2018-06

# SCHEDULING AMPHIBIOUS CONNECTORS TO DELIVER MULTIPLE COMMODITIES

Danielson, Matthew E.

Monterey, CA; Naval Postgraduate School

http://hdl.handle.net/10945/59643

# NAVAL POSTGRADUATE SCHOOL

**MONTEREY, CALIFORNIA**

# THESIS

**SCHEDULING AMPHIBIOUS CONNECTORS TO DELIVER MULTIPLE COMMODITIES**

by

Matthew E. Danielson

June 2018

| | |
|---|---|
| Thesis Advisor: | Michael P. Atkinson |
| Co-Advisor: | Kyle Y. Lin |
| Second Reader: | Emily M. Craparo |

THIS PAGE INTENTIONALLY LEFT BLANK

| **REPORT DOCUMENTATION PAGE** | | | *Form Approved OMB No. 0704-0188* |
|---|---|---|---|



| **1. AGENCY USE ONLY** *(Leave blank)* | **2. REPORT DATE** June 2018 | **3. REPORT TYPE AND DATES COVERED** Master's thesis |
|---|---|---|

**4. TITLE AND SUBTITLE**
SCHEDULING AMPHIBIOUS CONNECTORS TO DELIVER MULTIPLE COMMODITIES

**5. FUNDING NUMBERS**
W8A06

**6. AUTHOR(S)** Matthew E. Danielson

**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)**
Expeditionary Energy Office, Quantico, VA 22134

**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)**

Across the amphibious warfighting community, planners need to quickly develop schedules to deliver multiple supply commodities during ship-to-shore operations. We extend the single-commodity models underpinning the Marine Expeditionary Unit (MEU) Amphibious Connector Scheduler (MACS) tool to allow amphibious connectors to deliver multiple commodities, such as fuel, food, water, and ammunition. This extension provides amphibious planners with a flexible and versatile decision tool, which is much more operationally relevant compared to the one-commodity variant. Our primary contribution is a temporal network flow model that computes the optimal number of round trips for each connector configuration from the seabase to the shore to achieve both fast delivery and few connector trips. We formulate the optimization problem as a mixed integer linear program, and develop a linear program approximation to solve large-sized problems. Our approach generates several different schedules with different strengths, so the decision maker can choose the one that best meets mission requirements. Through our analysis of complex notional scenarios and a historical case study, we demonstrate the potential of the improved MACS as an amphibious planning tool of the future.

**14. SUBJECT TERMS**
sustainment, amphibious operations, MEU, scheduling, linear programming, connectors, ship-to-shore

**15. NUMBER OF PAGES**
169

**16. PRICE CODE**

| **17. SECURITY CLASSIFICATION OF REPORT** | **18. SECURITY CLASSIFICATION OF THIS PAGE** | **19. SECURITY CLASSIFICATION OF ABSTRACT** | **20. LIMITATION OF ABSTRACT** |
|---|---|---|---|
| Unclassified | Unclassified | Unclassified | UU |

NSN 7540-01-280-5500 Standard Form 298 (Rev. 2-89)
Prescribed by ANSI Std. 239-18

THIS PAGE INTENTIONALLY LEFT BLANK

**Approved for public release. Distribution is unlimited.**

**SCHEDULING AMPHIBIOUS CONNECTORS TO DELIVER MULTIPLE COMMODITIES**

Matthew E. Danielson
Captain, United States Marine Corps
BS, U.S. Naval Academy, 2010

Submitted in partial fulfillment of the
requirements for the degree of

**MASTER OF SCIENCE IN OPERATIONS RESEARCH**

from the

**NAVAL POSTGRADUATE SCHOOL**
**June 2018**

Approved by: Michael P. Atkinson
Advisor

Kyle Y. Lin
Co-Advisor

Emily M. Craparo
Second Reader

Patricia A. Jacobs
Chair, Department of Operations Research

THIS PAGE INTENTIONALLY LEFT BLANK

# ABSTRACT

Across the amphibious warfighting community, planners need to quickly develop schedules to deliver multiple supply commodities during ship-to-shore operations. We extend the single-commodity models underpinning the Marine Expeditionary Unit (MEU) Amphibious Connector Scheduler (MACS) tool to allow amphibious connectors to deliver multiple commodities, such as fuel, food, water, and ammunition. This extension provides amphibious planners with a flexible and versatile decision tool, which is much more operationally relevant compared to the one-commodity variant. Our primary contribution is a temporal network flow model that computes the optimal number of round trips for each connector configuration from the seabase to the shore to achieve both fast delivery and few connector trips. We formulate the optimization problem as a mixed integer linear program, and develop a linear program approximation to solve large-sized problems. Our approach generates several different schedules with different strengths, so the decision maker can choose the one that best meets mission requirements. Through our analysis of complex notional scenarios and a historical case study, we demonstrate the potential of the improved MACS as an amphibious planning tool of the future.

THIS PAGE INTENTIONALLY LEFT BLANK

## TABLE OF CONTENTS

THIS PAGE INTENTIONALLY LEFT BLANK

## LIST OF FIGURES

# LIST OF TABLES

THIS PAGE INTENTIONALLY LEFT BLANK

# LIST OF ACRONYMS AND ABBREVIATIONS

| | |
|---|---|
| AAV | Amphibious Assault Vehicle |
| ACE | Air Combat Element |
| ARG | Amphibious Ready Group |
| ASOM | Assault Support Optimization Model |
| ATV | Amphibious Tracked Vehicle |
| BLT | Battalion Landing Team |
| CE | Command Element |
| CLP | Combat Logistics Patrol |
| CSV | Comma Separated Value |
| EWS | Expeditionary Warfare School |
| FOB | Forward Operating Base |
| GCE | Ground Combat Element |
| HMMWV | High Mobility Multipurposed Wheeled Vehicle |
| ILP | Integer Linear Program |
| LAV | Light Armored Vehicle |
| LCAC | Landing Craft Air Cushion |
| LCE | Logistics Combat Element |
| LCU | Landing Craft Utility |
| LHD | Landing Helicopter Dock |
| LPD | Landing Platform Dock |
| LSD | Landing Ship Dock |
| LVSR | Logistic Vehicle System Replacement |
| LZ | Landing Zone |
| MACS | MEU Amphibious Connector Scheduler |
| MACS-MC | MEU Amphibious Connector Scheduler Multi Commodity |
| MAGTF | Marine Air Ground Task Force |
| MCPP | Marine Corps Planning Process |
| MEB | Marine Expeditionary Brigade |

