Rate %	Below Threshold Max %	Provider Plan Cost (\$K)
W20A12	-16,037	1	0.9	14	40	5,691
W30A15	-16,723	3	3.1	16	27	5,598
W24A14	-16,726	2	1.7	16	33	5,796
W30A20	-16,818	3	3.0	19	27	5,503
W20A10	-17,053	1	1.0	18	33	5,803
W30A20	-17,225	2	2.0	18	27	5,474
W30A24	-17,829	3	2.3	19	27	5,583
W30A21	-18,066	3	3.1	19	27	5,608
W21A15	-18,273	1	0.9	19	33	5,947
W24A24	-18,890	3	2.3	16	40	5,603
W24A15	-18,899	1	0.9	20	27	5,734
W30A24	-19,129	2	1.5	18	27	5,735
W15A15	-19,197	2	0.9	23	20	5,767
W20A20	-19,201	2	1.4	21	27	5,607
W15A14	-19,304	1	0.7	22	33	5,697
W21A12	-20,115	1	0.8	20	40	5,633
W24A21	-20,151	1	0.8	21	27	5,644
W45A15	-20,223	3	3.0	19	27	5,658
W12A12	-20,643	3	0.9	20	40	5,747
W21A20	-20,744	3	2.4	24	27	5,782
W21A20	-20,781	2	1.4	21	40	6,196
W15A15	-20,796	1	0.6	26	27	5,716
W28A21	-20,926	3	2.5	21	33	5,712
W28A14	-22,118	2	1.7	22	40	5,854
W15A15	-22,788	3	1.5	26	33	6,033
W14A14	-22,825	2	1.1	26	40	5,466
W14A14	-22,825	3	1.4	26	40	5,466
W14A14	-22,915	1	0.6	25	40	5,609
W24A24	-23,023	1	0.8	26	33	5,475
W28A14	-23,078	3	2.6	23	40	5,769

Table A.16. Scenario 4 rolling horizon results (91-120)

Config	Objective Value (\$K)	Time Limit (h)	Time Used (h)	Violation Rate %	Below Threshold Max %	Provider Plan Cost (\$K)
W21A21	-23,167	2	1.5	24	27	5,580
W12A12	-23,323	2	0.9	25	40	5,504
W10A10	-23,673	1	0.3	27	27	5,706
W10A10	-23,673	2	0.2	27	27	5,706
W10A10	-23,673	3	0.2	27	27	5,706
W21A21	-24,432	3	2.4	25	40	5,883
W12A12	-24,488	1	0.5	27	40	5,481
W24A12	-24,945	1	0.9	25	40	5,935
W24A24	-25,120	2	1.6	24	33	5,671
W30A28	-26,497	3	2.3	26	33	5,555
W21A21	-26,555	1	0.8	28	33	5,861
W28A21	-26,642	2	2.0	27	40	5,741
W30A20	-30,100	1	1.0	31	40	5,597
W28A28	-32,591	3	2.3	32	40	5,532
W30A21	-32,715	2	2.0	33	40	6,020
W30A30	-32,939	3	3.0	34	40	5,597
W45A15	-34,118	2	2.0	35	40	5,718
W45A45	-37,822	3	3.0	37	40	5,930
W45A30	-37,870	2	2.0	35	47	5,742
W28A20	-39,060	2	1.6	40	40	5,648
W28A15	-40,453	3	2.2	37	47	5,493
W28A20	-40,957	3	2.4	41	47	5,629
W28A28	-41,050	2	1.5	41	47	5,501
W21A20	-41,444	1	0.8	40	40	5,932
W30A28	-42,776	2	1.5	42	40	5,678
W45A30	-44,093	3	3.0	37	53	5,884
W28A14	-45,027	1	0.9	44	40	5,596
W20A20	-49,828	1	0.8	46	53	5,851
W45A45	-53,818	2	2.0	37	60	5,940
W28A15	-65,515	2	1.7	50	67	5,872

Table A.17. Scenario 4 rolling horizon results (120-135)

Config	Objective Value (\$K)	Time Limit (h)	Time Used (h)	Violation Rate %	Below Threshold Max %	Provider Plan Cost (\$K)
W28A21	-65,865	1	1.0	51	53	5,998
W30A30	-66,221	2	2.0	51	53	6,100
W30A28	-67,569	1	0.8	54	53	5,417
W30A21	-67,705	1	1.0	53	67	5,633
W28A20	-72,968	1	0.8	59	67	6,125
W28A24	-74,370	2	1.5	58	53	5,733
W45A15	-77,276	1	1.0	57	60	6,073
W45A30	-78,403	1	1.0	58	53	6,217
W30A30	-78,518	1	1.0	58	60	5,863
W28A15	-85,527	1	0.8	56	60	6,175
W30A24	-90,668	1	0.8	52	73	5,825
W30A15	-96,725	1	1.0	62	67	5,708
W45A45	-115,173	1	1.0	67	67	5,846
W28A28	-203,306	1	0.8	71	93	5,314
W28A24	-245,029	1	0.8	74	107	5,673

