



NAVAL POSTGRADUATE SCHOOL

MONTEREY, CALIFORNIA

THESIS

OPTIMIZING THE NAVY MISSION PLANNER

by

Joshua D. Baker

March 2019

Thesis Advisor:
Second Reader:

Gerald G. Brown
W. Matthew Carlyle

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 March 2019	3. REPORT TYPE AND DATES COVERED Master's thesis		
4. TITLE AND SUBTITLE OPTIMIZING THE NAVY MISSION PLANNER			5. FUNDING NUMBERS	
6. AUTHOR(S) Joshua D. Baker				
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) The Navy Mission Planner (NMP) is an optimization-based operational planning tool for decision makers at all levels of designing naval deployments, from theater-level planning to individual Carrier Strike Groups (CSGs) or Destroyer Squadrons (DESRONs). Schedulers are tasked with too many missions and not enough ships to fulfill those missions. This decision aid takes multimission-capable ships and assigns them to missions across a given planning horizon with the goal of maximizing the total value of mission accomplishment, taking into consideration the geographical locations of mission sets and ship capabilities and limitations. Previous versions used licensed commercial software and solvers for optimization, as well as a limited enumeration of alternative ship deployments. This thesis focuses on making the Navy Mission Planner available to all naval personnel by using open-source software and solvers. In addition, it offers persistence within the optimization, allowing schedulers to reconfigure schedules in the middle of the planning horizon with minimum changes to previously promulgated schedules. We also develop two approaches to deployment planning, a random path enumeration and a network flow formulation, both of which increase the mission accomplishment levels in the Navy Mission Planner. Additionally, we create a "force ratio escort" parameter that allows for non-combatant ships to be escorted by multiple defense-capable ships through hazardous regions.				
14. SUBJECT TERMS optimization, Navy Mission Planner, NMP, scheduling			15. NUMBER OF PAGES 89	
			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	

THIS PAGE INTENTIONALLY LEFT BLANK

Approved for public release. Distribution is unlimited.

OPTIMIZING THE NAVY MISSION PLANNER

Joshua D. Baker
Lieutenant, United States Navy
BS, Southern Illinois University at Carbondale, 2012

Submitted in partial fulfillment of the
requirements for the degree of

MASTER OF SCIENCE IN OPERATIONS RESEARCH

from the

**NAVAL POSTGRADUATE SCHOOL
March 2019**

Approved by: Gerald G. Brown
Advisor

W. Matthew Carlyle
Second Reader

W. Matthew Carlyle
Chair, Department of Operations Research

THIS PAGE INTENTIONALLY LEFT BLANK

ABSTRACT

The Navy Mission Planner (NMP) is an optimization-based operational planning tool for decision makers at all levels of designing naval deployments, from theater-level planning to individual Carrier Strike Groups (CSGs) or Destroyer Squadrons (DESRONs). Schedulers are tasked with too many missions and not enough ships to fulfill those missions. This decision aid takes multimission-capable ships and assigns them to missions across a given planning horizon with the goal of maximizing the total value of mission accomplishment, taking into consideration the geographical locations of mission sets and ship capabilities and limitations. Previous versions used licensed commercial software and solvers for optimization, as well as a limited enumeration of alternative ship deployments. This thesis focuses on making the Navy Mission Planner available to all naval personnel by using open-source software and solvers. In addition, it offers persistence within the optimization, allowing schedulers to reconfigure schedules in the middle of the planning horizon with minimum changes to previously promulgated schedules. We also develop two approaches to deployment planning, a random path enumeration and a network flow formulation, both of which increase the mission accomplishment levels in the Navy Mission Planner. Additionally, we create a “force ratio escort” parameter that allows for non-combatant ships to be escorted by multiple defense-capable ships through hazardous regions.

THIS PAGE INTENTIONALLY LEFT BLANK

TABLE OF CONTENTS

I.	INTRODUCTION.....	1
A.	NAVAL TACTICAL PLANNING.....	1
B.	MARITIME PLANNING	2
C.	NAVY PLANNING PROCESS	3
D.	BACKGROUND AND PRIOR WORK REVIEW	5
1.	Naval Logistics	5
2.	Navy Mission Planner	6
II.	NAVY MISSION PLANNER	9
A.	AN INTEGER LINEAR PROGRAM TO OPTIMIZE NAVY MISSION PLANNING: NMP AND LOGISTIC SUPPORT	9
B.	NETWORK FLOW FORMULATION	20
III.	SOLUTION GENERATION FOR NAVY MISSION PLANNER	23
A.	PYTHON AND PYOMO WITH CBC (COIN-OR BRANCH AND CUT) SOLVER.....	23
B.	RANDOM EMPLOYMENT PATH GENERATOR.....	24
1.	Modifying NMP Path Generation	25
2.	Modification of Backtracking Algorithm	25
3.	Minimizing Time within a Region	26
4.	Preventing Restricted Regions.....	27
C.	PERSISTENCE IN OPTIMIZATION	27
IV.	SCENARIOS, ANALYSIS, AND CONCLUSION	31
A.	MATCHING SCENARIOS	31
1.	Missions.....	31
2.	Area of Operation	32
3.	Ships	34
4.	Logistics	34
B.	REPEATING AND REPRODUCING HALLMANN’S RESULTS	36
1.	Without Supply Ships	37
2.	With Supply Ships	37
3.	With Escorts	38
4.	With Close Escorts.....	38
C.	RANDOM PATH GENERATOR RESULTS	39
1.	Without Supply Ships	39

2.	With Supply Ships	39
3.	Escorts	40
4.	Close Escorts.....	40
D.	DEPLOYMENT NETWORK FLOW RESULTS	41
1.	Without Supply Ships	41
2.	With Supply Ships	42
3.	Escorts and Close Escorts	43
E.	PERSISTENCE SCENARIO.....	44
F.	PHILIPPINE SEA SCENARIO	47
1.	Ships	47
2.	Regions	48
3.	Commodities	49
4.	Results	50
G.	CONCLUSION AND FUTURE RESEARCH	51
APPENDIX A: MISSION INPUTS.....		55
APPENDIX B: SHIP CATALOG FOR NMP		59
APPENDIX C: COMMODITY USAGE BASED ON CMC		61
APPENDIX D: PHILIPPINE SEA MISSIONS		63
LIST OF REFERENCES		65
INITIAL DISTRIBUTION LIST		67

LIST OF FIGURES

Figure 1.	Plan-Brief-Execute-Debrief Cycle. Source: Department of the Navy, 2013a.....	4
Figure 2.	Algorithm Enumerating all the (finite number of) s-t Paths of Length Bounded by T. Source: Brown et al., 2013, p. 46.....	16
Figure 3.	Region Map of AOO in Korean Peninsula	33
Figure 4.	Objective Value Output of Navy Mission Planner	37
Figure 5.	Sawtooth Chart for Network Flow with Supply Ships	43
Figure 6.	Region Map of the AOO in Philippine Sea	47

THIS PAGE INTENTIONALLY LEFT BLANK

LIST OF TABLES

Table 1.	Mission Inputs.....	32
Table 2.	Region User Inputs	33
Table 3.	Sample Ship Inputs	34
Table 4.	Sample Consumption Rates based on Combined Mission Capabilities	35
Table 5.	Capacities and Load for Each Class of Ship.....	35
Table 6.	Inventory Safety Levels and Penalties.....	36
Table 7.	Persistence with No Penalties	45
Table 8.	Persistence with Penalties	46
Table 9.	Ships for Philippine Sea Scenario.....	48
Table 10.	Regions for Philippine Sea Scenario.....	49
Table 11.	Commodities Load for Philippine Sea Scenario.....	49
Table 12.	Philippine Sea Transit.....	50
Table 13.	Persistence in Philippine Sea Scenario	51
Table 14.	NMP Mission Inputs (1 of 3).....	55
Table 15.	NMP Mission Inputs (2 of 3).....	56
Table 16.	NMP Mission Inputs (3 of 3).....	57
Table 17.	NMP Ship Catalog	59
Table 18.	NMP Commodity Usage Catalog Based on Combined Mission Capabilities	61
Table 19.	Philippine Sea Missions.....	63

THIS PAGE INTENTIONALLY LEFT BLANK

LIST OF ACRONYMS AND ABBREVIATIONS

AD	Air Defense
AOO	Area of Operation
ASW	Anti-Submarine Warfare
CG	Cruiser, guided missile
CLF	Combat Logistics Force
CMC	Combined mission capability
COA	Course of Action
CONOPS	Concepts of Operations
CSG	Carrier Strike Group
CSV	Comma Separated Value
CVN	Aircraft Carrier, nuclear
DDG	Destroyer, guided missile
DESRON	Destroyer Squadron
DFM	Distilled Fuel Marine (NATO F75/76)
DoD	Department of Defense
DoN	Department of the Navy
ESG	Expeditionary Strike Group
FFG	Frigate, guided missile
GAMS	General Algebraic Modeling System
INTEL	Intelligence
JFMCC	Joint Forces Maritime Combatant Commander
JP5	Aviation Fuel (NATO F44)
JTF	Joint Task Force
LHA	Landing Helicopter Assault
LHD	Landing Helicopter Dock
LPD	Landing Platform/Dock
MCM	Mine Countermeasures
MIO	Maritime Interdiction Operation

MOC	Maritime Operation Center
NMCI	Navy Marine Corps Internet
NMP	Navy Mission Planner
NOP	Naval Operational Planner
NPP	Navy Planning Process
NSFS	Naval Surface Fire Support
NWP	Naval Warfare Publication
PBED	Plan, Brief, Execute, and Debrief
RAS	Replenishment at Sea
RASP	Replenishment at Sea Planner
STRIKE	Strike (Attack) Mission
SUBINTEL	Submarine Intelligence
SUW	Anti Surface Warfare
T-AE	Ammunition ship
T-AFS	Combat stores ship
T-AKE	Modular dry cargo and ammunition ship
T-AO	Fleet replenishment Oiler
T-AOE	Fast Combat Support Ship
TBMD	Theater Ballistic Missile Defense
USN	United States Navy
VBA	Visual Basic for Applications

EXECUTIVE SUMMARY

The Navy Mission Planner (NMP) is an optimization-based decision support system for decision makers at all levels of naval planning from the Joint Forces Maritime Combatant Commander (JFMCC) operational perspective down to the Destroyer Squadron (DESRON) and Carrier Strike Group (CSG) tactical views of planning. The task of planning missions across a planning horizon is already challenging, before taking into consideration the multiple ships, their multiple mission capabilities, and their time required to transit from one geographical region to another. We must also take into consideration any prerequisite mission(s) required to be completed simultaneously with a supported mission. Determining logistical support predicting when and where ships will require more fuel and other supplies across the same planning horizon can be challenging as well. This is still a manual and time-consuming process for the JFMCC, or a Maritime Operation Center (MOC), or DESRON, or CSG.

NMP takes user inputs describing missions, geographical regions, available ships, their combined mission capabilities, commodity consumption rates by mission and ship capacities. NMP maximizes the total mission accomplishment value by assigning capable ships to regions for mission execution. NMP incorporates a Combat Logistics Force (CLF) planning element capable of tracking ship commodity levels and advising when and where to conduct Replenishment at Sea (RAS) events with customer ships. NMP also features *escort* and *close escort* options that allow for non-combatant ships to be escorted by defensive combatant ships, either one defense ship in the same region for all non-combatants therein (escort) or a one-to-one assignment of defender and non-combatant ships in the same region (close escort).

This research explores ways to improve upon NMP by switching from expensive, licensed proprietary software not approved for use on Navy Marine Corps Internet (NMCI) computers, Secret Internet Protocol (SIPR) computers, or for other classified networks, to open-source software. By transforming NMP to an open-source algebraic modeling language we eliminate all cost and this software is approved or can be approved

for use on secure computers. The solvers were also switched over from a licensed optimization software to an open-source mixed integer programming solver.

We also look into improving the way in which ships can transit from region to region, and add a way to make schedule changes with minimal impact to previously promulgated schedules.

NMP's previous method to plan ship deployment from region-to-region involved a partial stack-based enumeration that limits the diversity of regions explored by ships. In this research we focus on changing the diversity of this region-to-region exploration with two additional methods. The first is a random path generation; much like the partial stacked-based enumeration, this generates a set number of deployment paths for each ship. However, the deployment route in which the ship takes is randomized giving added diversity to the alternate routing of ships. Random path generation improves NMP by increasing the number of missions accomplished and greatly reduces the penalties for commodity consumption by getting ships where they need to be for RAS events.

Our second method adds a network flow deployment model to the NMP allowing ships to explore all possible routes from region-to-region. Network flow greatly reduces the runtime of the NMP from any other routing models and gives us a near-optimal, if not optimal solution. Both the random path generation and network flow deployment model are compared against the deployment stack-based enumeration within our Korean area of operation scenario, where we have 695 missions to be completed across a 15-day planning horizon.

This research adds an additional escorting feature specifically for aircraft carriers or amphibious assault ships that require more than one escort within a region. This force ratio requires every individual non-combatant vessel to be escorted through high hazard regions by multiple defense-capable ships. We test this in a scenario new to NMP with a defended CLF convoy transiting from Guam to Cebu, Philippines, through the Philippine Sea.

Finally, this research adds persistence in optimization to NMP, a scheduling feature that allows a user to add or subtract missions and/or ships anytime or anywhere

over the planning horizon and still achieve mission accomplishment with few changes to the legacy schedules. This greatly reduces turbulence, messaging, and confusion from too many ships ordered to make too many plan adjustments.

The addition of our random deployment path generation, network flow deployment model, and persistence make significant improvements to NMP. The random deployment path generation shows an increase in objective function as high as 45% in some scenarios while sampling less than 1 out of five million possible paths, 0.000019% of the 1,058,826,559,993 possible deployment paths. Our network flow deployment model produces the best results by exploring well over a trillion possible deployment paths giving us up to a 50% increase in objective value for some scenarios. The network flow deployment model has the fastest solve time when compared to stack-based enumeration and random path generation, solving most scenarios within one hour.

THIS PAGE INTENTIONALLY LEFT BLANK

ACKNOWLEDGMENTS

I would like to thank all the NPS staff and professors who have helped shape me throughout this course work. I especially would like to thank Professor Gerald Brown for his help and guidance throughout the thesis process. I do not believe I would have been able to accomplish as much without his continued support and patience. My success with my thesis is a direct reflection of his profound knowledge of optimization practices. Thanks to Professor Carlyle for the time he put into shaping my model. His knowledge of Python and Pyomo was crucial to my success throughout my thesis. I would also like to thank Captain Jeff Kline for getting me set up with this thesis. His continued involvement in the surface warfare community is greatly appreciated.

Mostly, I would like to thank my lovely wife, Ashleyann. She is the one who had to deal with me the most throughout this process and my career, and she did so with love and patience even if “this was not the shore duty you were promised.” I would not be where I am today without your love and support. I would also like to thank my children, Gabriel, Gracemarie, Jessicalynn, and Lillyann, for bringing me joy in every day.

THIS PAGE INTENTIONALLY LEFT BLANK

I. INTRODUCTION

Military planning is complicated, but commanders and staff at all levels have to make informed decisions and solve complex problems with the goal of operational success. Planning like this is crucial across all levels of warfare (strategic, operational, and tactical) and can often be manually labor intensive. Specific to the Navy Planning Process (NPP), a commander takes a current state and develops plans to move to a desired end state (Department of the Navy, 2013a). Many considerations are taken into account including the operational environment, logistics, and mission capabilities of available ships.

The joint Force Maritime Combatant Commander (JFMCC) and staff continuously plan and manage maritime operations. Joint Publication 5-0 (Joint Chiefs of Staff, 2011) outlines this planning as deliberate in determining the ways, means, ends, risk, and the operational art, for the accomplishment of specified national strategic objectives. For a commander to understand operational art, one must think of military planning as both a science and an art. The science involves tangible aspects such as number of ships, weapons, supplies, and consumption rates as well as operational factors of time and space. For military planning, art is more conceptual: commanders follow the principals of war when they think through their application and design their plan. Operational art allows for subordinate commands to understand and operate independently, while following their superior commander's intent.