| | |
|---|---|
| MEU | Marine Expeditionary Unit |
| MPF | Maritime Prepositioning Force |
| MRE | Meal Ready to Eat |
| MTVR | Medium Tactical Vehicle Replacement |
| NM | Nautical Mile |
| OTH | Over the Horizon |
| PAX | Personnel |
| PB | Patrol Base |
| POL | Petroleum, Oils, and Lubricants |
| SBLDSS | Sea-Based Logistics Decision Support System |
| SMAAT | Simulation Mobility Modeling and Analysis Toolbox |
| SSTP | Ship-to-Shore Transportation Problem |
| USN | United States Navy |
| USMC | United States Marine Corps |

# EXECUTIVE SUMMARY

Conducting offensive warfare via amphibious operations involves the landing of embarked ground forces onto an enemy beach. Notable for their speed and element of surprise, these operations are used to secure a foothold in an enemy-controlled area for the rapid buildup of a much larger force for follow-on operations. In this thesis, we develop models and algorithms that focus on facilitating an amphibious planner's ability to provide sustainment of forces ashore. Specifically, we focus on the transportation of supply materials from seabases to shore to meet the demands of a landing force operating ashore.

The current planning process remains rooted in rudimentary methods that differ only slightly from techniques used during the Second World War. This research focuses exclusively on the doctrine, tactics, and techniques used by the United States Marine Corps (USMC) and United States Navy (USN) to employ and sustain a Marine Expeditionary Unit (MEU) ashore from a seabase. The overall objective is to develop a logistics model that schedules the delivery of multiple supply commodities from the seabase to the shore based on a MEU's demand ashore.

This research is a continuation of work accomplished by Major Robert Christafore, USMC, who developed the MEU Amphibious Connector Scheduler (MACS) to schedule delivery vehicles, called *connectors*, to transport bulk fuel [1]. We extend this work by allowing connectors to transport multiple supply commodities, such as fuel, food, water, and ammunition. Our primary contribution is the formulation of multi-commodity temporal network flow that sends supplies from the seabase to shore as quickly as possible. We refer to this model as the Quickest Flow Model. The supply nodes in the network are various connector types and the demand nodes are locations ashore that require supplies such as beaches, landing zones, and forward operating bases. The final output is a minute-by-minute delivery schedule that routes connectors to various land nodes.

A key challenge is converting the flow of supplies in our network to a list of connector runs, each packed with various combinations of supplies. For each connector platform we define several load plans, or *configurations*. For example, a configuration of

a landing craft air cushion (LCAC) could consist of a load with 150 personnel, or 1,000 gallons of water and four vehicles, or a single Abrams tank. The final output from the Quickest Flow Model is the number of runs of each connector configuration to each land node. We formulate a mixed integer linear program to compute the optimal number of connector runs. For larger problems, when the mixed integer linear program cannot produce a solution in a reasonable amount of time, we use a linear program approximation and several rounding methods to produce near-optimal solutions.

In the Quickest Flow Model, there are two conflicting goals. On the one hand, we want to deliver as much supply as quickly as possible. On the other hand, we want to deploy as few connector runs as possible. To achieve both goals, we set the objective function in the mixed integer linear program as a linear combination of the two. By tuning their relative weights, we can generate different schedules with different strengths. This feature allows a user to produce MACS schedules that could favor faster deliveries, or surface over air connectors, or place a greater reliance on the ground network infrastructure. An amphibious planner can choose the schedule that best meets mission specific needs.

We demonstrate the versatility and efficiency of MACS by analyzing several notional scenarios. These demonstrations reflect the flexibility of our algorithms to accurately account for supply compatibility restrictions, vast numbers of configuration permutations, multiple supply commodities, and the ability to accurately produce a schedule of amphibious connector deliveries. Finally, we provide a comparison of our model output against a historical example to demonstrate the improvements MACS can provide to planners for the rapid and accurate development of a ship-to-shore landing schedule.

There are a few possible future research directions. First, an improved user interface will make this tool easier to use. Second, a more sophisticated rounding mechanism for the linear program may further improve the quality of the approximated solution. Third, one may want to schedule connectors based on an ordered list of connector priorities. Fourth, by expanding the research to include the use of amphibious personnel carriers and to integrate with concurrent research that produces robust delivery schedules (accounting for variable maintenance, sea states, enemy interference), planners will have a more accurate model that can account for all phases of an amphibious operations.

**Reference**


[1] R. M. Christafore, “Generating ship-to-shore bulk fuel delivery schedules for the Marine Expeditionary Unit.” M.S. thesis, Dept. of OR, NPS, Monterey, California USA, 2017. [Online]. Available: https://calhoun.nps.edu/handle/10945/55582

THIS PAGE INTENTIONALLY LEFT BLANK

# ACKNOWLEDGMENTS

I am extremely grateful to the faculty of the Naval Postgraduate School's Operations Research Department. Your professionalism and steadfast dedication to academic excellence is clear to all who have had the privilege of being educated under your charge. I am particularly grateful to the many excellent professors I had the opportunity to work with and who pushed me beyond what I considered to be my intellectual limits. I am confident that the instruction I received well benefit me no end in the years to come.