List of References

- Absi N, van den Heuvel W (2019) Worst case analysis of relax and fix heuristics for lot-sizing problems. *European Journal of Operational Research* 279:449–458.
- Baker J (2019) Optimizing the Navy Mission Planner, https://calhoun.nps.edu/bitstream/handle/10945/62282/19Mar_Baker_Joshua.pdf?sequence=1&isAllowed=y.
- Baker K (1977) An experimental study of the effectiveness of rolling schedules in production planning. *Decision sciences* 8:19–27.
- Blackburn W (2016) Otter: an optimized transit tool and easy reference, https://calhoun.nps.edu/bitstream/handle/10945/48585/16Mar_Blackburn_Warren.pdf?sequence=1&isAllowed=y.
- Brown G (2004) Top ten secrets to success with optimization. *PHALANX* 37(4):12.
- Brown G (2021) Studies and publications using the Replenishment at Sea Planner (RASP), [http://faculty.nps.edu/gbrown/docs/2021.06\[CA\]\(RASP-Studies.Theses.Papers.URLs\).pdf](http://faculty.nps.edu/gbrown/docs/2021.06[CA](RASP-Studies.Theses.Papers.URLs).pdf).
- Brown G, Carlyle W (2008) Optimizing the US Navy’s Combat Logistics Force. *Naval Research Logistics* 55:800–810, <https://doi.org/10.1002/nav.20318>.
- Brown G, DeGrange W, Price W, Rowe A (2017) Scheduling Combat Logistics Force replenishments at sea for the US Navy. *Naval Research Logistics* 64:677–693, <https://doi.org/10.1002/nav.21780>.
- Brown G, Dell R, Wood R (1997) Optimization and persistence. *Interfaces* 27(5):15–37.
- Brown G, Graves G, Ronen D (1987) Scheduling ocean transportation of crude oil. *Management Science* 33:335–346.
- Bynum M, Hackebeil G, Hart W, Laird C, Nicholson B, Siirola J, Watson JP, Woodruff D (2021) *Pyomo – Optimization Modeling in Python*, third ed., volume 67 (Springer Science & Business Media).
- Chand S, Hsu V, Sethi S (2002) Forecast, solution, and rolling horizons in operations management problems: A classified bibliography. *Manufacturing & Service Operations Management* 4:25–43.
- Colburn B (2015) Preserving logistical support for deployed battle groups in an anti-access, area denial (A2AD) environment, https://calhoun.nps.edu/bitstream/handle/10945/47240/15Sep_Colburn_Brian.pdf?sequence=1&isAllowed=y.

- Crawford D (2014) Alternative practices to improve surface fleet fuel efficiency, https://calhoun.nps.edu/bitstream/handle/10945/43897/14Sep_Crawford_Dustin.pdf?sequence=1&isAllowed=y.
- Cribbs M (2016) A baseline analysis of combat logistics force scheduling efficiency, https://calhoun.nps.edu/bitstream/handle/10945/49444/16Jun_Cribbs_Michael.pdf?sequence=1&isAllowed=y.
- Deiter J (2022) Statistical sensitivity analysis of the Replenishment at Sea Planner, https://calhoun.nps.edu/bitstream/handle/10945/71054/22Sep_Deiter_Jared.pdf?sequence=1&isAllowed=y.
- Deleon J (2015) Navy operational planner, <https://apps.dtic.mil/sti/pdfs/ADA620805.pdf>.
- Department of the Navy (2001) *Underway Replenishment*. NWP 4-01.4. Washington, DC, USA, <https://www.maritime.dot.gov/sites/marad.dot.gov/files/2023-01/unrep-nwp04-01.pdf>.
- Diaz A (2010) Heuristics in global combat logistics force operational planning, https://calhoun.nps.edu/bitstream/handle/10945/5456/10Mar_Diaz_Andres.pdf?sequence=1&isAllowed=y.
- Dillenberger C, Escudero LF, Wollensak A, Zhang W (1994) On practical resource allocation for production planning and scheduling with period overlapping setups. *European Journal of Operational Research* 75(2):275–286, [https://doi.org/https://doi.org/10.1016/0377-2217\(94\)90074-4](https://doi.org/https://doi.org/10.1016/0377-2217(94)90074-4).
- Doyle D (2006) Evaluation of fleet ownership versus global allocation of ships in the Combat Logistics Force, <https://apps.dtic.mil/sti/tr/pdf/ADA456929.pdf>.
- Dönmez K (2023) Evaluation of the performance of the multi-objective scalarization methods for the aircraft sequencing and scheduling problem using multi-criteria decision-making. *Aircraft Engineering and Aerospace Technology* 95(4):501–511.
- Friske M, Buriol L, Camponogara E (2022) A relax-and-fix and fix-and-optimize algorithm for a maritime inventory routing problem. *Computers & Operations Research* 137.
- Gilday M (2023) Chief of Naval Operations on the posture of the United States Navy. Senate Armed Services Committee, https://www.armed-services.senate.gov/imo/media/doc/18APR23_CNO_SASC_USN_Posture-Hearing_FINAL%20for%20Hill.pdf.
- Glomb L, Liers F, Rösel F (2022) A rolling-horizon approach for multi-period optimization. *European Journal of Operational Research* 300:189–206.