A. NAVAL TACTICAL PLANNING

Military planning has evolved over the years and is essential to how we manage today's Navy. The Navy must be prepared to simultaneously confront a wide range of dynamic situations. With more diverse mission sets, this often places commanders even at the lowest tactical levels into critical strategic roles in planning. Naval operations require specific threat-based planning involving mission, environment, and threat scenario. Any level of naval planning in a contested environment requires the need for our joint force to gain and maintain sea control, and allows JFMCC to focus on threat-

based planning (Department of the Navy, 2013a). The idea is to achieve the best chance of mission accomplishment when plans are orchestrated and executed correctly.

The Navy's attempt to mitigate any shortfalls in command and control at operational levels of war are addressed by standing up the Maritime Operation Center (MOC). In support of the commander's decision cycle (Department of the Navy, 2013b), the MOC's goal is to provide structure and act as an extension to the commander. The structure and organization of a MOC must be able to adapt to the changing mission assignments, the potential threats that advisories impose, where we conduct operations, and the time required to reach an end state. All numbered fleets, Navy component command headquarters, and JFMCC staff (if designated) have an established MOC. When a commander is directed to stand up a Joint Task Force (JTF) the MOC still remains the center for organization and planning. Command relationships are then determined by the JTF and commanders will maintain control of attached Navy forces. Monitoring, planning, directing, assessing, and communicating for the commander will be performed by the MOC. Then subordinates to the commander are placed in charge of tactical planning and execution (Department of the Navy, 2013b). MOC's continued support to commanders in planning is an invaluable asset for the Navy in combating continual threats.

B. MARITIME PLANNING

When planning down to the level of a Carrier Strike Group (CSG), Expeditionary Strike Group (ESG), or Destroyer Squadron (DESRON), tactical commanders are given the proper tools in order to follow the superior's intent and objectives.

Ultimately, our goal is not to create perfect situational awareness, but rather to leverage the most important commodity to a commander—time—to think through multiple concepts of operations (CONOPS) which can be used to jump start subordinate planning as battlefield conditions and objectives become more apparent. In the cognitive domain, time is a catalyst that, when mixed with a proper bit of trust, experience, intelligence, and planning, is a recipe for creating asymmetric options and controlling tempo on the modern maritime battlefield. (Swift, 2018)

When planning at the tactical level, commanders must be able to make quick and deliberate decisions in order to achieve mission success. There are many considerations when assigning a ship to a number of sets of missions, each called here a “combined mission capability” set. These combined mission capability (CMC) sets can include simultaneous conduct of parallel individual missions, such as, say, Air Defense (AD) and Maritime Interdiction Operation (MIO). Mission requirements can span several days within the same geographical region.

One of the more important considerations when planning is logistical support. The planner will have available Combat Logistics Force (CLF) ships in the area that the commander may utilize for underway replenishment of fuel and stores. All these considerations make maritime combat operations planning a long and scrupulously detailed task.

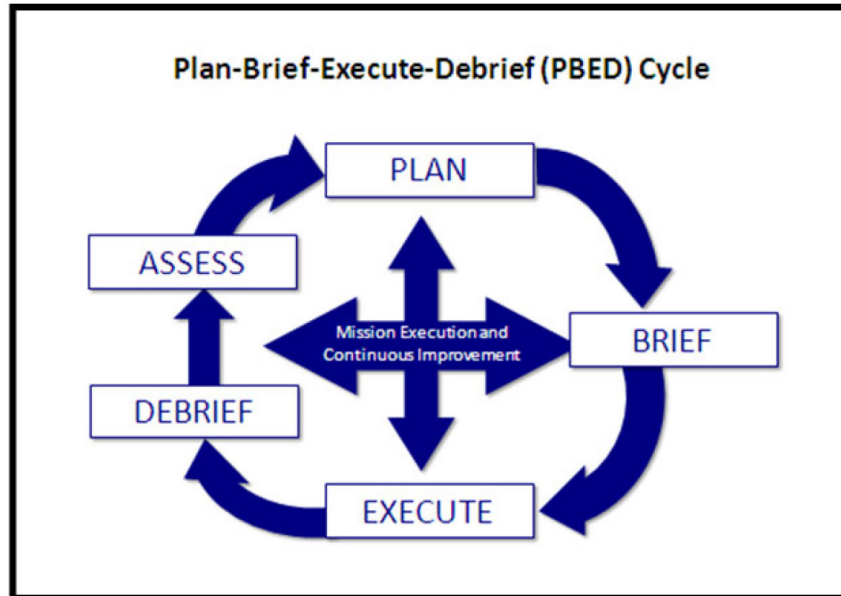
Another aspect of maritime planning is that some individual missions have prerequisite missions. A prerequisite mission must be executed for its dependent mission(s) to be feasible. For instance, an Air Defense (AD) mission may be required as a prerequisite for an Anti-Submarine Warfare (ASW) one. The AD mission and ASW one may, or may not, be feasible for a single ship as one of its combined mission capabilities. Simultaneous AD and ASW missions may require two ships operating in tandem.

A planning scenario may involve many combatants; each ship arriving on its own schedule at some location and time; and many geographically dispersed missions. Most of this planning is still manual, and there is a real and present need for a decision support system able to be distributed and utilized by commanders and their planning teams.

C. NAVY PLANNING PROCESS

The Navy Planning Process (NPP) is a six-step critical thinking aid in which commanders and decision makers will conduct mission analysis, course of action development, course of action analysis (wargaming), course of action comparison and decision, plan or order development, and transition. With integration into the command’s battle rhythm, the NPP is structured so the commanders can interact with the planning team and vice versa. At the tactical level, they do not utilize the NPP but rather an

organizational tool of Plan, Brief, Execute, and Debrief (PBED) cycle (See Figure 1). This aids tactical commanders for planning successful missions.



This illustrates an organizational tool to aid tactical commanders to plan for successful completion of missions.

Figure 1. Plan-Brief-Execute-Debrief Cycle. Source: Department of the Navy, 2013a.

The four steps in the PBED cycle utilized by tactical commands is outlined in NWP 5-01 (Department of the Navy, 2013a):

1. **Planning:** A crucial element for mission success. Planning covers everything from any minute [sic] detail to larger more complex operations or exercises that may require higher-level attention.
2. **Briefing:** A communication tool to ensure the commander and subordinates are on the same page. Commanders should ensure that all personnel involved in the execution of the plan being briefed are present. This allows walk-through or talk-through and ensures everyone

understands their role and responsibility in order to provide forceful back-up.

3. **Executing:** If a mission has been planned and briefed properly, that mission is set up for successful execution, allowing for the personnel executing the plan to adjust and correct any challenges that may arise during the evolution.
4. **Debriefing and Feedback:** At the end of every evolution commanders and their subordinates should cover any lessons learned during the evolution and cover any key facts. This feedback process is then used in the planning of future evolutions.

D. BACKGROUND AND PRIOR WORK REVIEW

1. Naval Logistics

In order for the U.S. Navy to maintain a high state of readiness, the scheduling of Replenishments at Sea (RAS) for combatant ships will always be a top priority. Globally there are approximately 30 special transport ships that comprise the Navy Combat Logistics Force (CLF). Each CLF transportation ship deploys from tankers or ports and is capable of carrying various cargoes to include Distilled Fuel Marine (NATO F75/76) (DFM), navy Aviation Fuel (NATO F44) (JP5), ordnance, repair parts, food and dry stores, and mail. The Replenishment at Sea Planner (RASP) (Brown et al., 2018) suggests an employment schedule for each CLF ship using an integer linear optimization. RASP alternately reduces the (peace-time) fuel consumption of the CLF, or maximizes the war-time volume of deliveries, and saves the U.S. Navy millions of dollars each year by suggesting rendezvous locations with U.S. combatants that are underway. This operational planning tool covers a planning horizon of several weeks with four-hour watch rotation time increments. RASP serves two purposes: it helps planners generate a daily schedule over a short planning horizon and displays the intended CLF movements on Google Earth, and RASP automates messages required for continuous logistics system operations.

2. Navy Mission Planner

Navy Mission Planner (NMP) (Dugan 2007) is a decision aid that quickly develops multiple Courses of Action (COAs) for the deployment scheduling of combatant ships over about a 20-day planning horizon. NMP is our optimizing decision aid that accommodates multiple ships and multiple missions with the goal maximizing the total value of mission accomplishment. NMP relies on an integer linear program that assesses the assignment of ships to missions. NMP utilizes a Microsoft Excel (2016) user interface with Visual Basic for Applications (VBA) (VBA Programming Office, 2019) to display scenario data and automatically generate extracts of a scenario as Comma Separated Value (CSV) (Wikipedia contributors, 2019) files for inputs into a commercial optimization solver. NMP generates a potentially large pool of employment schedules for each combatant by a path enumeration over all the possible regions to travel, using input missions, their regions, and time periods over the time horizon. NMP chooses from the portfolio of alternate deployments for each combatant the best one that, combined with the best deployments for all other available combatants, satisfies mission dependencies over the planning horizon while maximizing mission value attained. The results of the NMP lead to a near-optimal set of employment schedules and the missions that can or cannot be accomplished. NMP allows planners to quickly solve, adjust, and resolve for multiple COAs.

Silva (2009) continues Dugan's NMP work by constraining the path enumeration to restrict the maximum number of schedules, maximum schedules per ship, and maximum stall (days in which a ship can loiter in a region). For each combatant, a directed, acyclic deployment network is generated with nodes for location-days, and arcs connecting pairs of adjacent nodes between which the combatant might travel. A stack-based, depth-first search (details to follow) is employed to discover directed paths a ship could take through its deployment network. Because there is an exponential number of such paths, the search is truncated by a user-defined maximum number of paths per ship. This reduces computational burden by restricting the number of paths created with the goal of still offering a near-optimal solution.

Hallmann (2009) increases the capabilities of the NMP by adding logistic planning capabilities, not just providing an employment schedule for each combatant ship, but for the attending Combat Logistics Force (CLF) as well. The user inputs allow for which CLF ships are available and capable of providing logistical support to the combatants taking into consideration the planning time horizon, geographical regions, commodities, consumption factors, and inventory thresholds. The commodities on hand each customer combatant are illustrated with a saw-tooth diagram tracking the daily consumption of the four main consumable items onboard a combatant ship (DFM, JP5, stores and ordnance), and resupply increments, a useful tracking aid for decision makers.

Another useful tool Hallmann introduces is a new mission set of ships not capable of self-defense in regions requiring defensive escorts or close escorts. “Close escort” requirements stipulate that a defending combatant follow the escorted ship region-to-region, while the less restrictive “escort” restriction merely requires a defending combatant to be in the same region at the same time. This also introduces underway replenishment as a mission in which commodities can be transferred from CLF ships to customer ships. This practical decision aid proved to be useful during the planning efforts of the Navy’s Fleet Forces Command exercise, Trident Warrior 2009 (Hallmann, 2009).

Pearlswig (2013) continues to shape the NMP by developing a myopic heuristic route generator that reduces computational runtime by selecting one route per ship based on mission values per region. This slightly improves the solutions values of the NMP.

Deleon (2015) modifies NMP to develop a Navy Operational Planner (NOP) specifically for Mine Countermeasures (MCM). He includes specific factors that pertain to MCM missions: probability of detection, area of the minefield, sensor search width, and sensor search speed. Other factors included in his model are number of missions, cool down rates (i.e., the rate at which a cleared area might again be infested with mines), number of ships available for a given time period, the MCM ship degradation rate (rate of mission completion prior to interruption), and accomplishment threshold (a percentage level at which the mission is complete). The results of NOP assesses the time it would take to complete a MCM mission phase (set of missions required to move to the next phase).

THIS PAGE INTENTIONALLY LEFT BLANK

II. NAVY MISSION PLANNER

Due to developing technology and how that affects warfighting, there is a continual increase in how many missions the Navy must perform and too often there are more missions needing accomplishment than ships available. Missions have become multifaceted, often requiring complex configurations of multiple ships or one ship conducting multiple missions. A mission can span the entire length of the planning horizon or be as short as one watch rotation. Some ships require escorts from defending combatants in order to complete their missions. When decision makers analyze missions and the assets available, the task of deciding who goes where and when can be difficult.

A. AN INTEGER LINEAR PROGRAM TO OPTIMIZE NAVY MISSION PLANNING: NMP AND LOGISTIC SUPPORT

The model presented here derives from the purely operational Navy Mission Planner (NMP) introduced by Dugan (2007), the embellished version introduced by Silva (2009), and the one (followed closest here) by Hallmann (2009). The logistic portion of the new planning tool is inspired by the Combat Logistics Force (CLF) planning tool by Brown & Carlyle (2008) and more recent Replenishment At Sea Planner (RASP) by Brown, et al. (2017).

Navy Mission Planner anticipates three levels of advice.

1. The least complicated anticipates a set of spatially diverse missions in an Area of Responsibility (AOR), each with an anticipated execution day over the next few weeks. These missions are to be completed by armed Navy combatants (e.g., CG, DDG, LCS, MCM, CVN) as they become available day-by-day in the AOR. Geography is important, and the transit times from one location to another to complete various missions is a key consideration. Missions also have dependencies among them. For instance, an Air Defense mission may be required in some particular location to cover an Antisubmarine Warfare mission. Combatants are capable of performing more than one mission simultaneously, but with

varying degrees of effectiveness depending on the simultaneous mission mix and the particular combatant. Combatants must be scheduled for necessary logistics (e.g., refueling) missions at particular locations and times.

2. The next level of complication considers supplying combatants from Combat Logistics Force supply ships (e.g., T-AKE, T-AO, T-AOE; “delivery boy” sorties by supply ships to the combatants, or combatant customer visits to CLF “gas station(s)”). These undefended ships may require combatant escorts to visit certain locations, and such escort activities may be within the same area (e.g., Air Defense), or necessarily in close company of a combatant (e.g., Antisubmarine Warfare.)
3. The final complication is inclusion of unarmed (or lightly armed) combatants (e.g., LPD, LHD, LHA) that require armed combatant escorts.

In the simplest case (1), the mission sets have been worked out ahead of time and the remaining questions are which ships to assign to each location each day (“geotime”) to complete as many missions on time as possible. Missions have varying value, and we seek to maximize the total value we achieve while satisfying constraints on mobility, simultaneous and conditional mission completion, and varying effectiveness of our combatants and their assignments.

In case (2) we add logistics ships with their own mobility and commodity limitations.

Finally, case (3) can include in the mission set an increased diversity, including such things as an amphibious assault. This case would be most useful for early net assessment.

We anticipate that planners would start with case (3), then refine to case (2), and finally specify the details of case (1) at successively lower-level operational command planning.