I would be remiss if failed to take this opportunity to thank Professor Atkinson for his invaluable mentorship, instruction, and patience. Your personal dedication to my academic achievement and the countless hours you poured into my thesis work are greatly appreciated. This research would have been impossible for me to complete without you. I wish you the very best in the years to come and hope to collaborate with you again in the future.

Most importantly, to my amazing wife, Jackie. Your love and encouragement have been the driving force behind my every success in the Marine Corps. I owe you more than you could possibly imagine. You make me a better man, Marine, and husband on a daily basis and no challenge is ever insurmountable with your support.

THIS PAGE INTENTIONALLY LEFT BLANK

# I. INTRODUCTION

Conducting offensive warfare via amphibious operations involves the landing of embarked ground forces onto an enemy beach. Notable for their speed and element of surprise, these operations are used to secure a foothold in an enemy-controlled area for the rapid buildup of a much larger force for follow-on operations. In this thesis, we develop models and algorithms that focus on facilitating an amphibious planner's ability to provide sustainment of forces ashore. Specifically, we focus on the transportation of materials from seabases to shore to meet the supply demands of a landing force operating ashore.

## A. AMPHIBIOUS OPERATIONS

Amphibious operations provide a strategic capability that can be employed across the globe at short notice to seize advanced naval bases for follow-on land operations [1]. The Marine Expeditionary Unit (MEU) is the permanent standing amphibious organization within the United States military and can deploy rapidly. The advantages gained in flexibility, speed, and surprise of amphibious operations are acutely apparent when studying the Island-Hopping Campaign in the Pacific or the Normandy landings of the Second World War.

During the conflicts in Iraq and Afghanistan (2003–2014) the Marine Corps lost touch with its amphibious roots as the preponderance of the force was focused on supporting ground combat operations. Recently a paradigm shift has occurred within the organization to educate and train the force for amphibious warfare [1]. Marine Corps officers desiring to be competitive for promotion to the rank of Major must complete the Expeditionary Warfare School curriculum [2]. This course inculcates all Marine Officers with the fundamentals of amphibious operations and planning, including an in-depth overview of the planning efforts and products required to successfully land a force ashore.

The renewed emphasis on educating the force for amphibious operations is directly linked to the difficulty of planning and executing such operations. Landing a force ashore involves the movement of all the personnel, equipment, and supplies needed to conduct combat operations.

The current planning process remains rooted in foundational methods that differ only slightly from techniques used during the Second World War. This research focuses exclusively on the doctrine, tactics, and techniques used by the United States Marine Corps (USMC) and United States Navy (USN) to employ and sustain a MEU ashore from a seabase. The overall objective is to develop a logistics model that schedules the delivery of multiple supply commodities from the seabase to the shore based on a MEU's demand ashore.

## B. MOTIVATION

During most land operations, the most difficult aspect of a logistics planner's job is estimating consumption rates to determine the daily resupply demand. However, for amphibious operations, the estimate of daily demand is just the beginning of the planning process. Due to the rapid nature of the Marine Corps' maneuver warfare and the need to maximize utility of organic supplies in an expeditionary environment, efficiencies in amphibious sustainment operations must be found, to reduce superfluous or late deliveries of needed sustainment. Furthermore, due to the limited space on a MEU, the Logistics Combat Element (LCE) of a MEU is smaller than its equivalent land based units, which further strains the planners. Our work can help improve efficiencies in the planning process.

## C. BACKGROUND

This research is follow-on work to Christafore's [3] 2017 thesis "Generating Ship-to-Shore Bulk Fuel Deliveries for the Marine Expeditionary Unit." Christafore successfully developed a model to schedule one commodity type, bulk fuel, for delivery using waterborne and air platforms to sustain operations ashore. He accounted for transit times, vehicle capacity, and demand ashore in his model. However, Christafore's thesis only considers the delivery of bulk fuel vice all commodities required by a MEU.

While useful, Christafore's model is unrealistic because the delivery vehicles, called connectors, in amphibious sustainment operations are rarely dedicated to the exclusive delivery of one commodity type. Marine Corps doctrine and historical examples indicate that efficient and simultaneous delivery of multiple classes of supply is needed to properly sustain a force ashore. Our research expands upon Christafore's by extending his model to accommodate the delivery of multiple supply commodities.

This is a difficult problem as connector platforms moving multiple commodities ashore can be loaded many ways, in what we refer to as *configurations*. These configurations must account for various constraints, such as multiple commodities that cannot travel together because of restrictions unrelated to weight and space. For example, there are several types of ammunition that cannot be transported together and certain fuels and lubricants that cannot travel with ammunition. Our work captures the requirements for packaging and transporting supplies to produce realistic connector and configuration utilization based on embarkation restrictions.

Furthermore, if a certain supply commodity (e.g., ammunition) has a priority such that it must come ashore before others, the configuration and connector selection must accommodate the prioritized delivery schedule. Reality has shown that, in almost every sustainment operation, the supported unit will prefer which supply item reaches it first. These considerations must be reflected within the new model to generate an accurate schedule for supply delivery.

Improving Christafore's model to a realistic representation of sustainment operations requires us to effectively capture the loading capabilities and limitations of each platform type. Each commodity type will have different weights, dimensions, and compatibility requirements that need to be addressed. We develop a model and algorithms to successfully reflect these requirements and produce an efficient schedule to meet shore-based demand.

## D. APPROACH

We consider a scenario where a MEU is conducting prolonged operations ashore. A MEU consists of roughly 2,200 Marines and sailors. The scenario assumes that all combatant units are ashore with their authorized equipment and basic levels of supply. Our analysis focuses on the daily sustainment operations to replenish supplies.

Daily demands are made across multiple commodities at various locations ashore. These commodities include fuel, ammunition, water, food, personnel, and equipment. Our first objective is to use organic MEU air and waterborne craft to deliver required supplies as quickly as possible to their designated locations. We next adjust the model to account for prioritized deliveries.