- Guo L, Ge P, Sun D, Qiao Y (2020) Adaptive cruise control based on model predictive control with constraints softening. *Applied Sciences* 10(5).
- Hathorn B (2013) Resource burden of logistics to navy ships under threat scenarios, https://calhoun.nps.edu/bitstream/handle/10945/32833/13Mar_Hathorn_Brendon.pdf?sequence=1&isAllowed=y.
- Johnson S (1957) Sequential production planning over time at minimum cost. *Management Science* 3(4):435–437.
- Joncour C, Kritter J, Michel S, Schepler X (2023) Generalized relax-and-fix heuristic. *Computers & Operations Research* 149.
- Krenz J (2018) Multicommodity logistical support in an anti-access, area denial environment, https://calhoun.nps.edu/bitstream/handle/10945/58323/18Mar_Krenz_Jonathan.pdf?sequence=1&isAllowed=y.
- Kress M (2002) *Operational Logistics: the Art and Science of Sustaining Military Operations* (Kluwer Academic Publishers, Boston).
- Lantz S (2014) Incorporation of tropical cyclone avoidance into automated ship scheduling, https://calhoun.nps.edu/bitstream/handle/10945/42670/14Jun_Lantz_Stephen.pdf?sequence=1&isAllowed=y.
- Long J (2011) Optimizing logistical planning factors for fuel rate consumption within the FIFTH FLEET area of responsibility, https://calhoun.nps.edu/bitstream/handle/10945/5713/11Jun_Long.pdf?sequence=1&isAllowed=y.
- Marquantand J, Evins R, Carmeliet J (2015) Reducing computation time with a rolling horizon approach applied to a MILP formulation of multiple urban energy hub system. *Procedia Computer Science* 51:2137–2146.
- Miettinen K (1998) *Nonlinear Multiobjective Optimization*, 1st ed. 1998. ed. (Springer, New York, NY).
- Military Sealift Command (2020) Photo gallery. Nov. 2, <https://www.msc.usff.navy.mil/Press-Room/Photo-Gallery/igphoto/2002528306/>.
- Miller S (2012) Smart voyage planning model sensitivity analysis using ocean and atmospheric models including ensemble methods, https://calhoun.nps.edu/bitstream/handle/10945/17422/12Sep_Miller_Scott.pdf?sequence=1&isAllowed=y.
- Modigliani F, Hohn F (1955) Production planning over time and the nature of the expectation and planning horizon. *Econometrica* 23:46–66.

- Plotly Technologies Inc (2015) Collaborative data science. Montreal, QC, <https://plot.ly>.
- Ralphs T, Güzelsoy M (2006) Duality and warm starting in integer programming. 2006 *National Science Foundation Design, Service, and Manufacturing Grantees and Research Conference*, <https://coral.ise.lehigh.edu/~ted/files/papers/DMII06.pdf2>.
- Royset J, Wets R (2021) *An Optimization Primer* (Springer, Cham, Switzerland).
- Sentinella F (2021) Using data farming and optimization to enable analysis of concepts of employment for surface connectors, https://calhoun.nps.edu/bitstream/handle/10945/67811/21Jun_Sentinella_Forest.pdf?sequence=1&isAllowed=y.
- Sethi S, Sorger G (1991) A theory of rolling horizon decision making. *Annals of Operations Research* 29:387–415.
- Sierksma G, Zwols Y (2015) *Linear and Integer Optimization: Theory and Practice*, 3rd ed. (CRC Press, Boca Raton, FL).
- Stadtler H (2003) Multilevel lot sizing with setup times and multiple constrained resources: Internally rolling schedules with lot-sizing windows. *Operations Research* 51(3):487–502.
- United States 7th Fleet (2023) Facts sheet. Jul. 9, <https://www.c7f.navy.mil/About-Us/Facts-Sheet/>.
- Wagner H, Whitin T (1958) Dynamic version of the economic lot size model. *Management Science* 5:89–96.
- Walton T, Schramm H, Boone R (2019) *Sustaining the Fight: Resilient Maritime Logistics for a New Era* (Center for Strategic and Budgetary Assessments).
- Washburn A, Brown G (2016) An exact method for finding shortest routes on a sphere, avoiding obstacles. *Naval Research Logistics* 63(5):374–385.

Initial Distribution List

1. Defense Technical Information Center
Ft. Belvoir, Virginia
2. Dudley Knox Library
Naval Postgraduate School
Monterey, California



DUDLEY KNOX LIBRARY

NAVAL POSTGRADUATE SCHOOL

WWW.NPS.EDU

WHERE SCIENCE MEETS THE ART OF WARFARE