The following integer linear program, NMP with logistics, seeks the best achievable set of (combatant and CLF) ship deployment schedules, and is derived from Hallmann's (2009) model:

Sets and Indices [Cardinality]

$s \in S$	Ship (by hull number and name) [~ 90]
$cs \in CS \subseteq S$	Defended combatant Navy ship (e.g., CG, DDG, LCS) [~ 40]
$ns \in NS \subseteq S$	Undefended combatant Navy ship (e.g., LHA, LPD, MCM) [~ 40]
$ss \in SS \subseteq S$	Supply ship [~ 10]
$sx \in SX \equiv CS \cup NS \subseteq S$	Ships that can complete combat missions
$se \in SE \equiv NS \cup SS \subseteq S$	Ships that may require escorts
	$(CS \cup NS \in \text{Navy ships}, CS \cap NS \cap SS = \emptyset, CS \cup NS \cup SS = S)$
$m \in M$	Mission type (alias m') [~ 12] (e.g., ASW, AAW, NSG, ..., CAN_HIT, ESCORT, CLOSE_ESCORT)
$c \in C_s$	Combined (simultaneous) mission capability set for ship s [~ 10]
$m \in M_c$	Mission types in combined (simultaneous) mission set c (e.g., ship s can simultaneously perform mission types m in combined mission capability set c . (Note: non-combatant ships are only endowed with the "CAN_HIT" mission capability.)
$p \in P$	Employment schedules, alias p' [~ 1 million]
$p \in P_s \subseteq P$	Employment schedules for Navy ship s [~ 1 million] $(\bigcup_s P_s \equiv P, P_s \text{ is a partition of } P.)$

- $r \in R$ Regions in Area of Operations (alias $r1, r2$) [~ 30]
- $r \in RCS \subseteq R$ Regions navigable by armed combatant ships (CS)
- $rss \in RSS \subseteq R$ Regions navigable by unarmed ships (NS $\cup$ SS), some only if escorted by armed combatant ships
- $rloc \in RLOC = RCS \cap RSS \subseteq R$ Regions navigable by all
- $rssx \in RSSX \subseteq R$ Regions navigable by unarmed ships (NS $\cup$ SS) only if escorted by CS
- $rse \in RSE \equiv RSS \setminus RSSX$ Regions always navigable by unarmed ships (NS $\cup$ SS)
- $d \in D$ Days in planning horizon (alias d', d'' , $d1, d2$) [~ 14]
- $r(p, d)$ Region employment schedule p visits on day d
- $origin(s, d, r)$ Ship s comes into our control at the start of day d in region r .
- $n \in N$ Ordinal for multiple missions of the same mission type [~ 5]
(E.g., several ships may conduct ASW at the same time within the same region, but with different effectiveness.)
- $\{m, m'\} \in Q_{r,d}$ In region r on day d , mission m can be undertaken only if mission m' is fully accomplished
- $i \in I$ Commodity category (e.g., DFM, JP5, STOR, ORDN)

Data [Units]

$value_{m,n,r,d}$ Priority of n -th mission of type m , in region r on day d [1-10]
[value]

($\{m, n, r, d\} \in MNRD$ tuples exist only for non-zero values)

$accomplish_{c,m}$	Level of accomplishment of combined mission set $c \in C_s$, mission $m \in M_c$ [0.0-1.0] (Note that each ship may have its own set of combined mission capability sets, and that some of these sets may contain the same missions, but with different accomplish rates to represent the ship choosing to change emphasis between missions.)
$cap_{s,i}$	Capacity of ship s for commodity category i [i -units]
$init_load_{s,i}$	Initial load of ship s , commodity i [fraction of $cap_{s,i}$]
$use_{s,c,i}$	Daily consumption of commodity i by Navy ship s employing combined mission capability c . [i -units]
$safety_i$	Safety stock fraction of capacity for commodity i [fraction]
$extremis_i$	Extremis stock fraction of cargo category i [fraction] ($0 < extremis_i < safety_i < 1$)
pen_safe_i	Penalty per unit of violation of safety stock for commodity i [value/ i -unit]
pen_extr_i	Penalty per unit violation of extremis stock for commodity i [value/ i -unit]
pen_out_i	Penalty per unit violation below zero stock for commodity i [value/ i -unit] ($pen_out_i > pen_extr_i > pen_safe_i > 0$)
use_supply_ships	Indicates that supply ship employment scheduling and combatant commodity inventories should be included in employment plan [binary]

escort_required Indicates that every supply ship or unarmed combatant needs regional escort presence by armed combatants every day it is deployed [binary]

close_escort_required Indicates that every supply ship or unarmed combatant needs close escort by some dedicated armed combatant each day it is deployed [binary]

armed_escorts Restricts ships to enter regions with insufficient defense armament unless escorted by others with such armament

force_ratio_r force_ratio armed combatants to other escorted ships in region r

Induced Index Sets

$\{m, n, r, d\} \in MNRD$ 4-tuple exists only if $value_{m,n,r,d} > 0$ or $accomplish_{s,m} > 0$ for some ship that can employ a combined mission capability set that includes mission m in region r on day d

$\{m, r, d\} \in MRD$ 3-tuple exists only if $\{m, n, r, d\} \in MNRD$ does for some n

Variables [Units]

$U_{m,n,r,d}$ Level of accomplishment of the n -th mission type m assignment in region r on day d [0.0-1.0]

$V_{m,r,d}$ = 1 if mission m is fully accomplished in region r on day d [binary]

$W_{s,c,r,d}$ = 1 if ship s employs combined mission capability c on day d [binary]

$X_{se,cs,r,d}$ = 1 only if ships se and cs are both in region r on day d [0.0-1.0]

Y_p = 1 if schedule p is selected [binary]

$XFER_{ss,sx,d,i}$	Volume of commodity i transferred from supply ship ss to ship sx on day d [i -units]
$SLACK_{sx,d,i}$	Combatant sx , day d , commodity i stock in excess of safety-stock [I units]
$V_SAFE_{sx,d,i}$	Violation of safety stock level for combatant sx , day d , commodity i [i -units]
$V_EXTR_{sx,d,i}$	Violation of extremis stock level for combatant sx , day d , commodity i [i -units]
$V_OUT_{sx,d,i}$	Violation of positive stock level for combatant sx , day d , commodity i [i -units]
$ARMED_{r,d}$	Number of armed combatants in region r during day d [ships]
$UNARMED_{r,d}$	Number of unarmed ships in region r during day d [ships]

Sampled-Path Formulation

Stack-based enumeration (see Figure 2) can be used to sample a subset of the exponential number of paths $p \in P$. For any such subset, the following linear integer formulation evaluates a restriction of the problem with all paths present.

```

for i = 1 to n:
    onPath(i) = 0;
top = 1;
PATH[1] = 1;
onPath[1] = top;
next_arc[1] = point[1]
while top > 0:
    i = PATH[top];
    # (option to randomize array head[arc] for
    # arc = point[i], ..., point[i+1]-1)
    while next_arc[i]<point[i+1]:
        j = head[next_arc[i]]
        next_arc [i] = next_arc[i]+1
        if (onPath[j] == 0 and OK_to_add(j)):
            top = top+1
            PATH[top] = j
            onPath[j] = top
            next_arc[j] = point[j]
            if j == n:
                print(PATH[])
                onPath[n] = 0
                top = top-1
            i = PATH[top]
    onPath[PATH[top]] = 0
    top = top - 1

```

Figure 2. Algorithm Enumerating all the (finite number of) s-t Paths of Length Bounded by T. Source: Brown et al., 2013, p. 46.

$$\begin{aligned}
\max \quad & \sum_{\{m,n,r,d\} \in MNRD} value_{m,n,r,d} U_{m,n,r,d} \\
& - \sum_{sx \in SX, d \in D, i \in I} pen_safe_i V_SAFE_{sx,d,i} \\
& - \sum_{sx \in SX, d \in D, i \in I} pen_extr_i V_EXTR_{sx,d,i} \\
& - \sum_{sx \in SX, d \in D, i \in I} pen_out_i V_OUT_{sx,d,i} \tag{T0}
\end{aligned}$$

$$\text{s.t.} \quad \sum_{p \in P_s} Y_p \leq 1 \quad \forall s \in S \tag{T1}$$

$$\begin{aligned}
\sum_{c \in C_{sx}} W_{sx,c,r,d} &= \sum_{\substack{p \in P_{sx} \\ |r(p,d)}} Y_p \quad \forall sx \in SX, d \in D, \\
& r \in RLOC \tag{T2}
\end{aligned}$$

$$\begin{aligned}
\sum_{n|\{m,n,r,d\} \in MNRD} U_{m,n,r,d} &\leq \sum_{sx \in SX, c \in C_{sx}} accomplish_{c,m} W_{sx,c,r,d} \\
& \forall \{m,r,d\} \in MRD \tag{T3}
\end{aligned}$$

$$V_{m,r,d} \leq \sum_{n|\{m,n,r,d\} \in MNRD} U_{m,n,r,d} \quad \forall \{m,r,d\} \in MRD \tag{T4}$$

$$\begin{aligned}
U_{m,n,r,d} &\leq V_{m',r,d} \quad \forall \{m,n,r,d\} \in MNRD, \\
& m' \in \{m, m'\} \in Q_{r,d} \tag{T5}
\end{aligned}$$

Constraints (T6)-(T14) activated by *use_supply_ships*:

$$\begin{aligned}
\sum_{\substack{p \in P_{se} \\ |r(p,d)}} Y_p &\leq V_{ESCORT',r,d} \quad \forall se \in SE, r \in R, d \in D \\
& |escort_required \tag{T6}
\end{aligned}$$

$$\begin{aligned}
\sum_{\substack{p \in P_{se} \\ |r(p,d)}} Y_p &\leq \sum_{\substack{cs \in CS, \\ "CLOSE_ESCORT" \in M_c}} W_{cs,c,r,d} \quad \forall se \in SE, r \in R, d \in D \\
& |close_escort_required \tag{T7}
\end{aligned}$$

$$X_{se,cs,r,d} \leq \sum_{p \in P_{cs}|r(p,d)} Y_p \quad \forall se \in SE, cs \in CS, \quad r \in R, d \in D \quad (T8)$$

$$X_{se,cs,r,d} \leq \sum_{p \in P_{se}|r(p,d)} Y_p \quad \forall se \in SE, cs \in CS, \quad r \in R, d \in D \quad (T9)$$

$$X_{se,cs,r,d} \leq \sum_{\substack{c \in C_{cs} \\ |CAN_Hit| \in M_c}} W_{cs,c,r,d} \quad \forall se \in SE, cs \in CS, \quad r \in R, d \in D \quad (T10)$$

$$XFER_{ss,sx,d,i} \leq \sum_{r \in R} \min(init_load_{ss,i}, cap_{ss,i}, cap_{sx,i}) X_{ss,sx,r,d} \quad \forall ss \in SS, sx \in SX, \quad d \in D, i \in I \quad (T11)$$

$$\sum_{\substack{sx \in SX, \\ d' \leq d}} XFER_{ss,sx,d',i} \leq init_load_{ss,i} cap_{ss,i} \quad \forall ss \in SS, d \in D, i \in I \quad (T12)$$

$$\begin{aligned} \sum_{\substack{ss \in SS, \\ d' \leq d}} XFER_{ss,sx,d',i} - \sum_{\substack{c \in C_{sx}, \\ r \in R, \\ d' \leq d}} use_{sx,c,i} W_{sx,c,r,d'} \\ + SLACK_{sx,d,i} + V_SAFE_{sx,d,i} + V_EXTR_{sx,d,i} \\ = (1 - init_load_{sx,i}) cap_{sx,i} + V_OUT_{sx,d,i} \end{aligned} \quad \forall sx \in SX, d \in D, i \in I \quad (T14)$$

Constraints (T15) activated by *armed_escorts*:

$$\sum_{\substack{cs \in CS, \\ p \in P_{cs}|r(p,d)}} Y_p \geq force_ratio_r \sum_{\substack{se \in SE, \\ p \in P_{se}|r(p,d)}} Y_p \quad \forall r \in R, d \in D \quad (T15)$$

$$U_{m,n,r,d} \in [0,1] \quad \forall \{m,n,r,d\} \in MNRD$$

$$V_{m,r,d} \in \{0,1\} \quad \forall \{m,r,d\} \in MRD$$

$$W_{sx,c,r,d} \in \{0,1\} \quad \forall sx \in SX, c \in C_{sx}, d \in D$$

$$\begin{aligned}
X_{ss,cs,r,d} &\in [0,1] & \forall ss \in SS, cs \in CS, \\
& & r \in R, d \in D \\
Y_p &\in \{0,1\} & \forall p \in P \\
XFER_{ss,sx,d,i} &\in [0, \min(cap_{ss,i}, cap_{sx,i})] & \forall ss \in SS, sx \in SX, \\
& & d \in D, i \in I \\
SLACK_{sx,d,i} &\in [0, (1 - safe_i)cap_{sx,i}] & \forall sx \in SX, d \in D, i \in I \\
V_SAFE_{sx,d,i} &\in [0, (safe_i - extremis_i)cap_{sx,i}] & \forall sx \in SX, d \in D, i \in I \\
V_EXTR_{sx,d,i} &\in [0, extremis_i cap_{sx,i}] & \forall sx \in SX, d \in D, i \in I \\
V_OUT_{sx,d,i} &\geq 0 & \forall sx \in SX, d \in D, i \in I \text{ (T16)}
\end{aligned}$$

Discussion

The following descriptions with the exception of constraints (T15) is verbatim from Hallmann (2009):

The objective (T0) measures the weighted value of (partially) completed missions. Each (packing) constraint (T1) allows at most one employment schedule per ship. Each constraint (T2) permits a combatant to employ a combined mission capability on a day only if an employment schedule has been chosen for that ship. Each constraint (T3) bounds the sum of the partial completion values of all instances of a given mission, in a given region on a given day, by the total amount of activity for that mission in the region. Each constraint (T4) allows a task to be considered fully completed in a region on a given day if there is at least one total unit of activity for that mission in that region on that day. Each constraint (T5) allows activity in a region, mission, and day, only if a prerequisite mission in that region on that day has been fully accomplished. If close escort is not required, each constraint (T6) permits a supply ship to enter a region requiring escort only on a day for which the “ESCORT” mission has been fully accomplished there; if the “ESCORT” mission has been completed in a region, any number of supply ships may enter the region. If close escort is required, each constraint (T7) requires that the number of supply ships in a region on a day is limited by the level of accomplishment of the “CLOSE_ESCORT” mission in that region that day; this means that there

will be at least one combatant per escorted supply ship. Each constraint (T8) permits location of a supply ship for commodity transfer in a region of a selected employment schedule. Each constraint (T9) does this for a combatant, and each constraint (T10) allows collocation with a combatant only if the combatant employs the mission in the combined mission capable set. Each constraint (T11) limits transfer of a commodity between a supply ship and a combatant to a day when the ships are collocated in the same region. Each constraint (T12) limits the deliveries a shuttle ship can make during any epoch after a port visit to resupply. Each constraint (T14) accounts for a cumulative commodity used by a ship up to the end of a given day, and reckons any shortage below safety-, extremis-, or zero-stock levels (Note that any such shortage will be carried forward to later days until it is remedied by commodity transfer).

Each constraint (T15) ensures that a voyage for a ship within a region for which it is insufficiently armed is accompanied by some number of ships in that region with sufficient armament. Constraints (T15) determine the ratio of the minimum number of armed ships per unarmed ship in each region, each day. Variable domains are defined by (T16).

B. NETWORK FLOW FORMULATION

In lieu of the sampled-path formulation, which grows in size as the number of sampled paths approaches the exponential number extant, we can formulate an integer linear program that includes all paths in its feasible solutions, but perhaps also paths that are not admissible as mere network flows.

Additional Index Definitions

(s, d, r) Node: define for a ship s the days d and regions r that can be occupied

$arcs_{s,d1,r1,d2,r2}$ for ship s , adjacent nodes $(d1, r1)$ and $(d2, r2)$.

Additional Decision Variables

$FLOW_{s,d1,r1,d2,r2} = 1$ if path chosen from $(d1, r1)$ to $(d2, r2)$, 0 otherwise [binary]

$PRESENT_{s,d,r} = 1$ if a selected path passes through node (d, r) , 0 otherwise
[binary]

We define a deployment flow network for each ship, starting at the $origin(s, d, r)$ where it appears and comes into our scheduling control. Because our optimization is free to leave each ship at any location, we introduce artificial arcs from every location $(d1, r1)$ from which there is no destination (i.e., the out-degree is zero) (Ahuja et al. 1993, p. 35ff). These artificial arcs terminate the network flow at node $destination(s, d, r)$.

Replace constraint (T1) by a set of conservation-of-flow network constraints:

$$\begin{aligned} \sum_{arcs_{s,d,r,d2,r2}} FLOW_{s,d,r,d2,r2} - \sum_{arcs_{s,d1,r1,d,r}} FLOW_{s,d1,r1,d,r} \\ = \begin{cases} +1 \text{ } origin(s, d, r) \\ 0 \\ -1 \text{ } destination(s, d, r) \end{cases} \quad \forall s \in S, d \in D, r \in R \end{aligned} \quad (F1)$$

Equivalences

$$PRESENT_{s,d,r} \equiv \sum_{arcs_{s,d,r,d2,r2}} FLOW_{s,d,r,d2,r2} \equiv \sum_{\substack{p \in P_s, \\ r(p,d)}} Y_p .$$

Although it is possible to formulate either the sampled-path or the network flow model in terms of the binary variables $PRESENT$, this would obscure algebraic forms that our integer linear program solvers (and their presolve functions, e.g. CPLEX 2019) use to simplify formulations that necessarily contain scores of redundant constraints and variables. These presolve functions look for specific structure evident in the “set packing” constraint set (T1), or in the network flow constraints (F1), so we want to leave these, as is. Accordingly, for the alternate sampled-path and network flow models, we simply substitute the algebraic equivalents of expressions involving decision variable Y and $FLOW$.