We develop a temporal multi-commodity network flow problem as the basis for our modeling approach [4]. Land nodes represent demand locations that require a certain amount of each commodity. These nodes include forward operating bases (FOBs), beaches, and landing zones (LZs). These nodes vary in their ability to accommodate aircraft and water-based surface connectors. Some nodes may only be accessible via land links, such as truck deliveries via roads. The supply originates from the seabase and is pushed ashore via the connectors. We use historical data on amphibious operations and current Marine Corps logistic planning factors to estimate demand and connector characteristics.

The model outputs not just the delivery schedule, but also the configuration of each delivery. This process allows us to model an efficient delivery process when certain classes of supply have a higher level of delivery urgency.

We incorporate heuristics into the overall formulation, where appropriate, to ensure our algorithms run in a reasonable amount of time. It is important that our model can be used by planners in real time. The final output is a comprehensive schedule reflecting departure and arrival times of all amphibious connectors with their respective load plans.

## E. RELATED WORK

As previously discussed, this thesis is a continuation of work completed by Christafore in 2017 [3]. In his research, Christafore developed the MEU Amphibious Connector Scheduler (MACS) that schedules the delivery of bulk fuel via surface and air connectors associated with an Amphibious Ready Group (ARG). Christafore’s model has three components:

1. Quickest Flow Algorithm: Temporal network flow model.

2. Assignment Heuristic: The heuristic takes the number of connectors produced by the Quickest Flow Algorithm and schedules their runs to respective beaches or LZs in a naive fashion.

3. Scheduling Algorithm: Linear program that schedules all the sequenced runs ashore.

Christafore assumed constant and favorable weather conditions and no maintenance related impacts to connector availability [3]. Furthermore, he assumed all support equipment such as transport trucks had already been offloaded ashore. This thesis carries these assumptions forward. Our primary contribution is the incorporation of multiple commodities into the analysis.

In his research, Kearns (USN) developed a Simulation Mobility Modeling and Analysis Toolbox (SMMAT) to inform decision makers on the impact weather and distances have on LCAC operations [5]. His model consisted of six discrete steps every LCAC undertook when delivering supplies ashore, a method that this thesis replicated for all connectors. These six steps consisted of arriving, loading, offloading, and departing either a ship or beach. This thesis expanded on this six-step approach by extending it to multiple delivery locations (beaches, LZs, FOBs) and multiple connector platforms.

In research closely mirroring Christafore's, Powell (USMC) created an Assault Support Optimization Model (ASOM) to determine the optimal combination of CH-53Es and MV-22s needed to support a Ground Combat Element (GCE) [6]. Powell used a time-expanded network [4] to control movements, which resembles the approach that Christafore [3] took and one that is replicated in this thesis. Additionally, Powell focused on the delivery of multiple commodities [6]. However, to avoid compatibility issues, Powell normalized all deliveries into pounds and was unable to distinguish when certain commodities were moving or if the loadout of the aircraft was feasible. A major contribution of our work is to recognize that not all supplies can be lumped together due to compatibility restrictions.

In 1999, Reitter (USMC) developed the Sea-Based Logistics Decision Support System (SBLDSS), which mirrored Powell's work in that it exclusively focused on sustaining forces ashore via the use of air connectors [7]. A series of heuristics was applied to calculate the consumption of the unit and then determined the resupply quantities needed [7]. Focusing on maintaining stockage objectives (designated amount supplies that must be maintained) ashore, aircraft were scheduled to deliver supplies to keep the assigned unit above a predetermined stockage objective. Of note, Reitter implemented a priority for resupplies that, when triggered, would jump a scheduled resupply in the list to meet this emergent requirement. We took a similar approach to assign a priority of resupply for each of the amphibious connectors.

Work completed by D. Powell (USMC) in 2004 modeled the sustainment of a Marine Expeditionary Brigade (MEB) operating ashore from ship to shore via a shuttle method of varying Maritime Prepositioning Force (MPF) ships [8]. Powell adjusted the configurations of ships to account for the varying amounts of ammunition, food, fuel, water, and equipment needed to sustain a MEB. The greatest takeaway from Powell's work, and what we modeled with this thesis, is the compatibility issues that arise when trying to move certain types of commodities on the same amphibious connector.

Ward (USN) completed a mixed-integer optimization model in 2008 to model the ship to shore transportation problem (SSTP) associated with a hospital ship [9]. Ward's model focused on optimizing the transportation of personnel to and from a hospital ship using boats, helicopters, and tactical vehicles. Ward's amphibious network and resulting schedule of optimized vehicle and amphibious craft for the delivery of personnel was similar to Christafore's; as a result, it forms a foundation for the work this thesis expands upon.

## F. THESIS ORGANIZATION

In Chapter II, we provide an overview of amphibious operations, Chapter III provides a detailed description of the approach we took to develop and expands the model created by Christafore to account for multiple commodities. Chapter IV details techniques that we develop to enhance delivery plan efficiency. Chapter V demonstrates our model numerically via a case study. Lastly, Chapter VI provides conclusions of our work, and recommends follow-on research. We include several lengthy appendices that provide supplemental information and analysis.

THIS PAGE INTENTIONALLY LEFT BLANK

# II. BACKGROUND ON AMPHIBIOUS OPERATIONS

In this chapter, we describe the components of amphibious operations. An amphibious operation is a joint operation between the Navy and Marine Corps to transport land forces via the use of Navy shipping to a designated target beachhead. Once within 20 nautical miles of this target area the landing force, with all its equipment and personnel, deploys to the beach across various air and surface connector platforms. It is during this transfer ashore that the landing force is most vulnerable, highlighting the need to quickly and efficiently oversee their delivery. Understanding the varying types of equipment and craft that carry and deliver a MEU with their supplies and equipment is critical to understanding the detailed planning this work facilitates. The successful sustainment of amphibious operations involves the successful planning and synchronization of the following five interrelated items:

1. The ARG, which is a squadron of three ships designated to carry a MEU.

2. Two categories of surface and air connectors that are responsible for moving all required personnel, equipment, and supplies ashore.