THIS PAGE INTENTIONALLY LEFT BLANK

III. SOLUTION GENERATION FOR NAVY MISSION PLANNER

A. PYTHON AND PYOMO WITH CBC (COIN-OR BRANCH AND CUT) SOLVER

Previous work to the Navy Mission Planner utilized the General Algebraic Modeling System (GAMS) software (GAMS, 2019) and any of a variety of optimization solvers, such as CPLEX (CPLEX, 2019) for the optimization models. This is an effective and efficient mathematical programming and optimization software suite. Unfortunately, this proprietary code is not “open source” and is not an easy candidate to qualify for DoD or DoN classified computing systems or for Navy Marine Corps Internet (NMCI) computers. Use of this software outside academia requires individual commercial licensing. GAMS and a reliable solver would require a commercial license for each individual planning computer at a Carrier Strike Group (CSG), Destroyer Squadron (DESRON), or any military command desiring to utilize the NMP. The baseline cost for GAMS single user license is about \$3 thousand and this is the cost prior to any solvers (GAMS, 2019). With previous NMP models the solver used with GAMS was CPLEX. According to the IBM website CPLEX solver is faster on average by 23%, compared to open-source solvers, when dealing with large mixed integer programming model problems (>1000s of variables) such as the NMP. The addition of the CPLEX solver to purchasing of GAMS raises the license per-seat price to about \$10 thousand U.S. dollars.

Though GAMS and CPLEX are not open-source, they have been approved for use in some classified DoD computers. Such approvals are local decisions by information technology managers, and not easy to win for sites removed from research and development.

Shifting NMP to an open-source modeling and solver suite, with no license fees, and affording the ability to scan all underlying computer code for forbidden supervisor calls or other security-threatening mischief would mean that NMP could be widely adopted by Navy networks, including classified ones. Python (Python Core Team, 2019) is open-source. Within Python, the Pyomo (Hart, 2017) mathematical modeling language (also open source) offers structured optimization applications similar to that of GAMS.

Pyomo has been developed by Sandia National Laboratories as a COIN-OR (Computer Infrastructure for Operations Research) project (Hart, 2017). The COIN-OR foundation is a non-profit educational and scientific organization with the goal of creating and disseminating knowledge related to computational operations research. Like GAMS, Pyomo can be used to express general optimization problems, create the required problem instances, and solve these instances with a solver. Also, from the COIN-OR foundation there is a COIN-OR Branch and Cut (CBC) solver. This (open-source) solver is written in C++ and works well with Pyomo, GAMS, and other optimization applications (COIN-OR, 2019).

Herein, we compare PYTHON-PYOMO using the CBC solver with GAMS using CPLEX. NMP uses a graphical user interface implemented in Excel (Dugan, 2007). The GAMS-CPLEX solver depends on Visual Basic for Applications (VBA programming in Office, 2019) to extract data in files GAMS can read, and to import solutions back from GAMS into the interface. Shifting NMP to Python-Pyomo allows for access to multiple library's and packages that change how information can be imported and exported. One of these packages is xlwings (xlwings, 2019) that is designed to replace VBA code with PYTHON subroutine calls or vice versa by providing a direct interface with Microsoft Excel. With Python-Pyomo and the use of xlwings, python is able to import data directly from the NMP excel user interface. This also allows for easy outputs and can directly inject solutions into a Microsoft Excel file with multiple tabs, in contrast to the tedious import of GAMS-generated Comma Separated Value (CSV) files (*Wikipedia*, 2019) generated by the previous NMP models.

B. RANDOM EMPLOYMENT PATH GENERATOR

Previous works with NMP have used a brute-force, exhaustive stack-based path enumeration (Silva, 2009) or a myopic, greedy heuristic (Pearlswig, 2013) for path generation. The brute-force generation utilizes a stack-based depth-first search enumeration (Figure 2) across a network-based forward-star representation adjacency list (Ahuja et al. 1993, p. 35ff) until some maximum number of paths is reached. (Following, we show an NMP example with both this sample size and the total number of paths

available.) The greedy (Pearlswig) heuristic only generates one path for each ship based on a myopic maximum-valued path for a single ship. With brute-force enumeration the depth-first search method discovers a reasonable diversity of paths with a reasonable amount of path generation (say, ten thousand paths per ship). The greedy heuristic path generation would pick a path for each individual ship based on the highest locally visible adjacent mission accomplishment extending a subpath. Neither of these methods are optimal because of the small (well, infinitesimal) diversity in which paths are created.

1. Modifying NMP Path Generation

Silva's 2009 brute-force enumeration could eventually generate an optimal path, but that would require millions or even more paths generated. The stack-based enumeration employs depth-first search (Ahuja et al. 1993, p. 35ff.), which means early choices as paths are built are persistent through many-many subsequent path completions. This may be unfortunate if any early choice is a bad one.

Due to the backtracking and bookkeeping of this stack-based algorithm the path generation mimics the movements of an odometer, only changing the last day's region, one at a time, leaving most paths very similar to one another (i.e., the earlier a mission appears in a depth-first enumerated subpath, the longer it remains on all completions of that subpath into complete paths spanning the planning horizon; e.g., over a 15-day planning horizon with 16 regions) (Hallmann, 2009), the ten thousand subpaths never change over the first eight days for each ship and those subpaths often consist of eight days remaining in the same region. This method of path enumeration would not show diversity until the billions or even more paths are generated and computationally these large numbers would not be practical.

2. Modification of Backtracking Algorithm

Generally, the depth-first search with backtracking and bookkeeping follows a universal adjacency list pulling regions based on the next available region, until the entire list has been exhausted, and then the previous day's region will shift to the next region within its list (similar to how an odometer scrolls through numbers). The deployment graph for each ship is an acyclic directed graph lexicographically ordered by day and by

region. It is expressed in forward-star form (Ahuja et al. 1993, p. 35ff.). When the depth-first enumeration encounters a node, it must select from one of the adjacent nodes in its forward star (Ahuja et al. 1993, p. 35ff.). Rather than choose this adjacent node in fixed order, we randomize this choice. After finite amount of paths are generated using the brute-force stack-based enumeration chosen by the user (i.e., 15–50 enumerated paths) the adjacency list is then randomly shuffled. When the adjacency list is randomly shuffled this creates a uniquely diverse set of new paths each time changing almost every region after the start region. This method unfortunately allows for repeat paths. However, with a more diverse set of paths for each ship the NMP model would possibly find a more optimal result, even if there are fifty plus percent repeated paths.

Path duplication within VBA and GAMS are not admitted in the previous models, the GAMS software will reject duplicate paths.

Python has the ability to identify and remove duplicate paths with little computational workload, once the path list is placed in a pandas dataframe (a two-dimensional data structure within Python) (McKinney, 2010). This method does not guarantee an optimal result, and without controlling the amount of time spent in regions or taking other controlling measures the proportion of voyage path duplicates is high. We find that with the random shuffling of the adjacency list we can generate paths for each ship that, in concert with paths for sister ships, produce better suggested solutions.

3. Minimizing Time within a Region

By minimizing the time spent in a specific region we create a more diverse voyage plan and prevent a ship from loitering in one region for the first eight days, as seen in the brute force enumeration. This limit is a user input that bounds the number of successive days each ship can spend in any one region. Limiting this to four-to-six days will force the voyage plan to move the ship out of those regions earlier and prevent loitering for a long duration. By minimizing the amount of time spent within any region we greatly increase the diversity of voyage paths.

4. Preventing Restricted Regions

When generating routes for a ship requiring an escort in a region otherwise forbidden, we cannot know a priori whether such an escort will be available. Accordingly, we must generate such routes that include these escort-required regions in anticipation that they might be admissible later when we optimize and chose a route for each ship in an overall solution that may beneficially include ship-escorting-ship collocations. These regions vary from ship to ship depending on if they are a supply, combat, or unarmed ship. While enumerating through a voyage plan, if the ship is not allowed in a geographical region the algorithm will skip over that specific region within the adjacency list, leaving only paths in which the ships can actually transit due to missions or sensitive areas. For example, if a combat ship start region is homeport, for mission purposes that ship would not return to homeport, unless for unanticipated repairs or other reasons outside of NMP scope. The addition of this check will prevent that combat ship from returning to homeport and not add that specific region to that ship's voyage plan. This works the same for regions in which supply ships are forbidden even with escorts. This helps diversify the random paths generated.

C. PERSISTENCE IN OPTIMIZATION

Navy Mission Planning is like three-dimensional chess: We are moving discrete *ships* across *distance* and *time* to complete particular space-time missions (occupy positions on our chessboard) over a finite planning horizon, with side constraints on how and where we can move, and among the missions we can complete.

Such planning may be for purposes of assessment. We might ask questions such as:

- Can we complete this mission set over this time horizon with these ships?
- If another ship can be included, how does this influence our plan?
- If we lose a ship, or a ship is delayed arrival to the AOO, what influence does this have on our plan?
- Can we modify the mission schedule?

Many of these questions will arise as a plan is prepared long in advance of anticipated execution.

This prototypic research is also aimed at eventually providing optimization-based decision support to operational commanders when bad things happen during plan execution.

“No plan survives initial contact with the enemy.” (paraphrased from von Moltke, 1871)

Although optimization-based decision support can effectively solve complicated problems, it does have its limitations.

Most optimization-based decision support systems are used repeatedly with only modest changes to input data from scenario to scenario. Unfortunately, optimization (mathematical programming) has a well-deserved reputation for amplifying small input changes into drastically different solutions. A previously optimal solution, or a slight variation of one, may still be nearly optimal in a new scenario and managerially preferable to a dramatically different solution that is mathematically optimal. (Brown et al., 1997)

If we are deep in analysis of a plan under initial development, excessive numbers of revisions responding to slight refinements are an annoying distraction. Most planners would prefer to keep the parts they like, and merely improve those they do not.

If we are revising a plan already in execution, our ships have already been given deployment orders, perhaps loaded commodities in anticipation of carrying out certain missions, and may be underway or in engagement. The last thing we want is unnecessary turbulence leading to excessive messaging and confusion.

There are a number of ways to mitigate unnecessary changes and reduce turbulence between plan revisions, so-called “persistence” features in an optimization model.

The most severe persistent restriction is fixing a “presence” for a ship, location, and time period. If a planning system is being used over time, such fixing will be necessary as the near-term future becomes the present and past as we progress. If we are revising a future plan, we might find some actions so attractive and dominant they merit

fixing. For such future planning, moderation is a virtue: the last thing we want is to hobble our forces due to some unintended myopic restriction, or, worse, unintentionally render plans infeasible by inadmissible restriction (e.g., requiring underway speeds of 200 knots is inadmissible).

Although there are a number of techniques to introduce persistence (Brown et al. 1997), for purposes of illustration here we will focus on assignment of a ship on a day to a location. Assignments of other sorts of activities proceed in an analogous fashion.

Fixing a decision variable, i.e., $PRESENCE_{CG61,d8,r6} \equiv 1$, is trivial in a linear integer program, and much stronger than precluding some action, e.g., $PRESENCE_{CG61,d8,r6} \equiv 0$. Hallmann (2009) uses this expediency to fix the location of CLF shuttles for some time periods, creating “gas stations” for combatants.

A slightly less drastic restriction is to form a persistence constraint that senses changes to a legacy schedule.

Additional Data

$\widehat{presence}_{s,d,r} \in \{0,1\}$ Legacy plan to position ship s on day d in region r [binary]

Suppose we are very fond of the legacy plan for CG61 Monterey. We can track any modification of that plan in model revisions by:

$$\sum_{\substack{d \in D, r \in R, \\ \widehat{presence}_{CG61,d,r}=0}} PRESENCE_{CG61,d,r} + \sum_{\substack{d \in D, r \in R, \\ \widehat{presence}_{CG61,d,r}=1}} (1 - PRESENCE_{CG61,d,r}) .$$

The former term counts, over the entire planning horizon, any new assignment not part of the legacy plan, and the latter any legacy assignment that is abandoned by a revision. We can use expressions involving either, or both terms to count changes we dislike, and penalize these in our objective function. Conventional practice is to count changes of each type (respectively addition and deletion) by planning day, and use a time discount factor to mitigate the influence of these changes into the future, accounting for the “fog of war.” That is, if we do need to accommodate some change, better it be as far

into the future as possible to admit responsive actions to make up for any deficiency.
Here is an example:

Addition data

add_penalty cost per additional assignment to legacy plan [\$]

drop_penalty cost per deleted assignment from legacy plan [\$]

Additional decision variables

$ADDS_{s,d,r}$ counts additional assignment(s) to legacy plan

$DROPS_{s,d,r}$ counts deleted assignment(s) from legacy plan

Additional Persistence constraints:

$$ADDS_{CG61,d,r} = PRESENCE_{CG61,d,r} \mid \overline{presence}_{CG61,d,r=0} ,$$

$$DROPS_{CG61,d,r} = (1 - PRESENCE_{CG61,d,r}) \mid \overline{presence}_{CG61,d,r=1} .$$

Additional Persistence Penalties

Add to the objective function:

$$add_penalty e^{-0.1d} ADDS_{CG61,d,r} + drop_penalty e^{-0.1d} DROPS_{CG61,d,r} .$$

Here, the planner can penalize a revision with any additional assignment, or penalize at a different rate any deleted assignment. For illustration, a penalty discount rate of 10% per day is applied.

Using mechanisms like this, we can shape any revision to our liking.

The attractive charm of this technique is that it will never render a revision of a legacy plan infeasible, as fixing a variable might do.

The caution here is not to set persistence penalties so high as to end up “steering by our wake,” sticking with legacy plans to the detriment of otherwise attractive revisions.

IV. SCENARIOS, ANALYSIS, AND CONCLUSION

All of the following scenarios are solved using CPLEX unless otherwise noted. In order to show comparative results between CBC and CPLEX solver within the network flow model and the “with supply ships” scenario we test the CBC solver against CPLEX for solve time and display a “sawtooth” diagram suggested by Hallmann’s (2009) research.

A. MATCHING SCENARIOS

Previous NMP theses Dugan (2007), Silvia (2009), Hallmann (2009), and Pearlsweig (2013), develop their Area of Operation (AOO) around the Korean Peninsula. Each thesis keeps the same AOO but modifies the scenarios slightly for each new model and comparison. The most complicated of these scenarios is by Hallmann, who includes supply ships and cases where these need to be escorted by combatants. We choose this scenario as our verification specimen, comparing Hallmann’s results with GAMS-CPLEX with our results with Python-Pyomo-CBC.

The next few paragraphs give a brief overview of Hallmann’s scenario and how we have been able to match the results using Python-Pyomo and CBC solver.

1. Missions

Hallmann’s (2009) scenario used 695 missions spanning a time horizon of 15 days and contained 380 prerequisite missions. Types of missions included Air Defense (AD), Theater Ballistic Missile Defense (TBMD), Anti-Submarine Warfare (ASW), Surface Warfare (SUW), Strike, Naval Surface Fire Support (NSFS), Maritime Interdiction Operation (MIO), Mine Hunting Mine Countermeasure (MCM), Intel, and Submarine Intel (SubIntel) as outlined in Joint Publication 1-02 (2010). Each mission is assigned a value based on the desired mission accomplishment and any required simultaneous prerequisite mission(s) (Table 1).

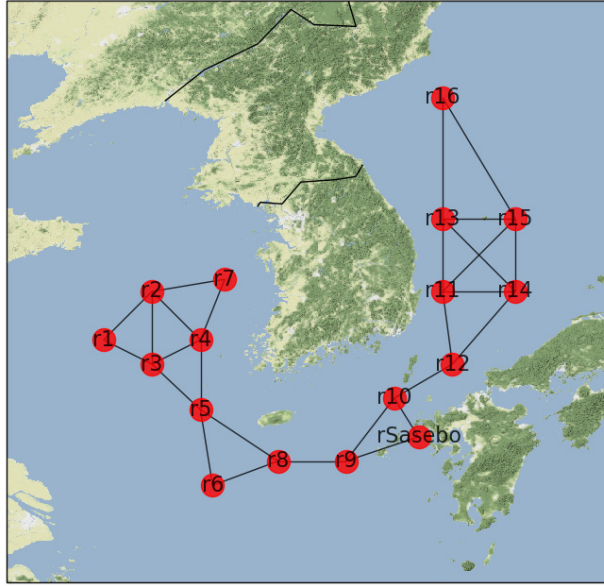
Table 1. Mission Inputs

Mission	Include	Type	Region	Start Day	End Day	Value	RequiresA	RequiresB
m1	x	MIO	r1	1	4	9	AD	
m2	x	AD	r1	1	4	7		
m3	x	ASW	r1	1	4	8	AD	SUW
m4	x	Intel	r1	1	4	7		
m5	x	TBMD	r2	1	15	20	AD	
m6	x	MIO	r3	1	4	5	AD	
m7	x	AD	r3	1	15	3		
m8	x	ASW	r3	1	4	4	AD	
m9	x	Intel	r3	1	15	3		
m10	x	MIO	r4	1	4	7	AD	
m11	x	AD	r4	1	15	5		
m12	x	ASW	r4	1	4	6	SUW	
m13	x	Intel	r4	1	15	5		
m14	x	Strike	r4	5	11	7	AD	

A snapshot of the first 14 missions that, across the planning horizon total 114 missions of the 695 missions from the Microsoft Excel user interface. Displayed are missions in regions 1–4, their time horizon from start day to end day, the value of each mission, and required prerequisite missions. For example, in region “r1” from days 1–4 there is an ASW mission with a value of 8 per day requiring AD and SUW for mission accomplishment.

2. Area of Operation

The scenario uses the maritime AOO surrounding the Korean Peninsula with 17 regions spread out from the Yellow Sea to the Sea of Japan. Each node represented is a region within the AOO rather than a specific location. The node acts as a center point for each region with a radius that encompasses the missions assigned to that region. An arc between a pair of nodes represents adjacency of these nodes by which a ship can travel from node (region) to adjacent node (region) via a great circle navigation (Figure 3). Each region is assigned a longitude and latitude, and a code indicating whether a region is a “hot zone,” a restricted area requiring more than one escort (ratio) for naval assets. Such areas are established when escorts are required for unarmed (e.g., LHA) or supply (e.g., TAO) vessels. Restricted areas are geographical locations where specific ships cannot operate. If a ship requires more than one escort, a ratio of defending escorts to defended ships is available (see Table 2).



Each region is represented as a node (red circles) and arcs (black lines) represent the connection between adjacent nodes. Some adjacencies (e.g., 3-to-4) can be traversed day-to-day, while others (e.g., 13-to-16) require two days with one intermediate day in transit. Figure from Python-Cartopy output (Cartopy, 2015).

Figure 3. Region Map of AOO in Korean Peninsula

Table 2. Region User Inputs

Region	LON	LAT	Hot?	Restrict?	RATIO
r1	123	35 y			1
r2	124	36 y			1
r3	124	34.5 y			1
r4	125	35 y		S	1
r5	125	33.56 y			1
r6	125.25	32			0
r7	125.5	36.25 y		S	1
r8	126.6	32.5			0
r9	128	32.5			0
r10	129	33.8			0
r11	130	36 y			1
r12	130.2	34.5			0
r13	130	37.5 y		S	1
r14	131.5	36			0
r15	131.5	37.5 y			1
r16	130	40 y		S	1
rSasebo	129.5	33		C	0

This table represents each region and its assigned center longitude and latitude. Also shown is whether it is a hot zone, restricted water space, and/or the ratio of escorts required therein. For example, region r4 is located at longitude 125E and latitude 35N, it is a hot zone requiring escorts for non-combatant ships, the “S” signifies supply ships are restricted from entering this water space unaccompanied by an escort, and it requires a ratio of 1 escort per unarmed naval ship. Also, we see that region rSasebo has a “C” under restrictions, indicating that combatant ships are restricted from that water space.

3. Ships

Within the AOO, each ship will arrive on an indicated day in an indicated region, and its availability will end on a forecast day, but perhaps in any region. Each ship is designated by its hull number, class, and the type (Combat, Unarmed, or Supply). Each ship is capable of adopting any one of a variety of combined mission capabilities (CMCs). Each CMC is a set of simultaneous missions the ship can complete (Table 3).

Table 3. Sample Ship Inputs

Ship	Name	Avail	Class	Type	Start Day	Start Region	CMC1	CMC2	CMC3	CMC4	CMC5
CG61	Monterey	x	CG	COMBAT	1	r2	C4	C5	C7	C13	C12
CG66	Hue City	x	CG	COMBAT	1	r13	C5	C6	C8	C13	C12
CG72	Vella Gulf	x	CG	COMBAT	4	r7	C6	C9	C12	C13	
CG58	Philippine Sea	x	CG	COMBAT	7	r10	C7	C5	C10	C13	C12
CG63	Cowpens		CG	COMBAT	1	r16	C4	C13			
CG56	San Jacinto		CG	COMBAT	1	r16	C4	C13			
CG65	Chosin		CG	COMBAT	1	r16	C4	C13			
DDG53	John Paul Jones	x	DDG	COMBAT	1	r1	C14	C18	C20	C23	C22
DDG62	Fitzgerald	x	DDG	COMBAT	1	r4	C14	C18	C20	C23	C22
DDG86	Shoup	x	DDG	COMBAT	1	r9	C15	C18	C20	C23	C22

This is snapshot of how each ship is described. It may be selected from a catalog of Navy and/or coalition ships as available, has a class, type, start day, start region, and a list of one or more combined mission capabilities. For example, the USS Vella Gulf (CG72) is available, class CG, type combat, starts on day 4 in region r7, and has 4 CMCs available for mission selection.

4. Logistics

Hallmann (2009) adds logistical support to the NMP scenario for combat ships (e.g., CG) and unarmed naval combatants (e.g., LHA), by adding CLF ships, commodities used by each combatant in terms of consumption rates (Table 4), loadouts, and capacities (Table 5). Penalties are added for combatants when any commodity remaining drops to a safety stock level, or perhaps an even lower extremis level (Table 6.) Hallmann also introduces alternate ways a CLF can transport fuel to combatants. The first takes Salvia's (2009) scenario and implements these penalties, displaying how adding eight CLF ships mitigates those. He next models a "delivery boy" scheme, developing an optimal employment schedule going to meet with customer ships where and when they are deployed. The "delivery boy" produces a better objective value. Hallmann completes his contributions with a "gas station" plan, manually creating a

single employment schedule for each CLF ship. Typically, each of these loiters in one region for the duration of the planning horizon. The “gas station” approach degrades the objective value but proves sufficient to replenish combatant customers. For our delivery boy scenario CLF and combatant customer employment schedules are created simultaneously and commodities can be transferred when combatants and CLF ships are collocated in the same region on the same day.

Table 4. Sample Consumption Rates based on Combined Mission Capabilities

Consumption	DFM	JP5	STOR	ORDN
C1	0	3000	53	0
C2	0	3000	53	0
C3	0	3000	53	0
C4	1429	5	2	0
C5	1429	5	2	0
C6	1429	5	2	0
C7	1429	5	2	0
C8	1429	5	2	0
C9	1429	5	2	0

This table shows a snapshot of user inputs for how much fuel is consumed while conducting specific combined mission capabilities. For example, combined mission capabilities C1 does not use DFM because C1 is reserved for aircraft carriers but, C4 is reserved for cruisers and they utilize DFM for propulsion.

Table 5. Capacities and Load for Each Class of Ship

Ship Class	Capacity				Initial Load			
	DFM	JP5	STOR	ORDN	DFM	JP5	STOR	ORDN
CVN	0	74,642	1,710	1,765	100%	100%	100%	100%
CG	15,032	475	68	94	100%	100%	100%	100%
DDG	10,518	475	55	48	100%	100%	100%	100%
FFG	4,286	475	35	16	100%	100%	100%	100%
LCS	2,663	579	5	20	100%	100%	100%	100%
SSN	0	0	10	50	100%	100%	100%	100%
SSGN	0	0	30	70	100%	100%	100%	100%
MCM	3,500	0	10	25	100%	100%	100%	100%
TAKE	7,000	17,000	1,963	3,647	100%	100%	100%	100%
TAFS	8,674	10,000	4,600	0	100%	100%	100%	100%
TAOE	62,400	93,600	952	2,016	100%	100%	100%	100%
TAO	72,000	108,520	220	0	100%	100%	100%	100%
TAE	8,647	1,000	38	4,928	100%	100%	100%	100%

Values for DFM and JP5 are measured in barrels, and stores and ordnance are measured in tons. Initial Load is the starting percentage of fuel, stores, and ordnance capacity. Here, each class of ship starts with 100 percent. For example, a frigate (FFG) is capable of carrying and starts with 4,286 barrels of DFM, 475 barrels of JP5, 35 tons of supplies, and 16 tons of ordnance for this scenario.

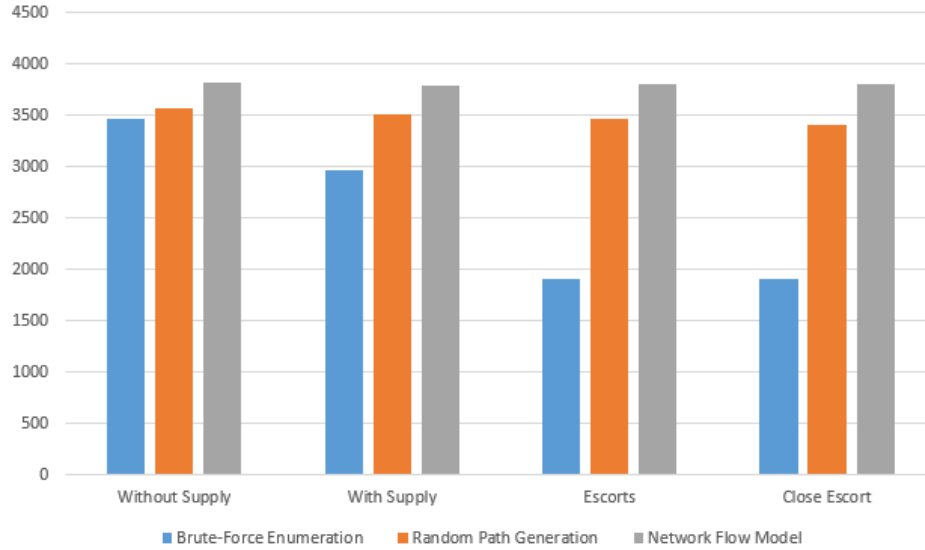
Table 6. Inventory Safety Levels and Penalties

Commodity	safety	extremis	pen safe	pen extr	pen out
DFM	50%	25%	0.0001	0.5	1
JP5	50%	25%	0.0001	0.5	1
STOR	50%	25%	0.001	0.5	1
ORDN	50%	25%	0.001	0.5	1

Safety and lower extremis inventory levels are percentages of capacity that trigger penalties if violated. In the model, it is possible for a customer to run out of a commodity, but this is severely penalized. (We would rather deliver a plan with identified infeasibilities than no plan at all, the better to help revise and improve the scenario.) Dollar penalty-per-unit violation of safety, extremis and stock-out levels are monotonically increasing.

B. REPEATING AND REPRODUCING HALLMANN'S RESULTS

When comparing the two mixed integer linear programs, the results of both GAMS and Python-Pyomo must match. Utilizing Hallmann's (2009) scenario described above we run the NMP model in both GAMS and Python-Pyomo with the paths generated by the truncated brute-force stack-based enumeration. The number of paths generated by VBA is limited to 196,298 paths averaging 6,768 paths per ship. The setting includes 19 combatant ships including three submarines, two unarmed ships, and eight supply ships. There are 695 different missions with values that take place across a 15-day time horizon and 380 prerequisite missions. Below is a bar chart displaying the results of all scenarios (with or without supply, escorts, and close escorts), compared against brute-force enumeration, random path generation, and the network flow model, as shown in Figure 4.



Here we can see the values from NMP without supply, with supply, utilizing escorts, or close escorts and the variation in objective value between brute-force enumeration, random path generation, and the deployment network flow model. We can see how comparatively better the random path and deployment network flow does with escorts and close escorts when compared against Silvia's (2009) stack-based enumeration.

Figure 4. Objective Value Output of Navy Mission Planner

1. Without Supply Ships

By running the scenario without supply ships we are not taking into consideration any of the penalties that might be acquired with running low on commodities (i.e., fuel, stores, or ordnance). Using only the objective function and the first five constraints (T1-T5), this creates a mixed integer program with 222,196 binary variables totaling to 226,563 variables, and 13,881 constraints, when the model is built utilizing Python-Pyomo it solves in approximately one minute. With this we observe an optimal objective value of 3,471, a duplicate objective value when same scenario in generated in GAMS. Both solutions complete 580 of the 695 missions or 380 prerequisite missions assigned across the 15-day planning horizon.

2. With Supply Ships

Utilizing supply ships within the same scenario as described in Hallmann (2009) allows for the eight supply ships to conduct replenishment at-sea with combatant vessels. This feature utilizes the same objective function with constraints T1-T5 and T8-T14,

which allow penalties for any combatant ship that reaches safety, extremist, or runs out of any of the commodities. This produces a mixed integer program with 264,863 binary variables totaling to 287,904 variables and 134,049 constraints within Python-Pyomo, giving us a solve time of one hour. This solution complete 465 of the 695 missions or prerequisite missions across the 15-day planning horizon. Using supply ships to replenish commodities in combatant ships gives us a mission accomplishment value degradation from our best case of 3,471 to 3,044 with total penalties of 81 giving us an objective value of 2,963 with an upper bound of 3,011, with the same numbers observed in both GAMS and Python-Pyomo.

3. With Escorts

Enabling the escort feature requiring any supply ship to have a combatant within a region that is marked as requiring escorts, this utilizes the same objective function with constraints T1-T6 and T8-T14 enabled. This produces a mixed integer program with 264,863 binary variables totaling to 287,904 variables, and 135,549 constraints, with a solve time of approximately 45 minutes. Requiring at least one combatant ship in the same geographical region as each supply ship derogates our mission accomplishment value to 2,063 with 160 penalties, giving an objective value of 1,902 with an upper bound of 1,912, both GAMS and Python-Pyomo are within these bounds.

4. With Close Escorts

The close escort feature requires one combatant ship capable of “Close Escort” within its combined mission capabilities per every one supply ship within the same geographical region at the same time. The difference between close escort and regular escort is the one-on-one interaction during the supply ship transits. With the same objective function and constraints T1-T5 and T7-T14 it produces a mixed integer program with 264,863 binary variables totaling 287,904 variables, and 135,549 constraints. Forcing the one-on-one combat-to-supply ship scenario gives us a total mission accomplishment of 2,063 with 160 in penalties, giving us an objective value of 1,903 with an upper bound of 1,908, solving in approximately 30 minutes. Because of the restriction of requiring a close escort, only four of the eight supply ships are used for this

scenario. An example of this one-to-one accomplishment is displayed with TAO199 and TAO197 on day 15 both in region “r14,” consequently CG58 and DDG97 are both in region “r14” on the same day.

C. RANDOM PATH GENERATOR RESULTS

Here we examine the benefits of randomizing the adjacency list while generating paths, giving the model a more diverse path list in comparison to the truncated brute-force stack-based enumeration. With the goal of generating at most 10,000 random paths per ship, the number of paths generated after removing duplicates, varies from 9,649 (SSN717 with a start date on day 1) to 1,331 (FFG47 with a start date on day 7) and 18 of the 28 ships have more than 8,000 diverse paths. The number of paths per ship varies due to the range of days and the diversity of the random generation: a ship with a planning horizon of 15 days is going to require more diversity than a ship with a planning horizon of 8 days. Each path generated also limits the number of days spent in one specific region to five days.