3. Establishment of Marine land network consisting of beach heads to receive surface connectors, LZs for aircraft, and Patrol Bases (PBs)/FOBs where supplies are consumed, and stored.

4. Needed sustainment for forces ashore that can be broken down into nine separate and distinct classes of supply.

5. Existing restrictions as they pertain to what commodities may be transported with others to facilitate their safe delivery.

This chapter provides a brief overview of the five components listed above that are associated with the sustainment of a MEU ashore. For greater detail on the individual equipment, containers, and planning process refer to Christafore’s [3] 2017 work.

## A. AMPHIBIOUS READY GROUP

The ARG is a composite squadron of amphibious Navy ships responsible for the embarkation and forward movement of the MEU to their final objective area. The current ARG consists of three ships: of a Landing Helicopter Dock (LHD), Landing Ship Dock (LSD), and Landing Platform Dock (LPD). The ARG transports all personnel, equipment, and supplies needed to support a MEU operating ashore for up to fifteen days.

## B. AMPHIBIOUS CONNECTORS

Amphibious connectors are vehicles used by the USN and USMC to move personnel, equipment, and supplies from sea to shore. These connectors are broken down into two categories: surface and air. While this work focuses primarily on the use of the amphibious connectors transporting items ashore, supplies are moved further inland via the use of land connectors known as tactical vehicles that are organized into convoys. Understanding the capabilities and restrictions of convoys is important to appreciate how supplies are moved between land nodes in our amphibious network.

### 1. Surface Connectors

The two surface connectors used in our model are the Landing Craft Air Cushion (LCAC) and Landing Craft Utility (LCU). LCACs and LCUs can both be stored in any of the ARG's three ships. The two platforms differ dramatically in speed and payload. The LCAC is known for speed as its hover capabilities allow it to move four times faster than the LCU [10]. However, what the LCU lacks in speed it makes up for in payload, capable of delivering three times the weight and volume of an LCAC [10].

### 2. Air Connectors

The two air connectors used in our model fall under the command of the MEU's Air Combat Element (ACE). The platforms consist of the CH-53E Super Stallion and the MV-22 Osprey. Due to their speed of delivery, these platforms are used frequently to transport personnel and urgent supplies. Air platforms are regularly used to supplement the flow of supplies provided by surface connectors with an added emphasis on making deliveries further inland from designated beach heads to reduce delivery time.

### 3. Land Connectors

While this thesis focuses primarily on the use of surface and air connectors, it is important to appreciate the specific capabilities and restrictions associated with transporting supplies on land. Once supplies and equipment are delivered ashore via either surface or air connector they are further distributed by tactical vehicles. This means of distributing assets via tactical vehicles is known as Combat Logistics Patrols (CLPs). CLPs are not doctrinally organized but instead are formed based on the set of equipment on hand and the forecasted need of supported units. For this model, we assume that CLPs consist of several MTVRs and are available at certain land nodes (FOBS/PBs/LZs, etc.), and are used to push supplies further inland.

## C. SUPPLY COMMODITIES

The Marine Corps organizes its supply system into nine separate groups of like items called classes. These classes of supply account for all the basic needs required of any Marine unit's ground supply system and are the items needed to replenish a force conducting any type of operation. However, this thesis largely focuses on the transportation and delivery of [11]:

- Class I: Food and Water
- Class III: Petroleum, Oils, and Lubricants (POLs)
- Class V: Ammunition
- Class VII: Major End Items

This section provides greater detail as to the compatibility issues that arise with the movement of certain supply items, their means of storage for transportation, as well as a basic overview of the configurations each of the amphibious and land connectors must adapt to transport certain commodities.

#### *a. Supply Compatibility*

There are certain restrictions on how some supplies can be transported. These restrictions are largely based on the safety of the individual transporting the supplies. Our model takes these specific restrictions into consideration with the development of connector configurations, which is discussed in greater detail in Section II.D.

(1) Ammunition

There are certain categories of ammunition that have compatibility issues that prohibit them from being transported together on the same connector. These types of compatibility issues are normally associated with explosives. Additionally, ammunition will never be transported with any type of POL. Therefore, any tactical vehicle, aircraft, or watercraft that has fuel cargo will not have any ammunition.

(2) Personnel (PAX)

Very few compatibility issues arise when transporting personnel; however, unlike LCUs, LCACs, and aircraft that can transport personnel and supplies simultaneously, no tactical vehicle can transport both personnel and cargo. Furthermore, no personnel will travel aboard any aircraft that is transporting bulk POLs. No issues arise with the transportation of American service members and ammunition.

(3) Food and Water

Food and water can be transported across any amphibious connector with any other supply commodity so long as they are stored in separate locations to avoid any potential contamination from a POL leakage. However, when being transported with tactical vehicles, no food item can travel with any POL.

#### *b. Methods of Transporting Supplies*

This subsection elaborates on additional packing instruments that were not used in Christafore's work but have been included in this work to transport supplies from the seabase to shore based demand nodes. For more information on other mechanisms of supply transportation see Chapter II of Christafore's work [3].

#### (1) Bulk Liquids

The collapsible coated fabric fuel and water drums can hold 500 gallons of bulk liquid [12]. These drums can either be lifted by an aircraft or be towed behind an MTVR/LVSR. The M149 Trailer, commonly referred to as a water-bull, is a tactical trailer capable of transporting up to 400 gallons of potable or non-potable water [13].

#### (2) Meals Ready to Eat (MRE)

The primary food source used by Marines operating ashore is the MRE. Individual MREs are packaged into cases of 12, which are then loaded onto warehouse pallets. Warehouse pallets can hold 48 cases of MREs and are routinely transported via MTVR. By doctrine ARGs contain sufficient MREs aboard amphibious shipping to support the entire MEU ashore for up to fifteen days.