1. Without Supply Ships

Running the NMP model without supply ships gives us a test of how many missions combatant ships can pick up without being constrained by commodity penalties. Here we use our random paths with our objective function and constraints T1-T5, and this creates 236,556 binary variables, totaling to 248,317 variables, and 20,889 constraints. After a run time of approximately 60 minutes, we get an objective value of 3,561 with an upper bound of 3,624, improving the mission completion of the truncated brute-force enumeration by 90, a 2.5% increase in objective value. This drops the total missions accomplished from 580 to 536 of the 695 missions with values and 380 prerequisite missions, but is able to achieve missions with higher values.

2. With Supply Ships

When using supply ships with our diversified random path generation, supply ships are more available to meet the needs of combatant ships. Utilizing the objective value and constraints T1-T5 and T8-T14, we create a mixed integer program with

274,776 binary variables totaling to 297,817 variables, and 134,049 constraints. After approximately four hours of solve time, our result in mission completion value is 3,516 with penalties resulting in an objective value of 3,514 with an upper bound of 3,647. The total missions completed increases from 465 with the brute-force enumeration to 519 with our random path generation.

3. Escorts

Using supply ships with escorts requires at least one combatant ship in a restricted region for supply ships to be in the same region. This uses the same objective function and constraints as the “with supply ships” scenario with the additional constraints T6. We create a mixed integer program with 274,776 binary variables totaling to 297,817 variables, and 135,549 constraints. This gives us a mission completion value of 3,461 with two in penalties giving an objective value of 3,459 with an upper bound of 3,647 after an approximate 4-hour solve time.

This is an increase of 45% in objective value over the truncated brute-force stack-based enumeration.

4. Close Escorts

This scenario places a one-to-one ratio for each supply ship that must be escorted by one combatant when in a restricted region. This uses the same objective function and constraints as the “with supply ships” scenario with the additional T7 constraint. We create a mixed integer program with 274,776 binary variables totaling to 297,817 variables, and 135,549 constraints. This gives us a mission completion value of 3,414 with two in penalties giving an objective value of 3,412 with an upper bound of 3,647 after an approximate 4-hour solve time.

This is an increase of 44% in the objective value when compared against the truncated brute-force stack-based enumeration.

D. DEPLOYMENT NETWORK FLOW RESULTS

In our network flow model, we add additional constraints F1 in place of T1 creating a linear integer deployment network flow model. This involves additional sets of nodes and arcs derived from our adjacency list based on a ship's starting region and starting day and every region that can be explored in the upcoming days in the planning horizon. This explores all possible paths each ship could make and gives us an optimal solution, using the same scenario as Hallmann (2009).

Because our deployment flow network is lexicographic and acyclic, it is easy to count the number of paths from origin at the start of the planning horizon to the destination at its end. There are 1,058,826,559,993 different paths across the 28 ships within the 15-day planning horizon, compared to the truncated brute-force enumeration of 196,298 paths or 206,211 with our random path generation.

Sampling only about 0.000019% of paths for both truncated brute-force enumeration achieves 90% mission value and random path generation achieves over 93% of mission value.

Adding network flow to the NMP model not only gives us a “best case” scenario but computationally this method is significantly faster than previous attempts with path generation.

1. Without Supply Ships

With this scenario we test our network flow model without the penalties from combatant ships running low on commodities. The network flow model with the objective constraints F1 and T2-T5, is a mixed integer program with 46,157 binary variables totaling to 57,918 variables, and 23,219 constraints. This results in an optimal objective value of 3,819 with a matching upper bound. With this scenario we also test the results

These results give us an increase in objective value of 9.1% when compared with the truncated brute-force stack-based enumeration, with a solve time of seconds.

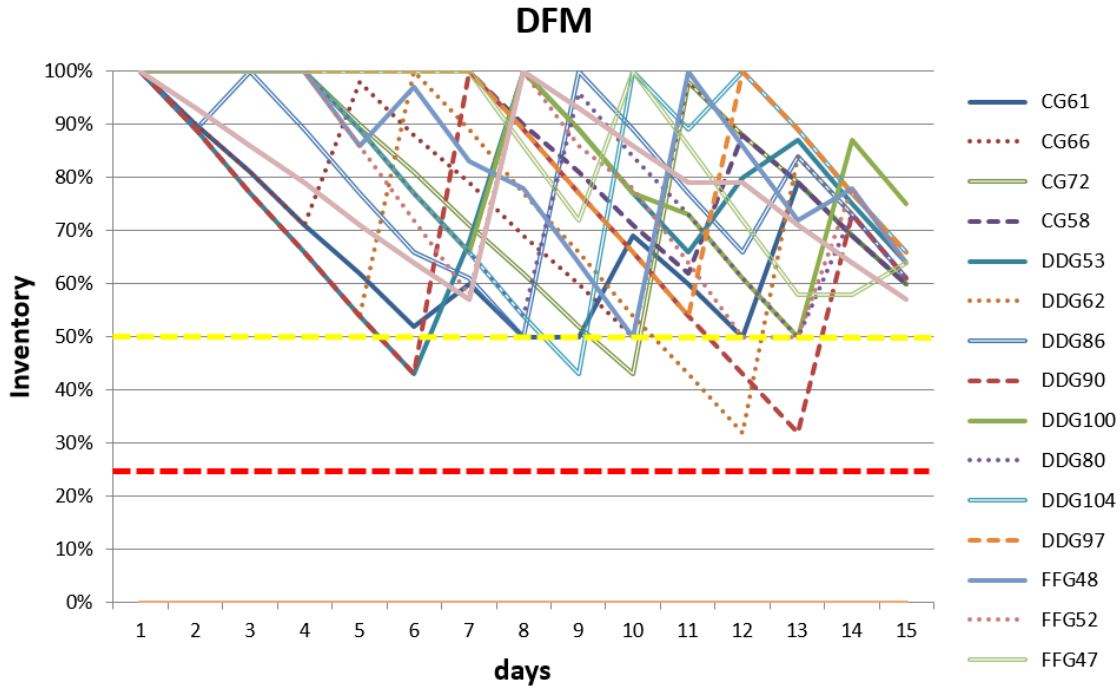
This results in completion of 564 missions of the 695 missions with values and 380 prerequisite missions.

2. With Supply Ships

Here we test our network flow model with supply ships able to conduct underway replenishment, capable of restocking combat ships if both supply and combatant customer ships are within the same region on the same day. With the objective and constraints F1, T2-T5, and T8-T14, we create a mixed integer program with 104,381 binary variables totaling to 127,422 variables, and 136,347 constraints. This results in a mission completion value of 3,798 with less than one in commodity penalties, giving us an objective value of 3,796 in an upper bound of 3,815. Having more freedom to move across regions within our network allows for minimal penalties nearly duplicating the results with or without supply ships,

giving us an increase in objective value of 22.3% when compared to the brute-force enumeration.

With a CPLEX solve time of approximately 45 minutes with a 0.05% solution, and CBC solve time of 60 minutes with a 0.1% solution (double the relative integer tolerance for CPLEX). This shows that CPLEX solver is faster at computing results, however CBC achieves the same results. This completes 578 missions; forcing the ships to pick up more missions of lower value to avoid higher commodity penalties. Fuel level violations are few and small, as we can see in the sawtooth chart in Figure 5 when the ships get refueled.



This figure displays a sawtooth diagram of the DFM fuel consumption for all of our combatant ships. It shows that only six ships go below their safety stock levels and only two ships are in the safety level for two days all, while other ships are only in safety levels for one day.

Figure 5. Sawtooth Chart for Network Flow with Supply Ships

3. Escorts and Close Escorts

For both escorts and close escorts options with the network flow model, supply ships must be accompanied by combatant ships similar to the random path generation model. However, if a supply ship starts in a region with no combatant ships, the path generation models will simply not use that supply ship for the duration of the planning horizon. In this situation, the network flow model will report infeasibility. Therefore, for comparable results the three supply ships not used in the random path generation are switched off for the network flow model. With escorts we get 86,302 binary variables totaling 105,563 variables and 99,097 constraints. For close escorts we get 104,381 binary variables totaling to 127,422 variables and 137,847 constraints. Even with only five of the eight supply ships, for both escort and close escort we get a similar objective value. With escorts we get a mission accomplishment value of 3,803 with less than a

penalty of one resulting in an objective value of 3,802 with an upper bound of 3,818, solving in under 20 minutes. Close escorts give us similar results with total mission accomplishment at 3,802 and one in penalties with an objective value of 3,801 with an upper bound of 3,818, and solves within 20 minutes.

E. PERSISTENCE SCENARIO

Our persistence scenario involves an unpredicted loss of a ship, DDG 62, on the fifth day of the planning horizon with our network flow model. Here we set our model to fix the first five days of the planning horizon, keeping the schedules unchanged, giving us a start region of the location of all ships on day five. NMP then calculates a new near-optimal result based on the loss of DDG62. With the add and drop penalties set to zero any change to the schedule does not affect the objective value. This results in changing all ships schedules for the remaining days on the planning horizon, shown in Table 7. This gives us an objective value of 3,614 with an upper bound of 3,615 and commodity penalties of less than one, a mission accomplishment value of 3,615. Changing the add and drop penalties to a value of one significantly reduces the number of schedule changes with minimal effect on the objective function value. Now, only five ships have a schedule change and only one or two days are changed across their planning horizon, as shown in Table 8. This gives us an optimal objective value of 3,606 with zero commodity penalties and six in persistence penalties yielding a mission accomplishment value of 3,612,

After loss of DDG62 on day five, an optimal revision of her sister ships' schedules changes the vast majority of their schedules to achieve a total mission accomplishment value only 0.08% better than a persistent schedule that makes only seven changes to five sister ships.

This small depreciation in mission accomplishment value would be worth not changing all ships schedules to a decision maker.

Table 7. Persistence with No Penalties

	d1	d2	d3	d4	d5	d6	d7	d8	d9	d10	d11	d12	d13	d14	d15
CG61	---	---	---	---	---	X	X	X	X	X	X	X	X	X	X
CG66	---	---	---	---	---	X	X	---	X	X	X	---	---	---	---
CG72	---	---	---	---	---	---	---	---	---	---	---	---	X	X	X
CG58	---	---	---	---	---	---	---	---	X	X	X	X	X	X	X
DDG53	---	---	---	---	---	---	---	---	X	X	X	---	X	---	---
DDG62	---	---	---	---	---	X	X	X	X	X	X	X	X	X	X
DDG86	---	---	---	---	---	X	---	---	X	X	X	---	---	X	X
DDG90	---	---	---	---	---	X	X	X	X	X	X	X	---	X	X
DDG100	---	---	---	---	---	---	X	X	X	X	X	X	X	X	X
DDG80	---	---	---	---	---	---	X	---	X	X	X	---	X	X	---
DDG104	---	---	---	---	---	X	---	---	---	---	X	---	---	X	X
DDG97	---	---	---	---	---	---	---	---	X	X	---	---	X	X	X
FFG48	---	---	---	---	---	---	---	---	---	---	---	X	---	---	---
FFG52	---	---	---	---	---	---	X	---	X	---	X	---	---	X	X
FFG47	---	---	---	---	---	---	---	X	X	X	X	---	---	---	---
SSN752	---	---	---	---	---	---	---	---	X	---	---	---	---	---	---
SSN718	---	---	---	---	---	---	X	X	---	---	---	---	---	---	---
SSN717	---	---	---	---	---	X	X	X	X	X	X	X	X	X	---
MCM6	---	---	---	---	---	---	---	X	---	---	---	---	---	---	---
MCM8	---	---	---	---	---	---	X	---	---	---	---	---	---	---	---
TAKE1	---	---	---	---	---	X	X	X	X	X	X	X	X	X	X
TAFS7	---	---	---	---	---	---	---	X	X	X	X	X	X	X	X
TAFS5	---	---	---	---	---	X	X	X	X	X	X	X	X	X	X
TAFS8	---	---	---	---	---	X	X	X	X	X	X	X	X	X	X
TAOE6	---	---	---	---	---	X	X	---	X	X	X	X	X	X	X
TAO187	---	---	---	---	---	---	X	X	X	X	X	X	X	X	X
TAO199	---	---	---	---	---	---	---	X	X	---	X	X	X	X	X
TAO197	---	---	---	---	---	X	X	X	---	---	---	X	X	---	X

From this table we can see all changes made to ships' schedules with no persistence penalties when DDG62 is removed from the planning horizon after day five. The "—" mark days where the ships geographical region is maintained within the schedule and "X" marks days where their schedule is changed. This turbulent response to a simple revision is ridiculous.

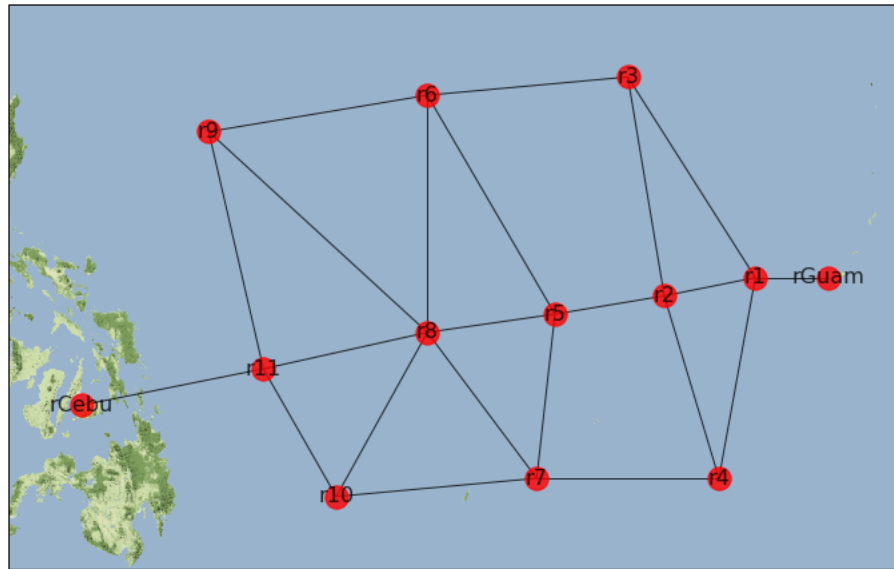
Table 8. Persistence with Penalties

	d1	d2	d3	d4	d5	d6	d7	d8	d9	d10	d11	d12	d13	d14	d15
CG61	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---
CG66	---	---	---	---	---	---	X	---	---	---	---	---	---	---	---
CG72	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---
CG58	---	---	---	---	---	---	---	---	---	X	X	---	---	---	---
DDG53	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---
DDG62	---	---	---	---	---	X	X	X	X	X	X	X	X	X	X
DDG86	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---
DDG90	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---
DDG100	---	---	---	---	---	---	X	---	---	---	---	---	---	---	---
DDG80	---	---	---	---	---	---	---	---	---	X	---	---	---	---	---
DDG104	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---
DDG97	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---
FFG48	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---
FFG52	---	---	---	---	---	---	X	---	---	---	X	---	---	---	---
FFG47	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---
SSN752	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---
SSN718	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---
SSN717	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---
MCM6	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---
MCM8	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---
TAKE1	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---
TAFS7	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---
TAFS5	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---
TAFS8	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---
TAOE6	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---
TAO187	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---
TAO199	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---
TAO197	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

Here we see how adding persistence, with the loss of DDG62 can affect rescheduling ships and reduce the alteration in schedules. The “---” mark days where the ships geographical region is maintained within the schedule and “X” marks days where a schedule is changed. We can see the adjustment of deployment and mission assessment for CG66, CG58, DDG100, DDG80, and FFG52, *merely seven revisions* of the legacy plan for these five ships. This retains 99.92% of the optimal achievable revisions in Table 7.

F. PHILIPPINE SEA SCENARIO

For an additional scenario testing NMP, we look at transporting logistics across contested waters in the Philippine Sea. This deploys a convoy of CLF ships carrying fuel, supplies, and ordnance from Guam to Cebu, Philippines. The CLF fleet must be escorted at all times by a minimum of four defensive combatants while making the transit. The largest threat in this scenario is a submarine attack disrupting this supply chain by sinking supply ships. We run this scenario utilizing the network flow mode and the armed escort constraint T15. We create each region from Guam to Cebu, Philippines, with no more than a one-day transit to its neighbors, as shown in Figure 6.



Each region is represented as a node (red circles) and arcs (black lines) represent the connection between adjacent nodes. The goal is getting a detachment of CLF ships from Guam to Cebu, Philippines, to deliver supplies. This is a contested transit, and defensive combat ships must defend this convoy, mainly against submarine threats.

Figure 6. Region Map of the AOO in Philippine Sea

1. Ships

For this scenario we utilize three CG's, eight DDG's, two LCS's, two SSN's, two MCM's, and one supply ship to defend the CLF convoy. All ships' CMC's are the same as Hallmann's (2009) scenario. Table 9, shows the inputs for NMP.

Table 9. Ships for Philippine Sea Scenario

Ship	Name	Avail	Class	Type	Start Day	Start Region	CMC1	CMC2	CMC3	CMC4	CMC5
CG61	Monterey	x	CG	COMBAT	2	rGuam	C4	C5	C7	C13	C12
CG72	Vella Gulf	x	CG	COMBAT	2	rGuam	C6	C9	C12	C13	
CG58	Philippine	x	CG	COMBAT	3	rGuam	C7	C5	C10	C13	C12
DDG53	John Paul	x	DDG	COMBAT	2	rGuam	C14	C18	C20	C23	C22
DDG62	Fitzgerald	x	DDG	COMBAT	2	rGuam	C14	C18	C20	C23	C22
DDG86	Shoup	x	DDG	COMBAT	2	rGuam	C15	C18	C20	C23	C22
DDG90	Chaffee	x	DDG	COMBAT	2	rGuam	C15	C18	C20	C23	C22
DDG100	Kidd	x	DDG	COMBAT	3	rGuam	C16	C19	C22	C23	
DDG80	Roosevelt	x	DDG	COMBAT	3	rGuam	C14	C18	C20	C23	C22
DDG104	Sterett	x	DDG	COMBAT	3	rGuam	C14	C18	C20	C23	C22
DDG97	Halsey	x	DDG	COMBAT	3	rGuam	C14	C18	C20	C23	C22
LCS1	Freedom	x	LCS	COMBAT	1	rGuam	C30	C33			
LCS2	Independence	x	LCS	COMBAT	1	rGuam	C30	C33			
SSN752	Pasadena	x	SSN	COMBAT	1	r3	C35	C40			
SSN718	Honolulu	x	SSN	COMBAT	1	r4	C37	C40			
MCM6	Devastator	x	MCM	UNARMED	1	rGuam	C44	C45			
MCM8	Scout	x	MCM	UNARMED	1	rGuam	C44	C45			
TAOE6	Supply	x	TAOE	SUPPLY	3	rGuam	C48				

For the transit from Guam to Cebu, Philippines we utilize three CG's, eight DDG's, two LCS's, two SSN's, two MCM's, and one aggregated (i.e., synthesized) supply ship (TAOE 6) representing four physical supply ships in the convoy.

2. Regions

The regions in this scenario represent a chain of passages on a one-way trip from Guam to Cebu, Philippines, with each region being spaced no more than a one-day transit to the following region. Regions rGuam, r1, r2, r5, r8, r11, and rCebu represents the direct path for the CLF convoy to take. These regions are given a force ratio value (see constraint T15) of four, requiring four defensive combatants to escort the CLF convoy. We force the CLF convoy down this path by making the force ratio a value of eight for the regions we do not want the CLF ships to transit, which would require eight ships to be escorted (an unobtainable number), as shown in Table 10.

Table 10. Regions for Philippine Sea Scenario

Region	LON	LAT	Hot?	Restrict?	RATIO
rGuam	144.5	13.5			4
r1	142.5	13.5			4
r2	140	13			4
r3	139	18.5			8
r4	141.5	8			8
r5	137	12.5			4
r6	133.5	18			8
r7	136.5	8			8
r8	133.5	12			4
r9	127.5	17			8
r10	131	7.5			8
r11	129	11			4
rCebu	124	10			4

Here we can see there are no restricted regions for the CLF convoy, however every region requires at least four defensive combatants for the CLF convoy to transit. We set the ratio for regions outside our preferred route to eight to prevent the CLF convoy from transiting these undesirable regions.

3. Commodities

For this scenario we change commodity capacities of a single CLF supply ship to represent the to represent a convoy of supply ships. In Table 11, we can see that the commodity for TAOE6 have increased by approximately four times her actual capacity.

Table 11. Commodities Load for Philippine Sea Scenario

Ship Class	Capacity			
	DFM	JP5	STOR	ORDN
CVN	0	74,642	1,710	1,765
CG	15,032	475	68	94
DDG	10,518	475	55	48
FFG	4,286	475	35	16
LCS	2,663	579	5	20
SSN	0	0	10	50
SSGN	0	0	30	70
MCM	3,500	0	10	25
TAKE	7,000	17,000	1,963	3,647
TAFS	8,674	10,000	4,600	0
TAOE	249,600	374,400	3,808	8,064
TAO	72,000	108,520	220	0
TAE	8,647	1,000	38	4,928

Highlighted is the amount increased for one TAOE to represent a convoy of four CLF ships. This allows for convoy ships to conduct replenishment at-sea while still maintaining an appropriate amount of commodities for delivery to Cebu.

4. Results

The results of our Philippine Sea scenario show a transit from Guam to Cebu, Philippines, across a nine-day planning horizon. Our CLF fleet must be accompanied by a minimum of four defensive combatants while making the transit. This gives us a mixed integer program with 15,238 binary variables totaling to 22,691 variables and 8,503 constraints. This results in an optimal objective value of 862 accomplishing 188 of the 204 possible missions. As shown in Table 12, we can see the assignment of ships to regions for the purpose of escorting the CLF convoy from Guam to Cebu.

Table 12. Philippine Sea Transit

	d1	d2	d3	d4	d5	d6	d7	d8	d9
CG61	---	rGuam	r1	r4	r7	r10	r7	r10	r7
CG72	---	rGuam	r1	r2	r5	r8	r11	rCebu	r11
CG58	---	---	rGuam	rGuam	r1	rGuam	r1	r2	r5
DDG53	---	rGuam	r1	r3	r6	r9	r9	r11	r11
DDG62	---	rGuam	r1	r4	r4	r7	r10	r11	rCebu
DDG86	---	rGuam	r1	r2	r5	r5	r8	r11	r11
DDG90	---	rGuam	r1	r2	r2	r5	r8	r11	rCebu
DDG100	---	---	rGuam	r1	r3	r6	r8	r6	r5
DDG80	---	---	rGuam	r1	r2	r5	r5	r8	r10
DDG104	---	---	rGuam	r1	r2	r1	r3	r1	rGuam
DDG97	---	---	rGuam	r1	r2	r5	r8	r9	r9
LCS1	rGuam	r1	r3	r2	r5	r8	r11	rCebu	rCebu
LCS2	rGuam	r1	r4	r7	r5	r8	r11	rCebu	r11
SSN752	r3	r6	r8	r7	r10	r11	r11	rCebu	rCebu
SSN718	r4	r7	r10	r8	r5	r8	r8	r11	r8
MCM6	rGuam	r1	r2	r5	r8	r11	r11	rCebu	rCebu
MCM8	rGuam	r1	r2	r5	r8	r11	r11	rCebu	rCebu
TAOE6	---	---	rGuam	r1	r2	r5	r8	r11	rCebu

Here we can see the CLF convoy (TAOE6) transiting across regions rGuam, r1, r2, r5, r8, r11, and rCebu starting on day d3 and ending on day d9 (highlighted in the last row in blue). During that transit the CLF ships are accompanied by at least four combatant ships (highlighted in yellow).

Testing our persistence feature in this scenario, we add additional missions for a group of combatants to conduct anti-surface warfare missions in region r6 from day d6 to d9. This simulates an unanticipated surface threat and how the NMP would reassign ships to address the threat. For this we add three SUW mission to the planner starting on day d6 and ending on day d9 and give each a value of 20, a value equivalent to that of our CLF missions (see Appendix D). We set the persistence start day to day four to simulate only

knowing of this threat one day prior and giving the combatant ships time to intercept the threat in region r6. As shown in Table 13, the CLF fleet maintains its transit to Cebu with four combatant escorts and the SUW missions are accomplished in region r6. This gives us a new objective value of 1,080 with an upper bound of 1,081 an increase due to the new SUW mission in region r6.

Table 13. Persistence in Philippine Sea Scenario

	d1	d2	d3	d4	d5	d6	d7	d8	d9
CG61	---	rGuam	r1	r4	r7	r10	r10	r10	r10
CG72	---	rGuam	r1	r2	r5	r8	r11	rCebu	r11
CG58	---	---	rGuam	rGuam	r1	rGuam	r1	r2	r5
DDG53	---	rGuam	r1	r3	r6	r9	r9	r11	r11
DDG62	---	rGuam	r1	r4	r2	r5	r6	r6	r6
DDG86	---	rGuam	r1	r2	r5	r6	r8	r11	r11
DDG90	---	rGuam	r1	r2	r3	r6	r8	r11	rCebu
DDG100	---	---	rGuam	r1	r2	r5	r8	r6	r5
DDG80	---	---	rGuam	r1	r2	r5	r6	r6	r6
DDG104	---	---	rGuam	r1	r3	r6	r6	r6	r6
DDG97	---	---	rGuam	r1	r2	r5	r8	r9	r9
LCS1	rGuam	r1	r3	r2	r5	r8	r11	rCebu	rCebu
LCS2	rGuam	r1	r4	r7	r5	r8	r11	rCebu	rCebu
SSN752	r3	r6	r8	r7	r10	r11	r11	rCebu	rCebu
SSN718	r4	r7	r10	r8	r5	r8	r8	r11	r8
MCM6	rGuam	r1	r2	r5	r8	r11	r11	rCebu	rCebu
MCM8	rGuam	r1	r2	r5	r8	r11	r11	rCebu	rCebu
TAOE6	---	---	rGuam	r1	r2	r5	r8	r11	rCebu

Here we can see our CLF convoy maintains the same route throughout the planning horizon (highlighted at the bottom in blue), and maintains a four defensive combatant escort throughout the planning horizon (highlighted in yellow). Adding a one-day intercept time for our combatant ships to intercept the threat in region r6 allowed for all three of the SUW missions to get accomplished for the four remaining days in the planning horizon (bolded and highlighted in green).

G. CONCLUSION AND FUTURE RESEARCH

Maritime planning, whether at the tactical or operational level, is a time-consuming manual process for commanders and their staff. By developing a tool that can be accessed on government computers, without commercial licensing, decision makers can quickly make scheduling decisions with minimal staff. We accomplish this by converting NMP from a 1,700-line GAMS mixed integer program script solved with CPLEX into a 1,200-line Python-Pyomo program solved with CBC solver.

Examining a randomization approach to the brute-force enumeration improved the optimality of the NMP solution, however it is more computationally intensive (taking over 3-to-4 hours to solve the integer linear program) than the previous truncated depth-first enumeration.

Adding a deployment network flow model to NMP allows for the best improvements in objective value and makes NMP solvable in a matter of minutes. Persistence in optimization allows for us to make changes to an already release schedule with minimal degradation to solution value and with fewer revisions to the legacy plan.

The next steps for NMP would be to distribute it to the fleet, putting it into the hands of decision makers. This would allow it to be tested in real-time scenarios resulting in real-world outcomes.

Random path generation, deployment network flow models, and persistence can be applied to a number of other optimization models. Current changes are being made to RASP 1.0 (Brown et. al., 2017) and will incorporate changes made to the NMP model when replaced by RASP 2.0 and be utilized by every numbered fleet by Military Sealift Command (MSC) worldwide. This work can also be applied to the Air Tasking and Efficiency Model (ATEM) (Brown et. al., 2013) used by Scott AFB Air Mobility Command for the transportation of logistical supplies and personnel in war efforts. Also, the Combat Logistics Force (CLF) Model (Brown G. and Carlyle W.M., 2008), utilized by N81, N42, and OSD CAPE, could use this work to improve Underway Replenishment (UNREP) and Vertical Replenishment (VERTREP) scheduling capabilities. Ayik (1998) compares a deployment network flow model with a deployment path-generation restriction like the one presented here. His deployment model is complicated by side constraints on maintenance availabilities. In contrast with our results, his computational experiments favored path generation over the deployment network flow model, for which initial continuous solutions exhibited large objective value integrality gaps with integer incumbents produced by subsequent branch-and-bound enumeration. We speculate that the ensuing 21 years of improvement of linear integer solvers has changed the relative difficulty of these competing optimization models, especially due to significant advances in valid integer and flow cover cuts. Running contemporary solvers, we see the automatic

and energetic application of such cuts, and suppressing these has deleterious effects on solution quality and model solution time. With the change in solver advancements over the past decades, future work implementing a network flow constraint or randomization to the path generation, and a persistence feature could be useful in improving those models.

THIS PAGE INTENTIONALLY LEFT BLANK

APPENDIX A: MISSION INPUTS

Table 14 is a list of mission inputs used to analysis random path, network flow, and persistence against Hallmann’s (2009) scenario. Each mission is assigned a value between 1 and 20 based on the mission priority assigned by the Commander, the higher the number the higher the priority. We can see with mission “m3” we have two required prerequisite missions of Air Defense (AD) and Surface Warfare (SUW) as labeled in the RequiresA and RequiresB columns.

Table 14. NMP Mission Inputs (1 of 3)

Mission	Include	Type	Region	Start Day	End Day	Value	RequiresA	RequiresB
m1	x	MIO	r1	1	4	9	AD	
m2	x	AD	r1	1	4	7		
m3	x	ASW	r1	1	4	8	AD	SUW
m4	x	Intel	r1	1	4	7		
m5	x	TBMD	r2	1	15	20	AD	
m6	x	MIO	r3	1	4	5	AD	
m7	x	AD	r3	1	15	3		
m8	x	ASW	r3	1	4	4	AD	
m9	x	Intel	r3	1	15	3		
m10	x	MIO	r4	1	4	7	AD	
m11	x	AD	r4	1	15	5		
m12	x	ASW	r4	1	4	6	SUW	
m13	x	Intel	r4	1	15	5		
m14	x	Strike	r4	5	11	7	AD	
m15	x	NSFS	r4	5	8	5	AD	
m16	x	SUW	r4	5	11	5	AD	
m17	x	MIO	r4	12	15	3	AD	
m18	x	ASW	r4	12	15	3	SUW	
m19	x	MIO	r5	1	4	5	AD	
m20	x	AD	r5	1	15	3		
m21	x	ASW	r5	1	4	4	SUW	
m22	x	Intel	r5	1	15	3		
m23	x	Strike	r5	5	11	15	AD	
m24	x	NSFS	r5	5	8	7	AD	
m25	x	SUW	r5	5	11	7	AD	
m26	x	MIO	r5	12	15	5	AD	
m27	x	ASW	r5	12	15	5	SUW	
m28	x	MIO	r7	1	4	9	AD	
m29	x	AD	r7	1	15	7		
m30	x	ASW	r7	1	15	8	AD	SUW

Table 15. NMP Mission Inputs (2 of 3)