#### (3) Ammunition

The normal means of transporting ammunition resembles that of MREs: using pallets. The cases of ammunition are then stacked and secured on warehouse pallets with varying weights depending on the type of ammunition being transported.

## D. CONFIGURATIONS

There are many ways to load amphibious connectors with supplies, equipment, and personnel. For our model, we enumerate several common configurations for each of the platforms, which adhere to all compatibility, size, and weight constrictions.

### 1. LCU/LCAC Configurations

LCACs and LCUs have two basic configurations: personnel or cargo. The LCAC can carry up to 180 personnel at a time; however, this requires the installation of a personnel carrier aboard the LCAC, which negates the transportation of any cargo. Likewise, the LCU can hold up to 300 personnel for delivery; however, this takes up all its cargo space. Supplies being transported by both platforms can be either mobile loaded on a tactical vehicle or secured to the deck of each craft. It is faster to have all supplies mobile loaded; however, this is not always feasible due to tactical vehicle availability.

LCACs are always capable of moving up to 24 Marines when not configured for personnel transport. At maximum capacity the LCAC can carry either: one M1A1 Abrams tank, 11 HMMWVs, two MTVRs, or four LAVs [11]. By comparison, the LCU has a maximum capacity of carrying either: two M1A1 Abrams tanks, four MTVRs, 12 HMMWVs, or nine LAVs [11]. These numbers can be reduced to provide space for supplies with the understanding that compatibility restrictions addressed in Section II.C.a are not violated. Common configurations of LCACs include: one M1A1 tank, two MTVRs with towed howitzers, 11 HMMWVs, and two MTVRs preloaded with varying supplies on their bed space.

#### 2. CH-53E/MV-22 Configurations

The CH-53Es and MV-22s have two generic configurations: internal or external load. When transporting equipment or supplies externally via a sling each aircraft is incapable of transporting anything inside the aircraft. Likewise, the opposite is also true. The CH-53E has the ability to internally transport seven warehouse pallets of supplies compared to four for the MV-22 [14]. If some pallet positions are not being used, then personnel are permitted to travel with supplies (except for POLs). Internal loads can therefore consist of any combination of palletized supplies/personnel that meet the restrictions enumerated by Section II.C.a. Several common configurations include: the external load of a single HMMWV, eight–14 warehouse pallets of MREs (aircraft dependent), or an internal load equally split between ammunition pallets and passengers.

#### 3. CLP Configurations

CLPs consist of several MTVRs. MTVRs can be configured to carry either personnel or cargo. If configured for the transportation of personnel, an MTVR can carry 18 combat-loaded Marines in its bed space; these Marines are not permitted to travel with any cargo. When configured for carrying cargo, multiple commodities may travel simultaneously on the same bed space so long as compatibility issues are not violated. Similar items are grouped together for ease of loading, unloading, and storage. However, there are still many hundreds of possible configurations an MTVR may assume when transporting supplies.

## E. CONTRAST TO PREVIOUS WORK

We expand the original model developed by Christafore to include multiple supply commodities and platform configurations to develop a realistic schedule for the ship-to-shore delivery process needed to sustain a MEU ashore.

In Christafore's model a single commodity, fuel, is delivered to satisfy MEU consumption [3]. Once a MEU begins to conduct land operations, its consumption of supplies will encompass all nine classes of supply in addition to the need of additional personnel. As such, it is unrealistic to assume that connectors will deliver a single commodity at a time as multiple types are needed simultaneously, and platforms should be scheduled using different platform configurations to meet this demand.

Another contrast to Christafore's is the use of platform configurations to transport supplies. In Christafore's model a single configuration was used for each amphibious connector to transport fuel; namely each connector transported 100% fuel [3]. There are many hundreds, if not thousands, of configurations each platform can assume when transporting multiple commodities; however, this work focuses on defining a reasonable number of common configurations.

By including multiple commodities, we must incorporate a prioritized way to deliver supplies to meet the most urgent needs of the MEU. This is accomplished by organizing our delivery schedule to ensure the prioritized demand is most rapidly satisfied.

THIS PAGE INTENTIONALLY LEFT BLANK

# III. MATHEMATICAL FORMULATION

In this chapter we present the mathematical formulation to expand the models and algorithms underlying the original MACS tool developed by Christafore [3]. Our primary contribution is incorporating multiple supply commodities into the modeling framework to create the MACS multi-commodity (MACS-MC). The original MACS tool consists of three modeling components, , and we build upon each of those models in our work as we create MACS-MC. We label these three components:

1. Quickest Flow Model
2. Assignment Heuristic
3. Scheduler Linear Program

Each component is a standalone model, and we discuss each separately in its own section in this chapter. We focus on incorporating multiple commodities, which requires new input parameters, multiple configurations for each connector type, and specific loading and unloading requirements associated with transporting multiple commodities simultaneously.

The final output is a minute-by-minute schedule of each *run* for each amphibious connector type. We define a run as the completion of a round trip delivery for a single connector. For example, a run would encompass the entire process of an LCAC departing a ship, delivering its supplies ashore, and then returning to the ship. Unlike the original formulation, our model specifies the connector configuration used on each run. For example, the first run may use a configuration A LCAC while the second run uses a configuration B LCAC.

In this chapter, we assume the parameters of each model are given. In reality some of these parameters are derived from user inputs. These user inputs include the number of connectors available, connector characteristics (e.g., velocity and capacity), demand requirements ashore, and distances between demand locations. We defer a full discussion of user inputs until Chapter V and here highlight some of the new inputs required for our modeling effort.

1. The configurations for each connector type. Each configuration specifies the quantity of each commodity a connector can carry in a single run.

2. Loading and unloading times for each configuration.

3. Demand and storage requirements for multiple supply commodities at each land node.

4. Introduction of CLPs that deliver multiple commodities simultaneously between land nodes.

Figure 1 provides a high-level overview of the three models.

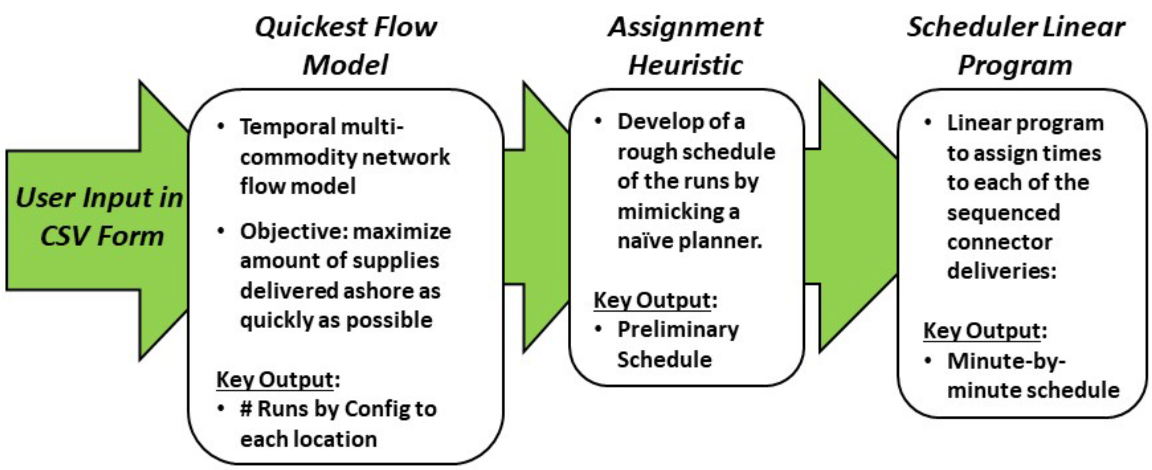



Figure 1. Overview of MACS-MC Tool

The *Quickest Flow Model* is a temporal network flow model that seeks to maximize the amount of demand satisfied for all commodities as quickly as possible. The output of this model is the number of runs for each connector configuration to each land node. An example solution might involve five LCAC-A runs to Beach 1, three LCAC-B runs to Beach 1, and four LCAC-A runs to Beach 2. This information feeds into the *Assignment*

*Heuristic,* which creates a rough schedule by mimicking a myopic planner who generates the schedule in real time. This rough schedule is refined by the *Scheduler Linear Program* to create a minute-by-minute plan associated with every run to shore.

Our primary contribution is modifying the Quickest Flow Model to incorporate multiple commodities. This chapter primarily focuses on these modifications, which appear in Chapter III.A. We provide general overviews of the *Assignment Heuristic* and *Scheduler Linear Program* in Chapter III.B and III.C, respectively. Additional information on the latter two models appear in Chapter III of Christafore [3] and Appendices A and C.

## A. QUICKEST FLOW MODEL

Like Christafore's model, the primary output of the Quickest Flow Model linear program component is the number of runs (i.e., round trips) needed by connectors to satisfy demand. Rather than dealing directly with discrete runs, we approximate these discrete runs by a continuous flow of supplies. We can then leverage network flow models (e.g., see [3]) to solve this approximate system. For example, rather than modeling five LCAC runs, each carrying 3,000 gallons of fuel, over 10 hours, we assume that LCACs supply 15,000 gallons of continuous flow ashore over 10 hours. Once the Quickest Flow Model solves for the continuous flow, we must convert the flow to discrete runs. In the original one-commodity model, converting from flow to runs is trivial: in the above example with a fixed LCAC capacity of 3,000 gallons, 15,000 gallons in flow translates to five runs. It is more challenging to make this conversion from flow to runs in our multiple commodity model. Before describing the model formulation in Section III.A.2, we discuss how we modify the model to more easily convert from flow to runs.

### 1. Continuous Flow

In the original model Christafore [3] used the decision variable $X_{i,j,t}$ to track the flow of fuel between nodes *i* and *j* at time period *t*. As discussed in the introductory portion of this subsection, it is straightforward to use this flow variable to directly estimate the number of runs in the one-commodity model. We discuss in more detail in the next subsection our new commodity flow variable $X_{i,j,t,c}$, which captures the flow of commodity type *c* between nodes *i* and *j* at time period *t*. We can no longer use $X_{i,j,t,c}$ to directly determine the number of runs of each connector.

We now model the flow of runs separately. Since we transport multiple supply commodities across different connecter configurations, we directly track runs by configuration. That is, we define a new runs flow variable $Z_{s,l,g,t}$, which tracks the flow of runs to node *l* at time *t* using configuration *g* of connector type *s*. Just as with the commodity flow variable *X*, this *Z* variable is a continuous flow approximation of an underlying discrete entity. Rather than modeling five discrete runs of LCAC-A to Beach 1 over eight hours, we assume those five runs continuously flow over eight hours. By using this new decision variable, we can determine the total amount of supplies delivered by tracking the commodity flow of *X* and use *Z* to determine what configurations are used to satisfy demand. For example, consider five LCACs using configuration A to deliver 5,000 gallons of fuel and 4,000 gallons of water in four hours. The commodity flow of fuel and water would be captured by *X* variables, whereas the five runs of LCAC-A would be captured as Z variables. A key part of our modeling formulation is ensuring that the flow of commodities (*X*) is consistent with the flow of runs (*Z*). By approximating the discrete runs via the continuous run-flow variable Z, the Quickest Flow Model will likely produce a non-integer number of runs over the course of the day. For example, the solution may generate 5.3 total runs during the day of LCAC-A to Beach 1. We then must perform additional rounding to ensure an integer number of runs. We discuss our rounding approaches in detail in Appendix B. One option is introducing an integer decision variable to force integrality, which we discuss in Section III.A.4.