Mission	Include	Type	Region	Start Day	End Day	Value	RequiresA	RequiresB
m31	x	Intel	r7	1	15	7		
m32	x	Strike	r7	5	11	15	AD	
m33	x	NSFS	r7	5	8	7	AD	
m34	x	SUW	r7	5	11	7	AD	
m35	x	MIO	r7	12	15	5	AD	
m36	x	ASW	r7	12	15	5	SUW	
m37	x	MIO	r8	1	4	5	AD	
m38	x	AD	r8	1	15	3		
m39	x	ASW	r8	1	4	4	SUW	
m40	x	Intel	r8	1	15	3		
m41	x	Strike	r8	5	11	5	AD	
m42	x	NSFS	r8	5	8	3	AD	
m43	x	SUW	r8	5	11	3	AD	
m44	x	MIO	r9	1	4	7	AD	
m45	x	AD	r9	1	15	5		
m46	x	ASW	r9	1	4	6	SUW	
m47	x	Intel	r9	1	15	5		
m48	x	Strike	r9	5	11	5	AD	
m49	x	NSFS	r9	5	8	3	AD	
m50	x	SUW	r9	5	11	3	AD	
m51	x	MIO	r10	1	4	5	AD	
m52	x	AD	r10	1	15	3		
m53	x	ASW	r10	1	4	4	SUW	
m54	x	Intel	r10	1	15	3		
m55	x	Strike	r10	5	11	7	AD	
m56	x	NSFS	r10	5	8	5	AD	
m57	x	SUW	r10	5	11	5	AD	
m58	x	MIO	r10	12	15	3	AD	
m59	x	ASW	r10	12	15	3	SUW	
m60	x	MIO	r11	1	4	7	AD	
m61	x	AD	r11	1	15	5		
m62	x	ASW	r11	1	4	6	SUW	
m63	x	Intel	r11	1	15	5		
m64	x	Strike	r11	5	11	7	AD	
m65	x	NSFS	r11	5	8	5	AD	
m66	x	SUW	r11	5	11	5	AD	
m67	x	MIO	r11	12	15	3	AD	
m68	x	ASW	r11	12	15	3	SUW	
m69	x	ASW	r12	1	15	20		
m70	x	MIO	r13	1	4	9	AD	

Table 16. NMP Mission Inputs (3 of 3)

Mission	Include	Type	Region	Start Day	End Day	Value	RequiresA	RequiresB
m71	x	AD	r13	1	15	7		
m72	x	ASW	r13	1	15	8	AD	SUW
m73	x	Intel	r13	1	15	7		
m74	x	Strike	r13	5	11	15	AD	
m75	x	NSFS	r13	5	8	7	AD	
m76	x	SUW	r13	5	11	7	AD	
m77	x	MIO	r13	12	15	5	AD	
m78	x	ASW	r13	12	15	5	SUW	
m79	x	AD	r2	1	15	20		
m80	x	SubIntel	r16	2	15	20		
m81	x	AD	r15	1	4	15		
m82	x	SUW	r15	1	15	15	AD	
m83	x	ASW	r15	1	15	15	AD	
m84	x	TBMD	r14	1	15	15	AD	
m85	x	AD	r14	1	15	15		

THIS PAGE INTENTIONALLY LEFT BLANK

APPENDIX B: SHIP CATALOG FOR NMP

Each ship marked as available is inputted into the NMP planner along with her type, start day, start region, and CMC's within her capabilities.

Table 17. NMP Ship Catalog

Ship	Name	Avail	Class	Type	Start Day	Start Region	CMC1	CMC2	CMC3	CMC4	CMC5
CG61	Monterey	x	CG	COMBAT	1	r2	C4	C5	C7	C13	C12
CG66	Hue City	x	CG	COMBAT	1	r13	C5	C6	C8	C13	C12
CG72	Vella Gulf	x	CG	COMBAT	4	r7	C6	C9	C12	C13	
CG58	Philippine Sea	x	CG	COMBAT	7	r10	C7	C5	C10	C13	C12
CG63	Cowpens		CG	COMBAT	1	r16	C4	C13			
CG56	San Jacinto		CG	COMBAT	1	r16	C4	C13			
CG65	Chosin		CG	COMBAT	1	r16	C4	C13			
DDG53	John Paul Jones	x	DDG	COMBAT	1	r1	C14	C18	C20	C23	C22
DDG62	Fitzgerald	x	DDG	COMBAT	1	r4	C14	C18	C20	C23	C22
DDG86	Shoup	x	DDG	COMBAT	1	r9	C15	C18	C20	C23	C22
DDG90	Chaffee	x	DDG	COMBAT	1	r7	C15	C18	C20	C23	C22
DDG100	Kidd	x	DDG	COMBAT	4	r5	C16	C19	C22	C23	
DDG80	Roosevelt	x	DDG	COMBAT	4	r13	C14	C18	C20	C23	C22
DDG104	Sterett	x	DDG	COMBAT	4	r4	C14	C18	C20	C23	C22
DDG97	Halsey	x	DDG	COMBAT	7	r11	C14	C18	C20	C23	C22
DDG78	Porter		DDG	COMBAT	1	r16	C14	C23			
DDG74	McFaul		DDG	COMBAT	1	r16	C14	C23			
DDG72	Mahan		DDG	COMBAT	1	r15	C14	C23			
DDG75	Donald Cook		DDG	COMBAT	1	r16	C14	C23			
DDG71	Ross		DDG	COMBAT	1	r16	C14	C23			
DDG54	Curtis Wilbur		DDG	COMBAT	1	r16	C14	C23			
DDG67	Cole		DDG	COMBAT	1	r16	C14	C23			
FFG48	Vandegrift	x	FFG	COMBAT	4	r10	C24	C28	C29		
FFG52	Carr	x	FFG	COMBAT	4	r11	C25	C28	C29		
FFG47	Nicholas	x	FFG	COMBAT	7	r8	C26	C29			
FFG60	Rodney M Davis		FFG	COMBAT	1	r16	C24	C29			
LCS1	Freedom		LCS	COMBAT	1	r16	C30	C33			
LCS2	Independence		LCS	COMBAT	1	r16	C30	C33			
SSN752	Pasadena	x	SSN	COMBAT	1	r12	C35	C40			
SSN718	Honolulu	x	SSN	COMBAT	6	r7	C37	C40			
SSN717	Olympia	x	SSN	COMBAT	1	r16	C40				
SSN770	Tucson		SSN	COMBAT	1	r16	C34				
SSN706	Albuquerque		SSN	COMBAT	1	r16	C34				
SSN764	Boise		SSN	COMBAT	1	r16	C34				
SSGN726	Ohio		SSGN	COMBAT	1	r16	C41				
MCM6	Devastator	x	MCM	UNARMED	1	r1	C44	C45			
MCM8	Scout	x	MCM	UNARMED	1	r1	C44	C45			
MCM10	Warrior		MCM	UNARMED	1	r1	C44	C45			
MCM14	Chief		MCM	UNARMED	1	r1	C44	C45			
TAKE1	Lewis and Clark	x	TAKE	SUPPLY	1	r1	C47				
TAFS7	San Jose	x	TAFS	SUPPLY	1	r2	C48				
TAFS5	Concord	x	TAFS	SUPPLY	1	r6	C48				
TAFS8	Sirius	x	TAFS	SUPPLY	1	r8	C48				
TAOE6	Supply	x	TAOE	SUPPLY	1	r6	C45				
TAO187	Henry J. Kaiser	x	TAO	SUPPLY	1	r14	C46				
TAO199	Tippecanoe	x	TAO	SUPPLY	1	r9	C46				
TAO197	Pecos	x	TAO	SUPPLY	1	rSasebo	C46				

THIS PAGE INTENTIONALLY LEFT BLANK

APPENDIX C: COMMODITY USAGE BASED ON CMC

Each class of ship utilizes commodities differently based on which missions are being accomplished.

Table 18. NMP Commodity Usage Catalog Based on Combined Mission Capabilities

Consumption	DFM	JP5	STOR	ORDN	Consumption	DFM	JP5	STOR	ORDN
C1	0	3000	53	0	C26	600	5	1	0
C2	0	3000	53	0	C27	600	5	1	0
C3	0	3000	53	0	C28	600	5	1	0
C4	1429	5	2	0	C29	600	5	1	0
C5	1429	5	2	0	C30	360	0	0.25	0
C6	1429	5	2	0	C31	360	0	0.25	0
C7	1429	5	2	0	C32	360	0	0.25	0
C8	1429	5	2	0	C33	360	0	0.25	0
C9	1429	5	2	0	C34	0	0	0	0
C10	1429	5	2	0	C35	0	0	0	0
C11	1429	5	2	0	C36	0	0	0	0
C12	1429	5	2	0	C37	0	0	0	0
C13	1429	5	2	0	C38	0	0	0	0
C14	1200	5	2	0	C39	0	0	0	0
C15	1200	5	2	0	C40	0	0	0	0
C16	1200	5	2	0	C41	0	0	0	0
C17	1200	5	2	0	C42	0	0	0	0
C18	1200	5	2	0	C43	0	0	0	0
C19	1200	5	2	0	C44	250	0	0.5	0
C20	1200	5	2	0	C45	250	0	0.5	0
C21	1200	5	2	0	C46	2570	10	1	0
C22	1200	5	2	0	C47	960	10	1	0
C23	1200	5	2	0	C48	960	10	1	0
C24	600	5	1	0	C49	960	10	1	0
C25	600	5	1	0					

THIS PAGE INTENTIONALLY LEFT BLANK

APPENDIX D: PHILIPPINE SEA MISSIONS

Missions used in the Philippine Sea scenario. Missions m43, m44, and m45 are not introduced into the scenario until day four for our persistence scenario.

Table 19. Philippine Sea Missions

Mission	Include	Type	Region	Start Day	End Day	Value	RequiresA	RequiresB
m1	x	UNREP	rCebu	9	9	20		
m2	x	UNREP	r11	8	8	20		
m3	x	UNREP	r8	7	7	20		
m4	x	UNREP	r5	6	6	20		
m5	x	UNREP	r2	5	5	20		
m6	x	UNREP	r1	4	4	20		
m7	x	ASW	r1	1	4	12	AD	
m8	x	ASW	r2	2	5	12	AD	
m9	x	ASW	r5	3	6	12	AD	
m10	x	ASW	r8	4	7	12	AD	
m11	x	ASW	r11	5	8	12	AD	
m12	x	ASW	rCebu	6	9	12	AD	
m13	x	ASW	r4	1	2	7	SUW	
m14	x	ASW	r7	3	5	7	SUW	
m15	x	ASW	r10	6	9	7	SUW	
m16	x	ASW	r3	1	2	7	SUW	
m17	x	ASW	r6	3	5	7	SUW	
m18	x	ASW	r9	6	9	7	SUW	
m19	x	ASW	r1	1	4	12	AD	
m20	x	ASW	r2	2	5	12	AD	
m21	x	ASW	r5	3	6	12	AD	
m22	x	ASW	r8	4	7	12	AD	
m23	x	ASW	r11	5	8	12	AD	
m24	x	ASW	rCebu	8	10	12	AD	
m25	x	MCM	r1	1	2	15		
m26	x	MCM	r1	1	2	15		
m27	x	MCM	r2	3	3	15		
m28	x	MCM	r2	3	3	15		
m29	x	MCM	r5	4	4	15		
m30	x	MCM	r5	4	4	15		
m31	x	MCM	r8	5	5	15		
m32	x	MCM	r8	5	5	15		
m33	x	MCM	r11	6	7	15		
m34	x	MCM	r11	6	7	15		
m35	x	MCM	rCebu	8	10	15		
m36	x	MCM	rCebu	8	10	15		
m37	x	ASW	r1	1	4	12	AD	
m38	x	ASW	r2	2	5	12	AD	
m39	x	ASW	r5	3	6	12	AD	
m40	x	ASW	r8	4	7	12	AD	
m41	x	ASW	r11	5	8	12	AD	
m42	x	ASW	rCebu	6	9	12	AD	
m43	x	SUW	r6	6	9	20	AD	
m44	x	SUW	r6	6	9	20	AD	
m45	x	SUW	r6	6	9	20	AD	

Missions shown are for the Philippine Sea scenario. The last three missions (m43, m44, and m45; highlighted in yellow) are added to the schedule on day four for our persistence scenario.

THIS PAGE INTENTIONALLY LEFT BLANK

LIST OF REFERENCES

- Ahuja, R.K., Magnanti, T.L., & Orlin, J.B. (1993). *Network flows*. Upper Saddle River, NJ: Prentice Hall.
- Ayik, M. (1998). *Optimal long-term aircraft carrier deployment planning with synchronous depot level maintenance scheduling*. MS thesis, Operations Research, Naval Postgraduate School, Monterey, CA.
- Brown, G.G., Dell, R.F., & Wood, R.K. (1997). Optimization and Persistence, *Interfaces*, 27, 15–37.
- Brown, G.G., DeGrange W.C., Price W.L. & Rowe A.A. (2017). Scheduling combat logistics force replenishments at sea for the U.S. Navy, *Naval Research Logistics*, 64(8), 677–693.
- Brown, G.G. & Carlyle, W.M. (2008). Optimizing the U.S. Navy’s combat logistics force, *Naval Research Logistics*, 55(8), 800–810.
- Brown, G.G., Carlyle, W.M., Dell, R.F. & Brau, J.W. (2013). Optimizing intratheater military airlift in Iraq and Afghanistan, *Military Operations Research*, 18(3), pp. 35–52.
- Cartopy. (2015). Cartopy: A cartographic python library with a matplotlib interface. Retrieved from <http://scitools.org.uk/cartopy>
- COIN-OR (2019). Welcome to the Cbc home page, *Cbc (Coin-or branch and cut)*. Retrieved from <https://projects.coin-or.org/Cbc>
- CPLEX. (2019). Introduction to Presolve, *CPLEX*. https://www.ibm.com/support/knowledgecenter/en/SSSA5P_12.6.1/ilog.odms.cplex.help/CPLEX/UsrMan/topics/progr_adv/presolve_adv/02_presolve_intro.html
- Deleon, J. (2015). *Navy Operational Planner*. MS Thesis in Operations Research, Naval Postgraduate School, Monterey, CA, March.
- Department of the Navy, Office of the Chief of Naval Operations. (2013a). *Navy Planning*, NWP 5-01. Washington, DC: Government Printing Office.
- Department of the Navy, Office of the Chief of Naval Operations. (2013b). *Maritime operations center*, NTTP 3-32.1. Washington, DC: Author.
- Dugan, K. (2007). *Navy mission planner*. MS Thesis, Operations Research, Naval Postgraduate School, Monterey, CA.

- GAMS Development Corporation, 2019. *General algebraic modeling system*. Retrieved from www.gams.com
- Hallmann, F. (2009). *Optimizing operational and logistical planning in a theater of operations*, MS Thesis, Operations Research, Naval Postgraduate School, June.
- Hart, William E., Carl D. Laird, Jean-Paul Watson, David L. Woodruff, Gabriel A. Hackebeil, Bethany L. Nicholson, & John D. Sirola. *Pyomo – Optimization modeling in python*, 2nd ed. Vol. 67. Springer, 2017.
- Joint Chiefs of Staff, 2001. *Department of Defense dictionary of military and associated terms*, Joint Publication 1-02 (amended 2008).
- Joint Chiefs of Staff, 2011. *Joint operation planning*, Joint Publication 5-0. Washington, DC: Author.
- McKinney, W., 2010. Data structures for statistical computing in python, *Proceedings of the 9th Python in Science Conference*, Retrieved from <https://pandas.pydata.org>
- Python Core Team, 2019. Python: A dynamic, open source programming language. *Python Software Foundation*, Retrieved from <https://www.python.org/>
- Silva, R.A. (2009). *Optimizing multi-ship, multi-mission operational planning for the joint force maritime component commander*, MS Thesis, Operations Research, Naval Postgraduate School, Monterey, CA.
- Swift, S.H., 2018. A fleet must be able to fight. *Proceedings*, Retrieved from <https://www.usni.org/magazines/proceedings/2018-05/fleet-must-be-able-fight>
- VBA programming in Office, 2019. Retrieved from <https://docs.microsoft.com/en-us/office/vba/api/overview/>
- Von Moltke, H. (1871). “On Strategy”/“Über Strategie,” as translated in Moltke on the *Art of War: Selected Writings*, 1993 by Daniel J. Hughes and Harry Bell, p. 92; German collected in: *Kriegsgeschichtliche Einzelschriften*, H.13, 1890, hier zitiert nach: *Militärische Werke*, Band 2, Teil 2. Mittler & Sohn Berlin 1900.
- Wikipedia contributors, 2019. Comma-separated values. *Wikipedia*. Retrieved March 14, 2019, from https://en.wikipedia.org/w/index.php?title=Comma-separated_values&oldid=886749600
- xlwings, 2019. *xlwings – Make Excel Fly!* Retrieved from <https://docs.xlwings.org/en/stable/index.html>

INITIAL DISTRIBUTION LIST

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