### 2. Formulation

The Quickest Flow Model is a temporal approach to a classic network flow problem. Flow between nodes takes time, and our formulation maximizes the flow from source nodes to sink nodes within a specified time window. As such, there is the standard spatial component associated with a network, which includes nodes and arcs. In addition, our model includes a temporal component, where time is discretized into discrete periods. The level of temporal discretization can be as fine or coarse as the user desires. Our default discretization is 15 minutes per time period. The user specifies the time period of interest via the *maxTime* parameter and this generates the index $t \in T$, where $|T|$ is the number of time steps available for the algorithm. For example, if a user specifies that the model covers a 10-hour scenario ($maxTime = 10$), those 10 hours are broken into 40 discrete time periods, each lasting 15 minutes.

The Quickest Flow Model formulation begins with an amphibious network consisting of supply and demand nodes. Demand nodes are points on land that have a requirement for supplies. For example, these would represent different beaches, FOBs, and LZs, which have consumption requirements driving demand for various commodities. These demand nodes on land are denoted as $l \in L$. The first major distinction in our work from Christafore's is with the inclusion of the index $c \in C$, which reflects the specific supply commodity type (e.g., fuel, water, ammunition). Demand for each class of supply is defined as $demand_{l,c}$, while the storage capacity at each of the land nodes is defined as $nodeCap_{l,c}$. This is a change from the original MACS model as we are now tracking the storage and demand quantities for multiple supplies at each land node. As with many network flow models we include a global sink node. All the land nodes within the network are connected to this sink node, which we refer to as the *superSink*. The purpose of the *superSink* is to capture all satisfied demand for every commodity through the network.

The supply nodes we use in our model represent individual connector platforms. The amphibious connectors are represented as the index $s \in S$. For example, the connectors "CH-53," "MV-22," and "LCAC" would belong in the set *S*. Each type of connector has a maximum number of total runs that can be executed during the time window across all assets of that connector type. This parameter, which we denote as $maxRuns_s$, depends upon the number of connectors and time, distance, and speed calculations. An important distinction is that demand (i.e., $demand_{l,c}$) and supply (i.e., $maxRuns_s$) are tracked using different units. The demand relates to specific commodities (e.g., gallons of fuel), whereas supply is just a count of the number of runs. This relates to the discussion of the commodity flow variable *X* and the run flow variable *Z* discussed in

Section III.A.1. We need to convert from runs to commodities. We discuss this issue more in a few paragraphs and illustrate with an example.

The supply and land nodes, to include the $superSink$, are grouped into our set of nodes *N*, which is indexed by *n*, *i*, and *j*. We use *N* to build our set of arcs $(i, j) \in A$. Arcs between land nodes ($i, j \in L$) are given by the specific geographic topology, with *L* being our set of land nodes. Arcs emanating from a supply node ($i \in S$) only exist if the target node is accessible by the corresponding connector type. All arcs have distances associated with them. An example of a basic amphibious network is represented in Figure 2.

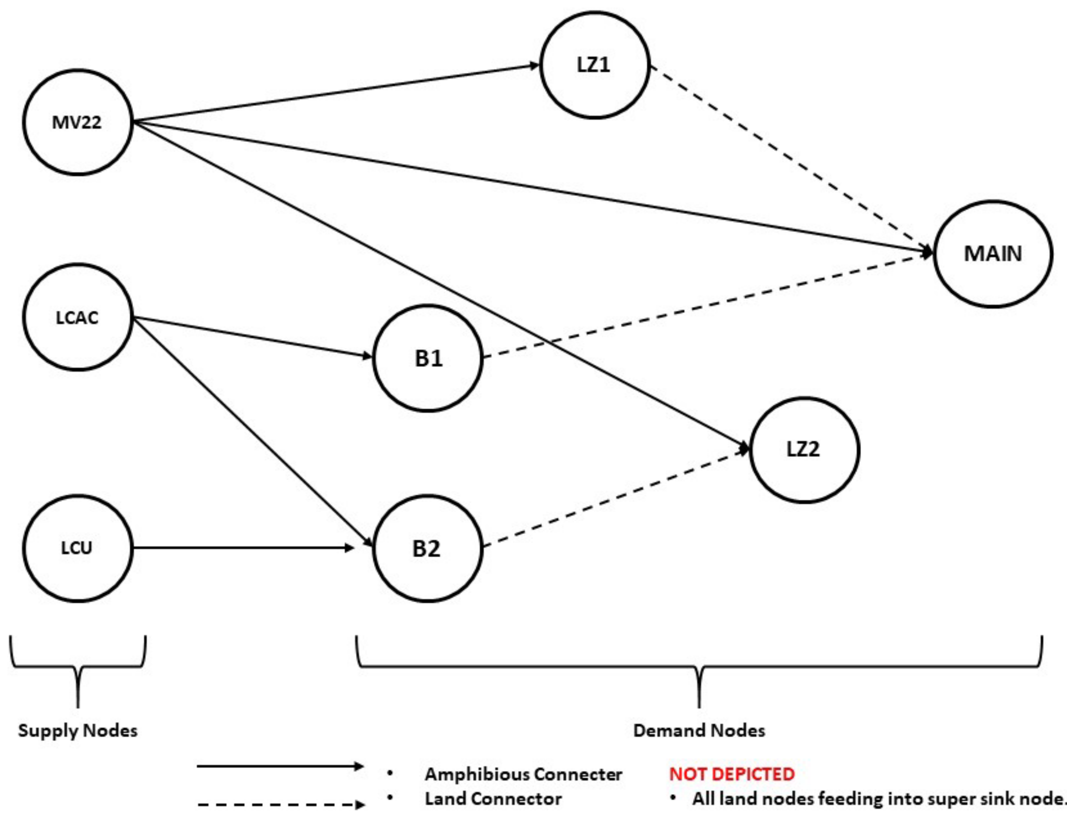



Figure 2. Simple Example of an Amphibious Network

Additionally, we include the set $g \in Gs_s$ that indexes the configurations any connector-type *s* can use. Each configuration transports a certain maximum amount of each commodity (e.g., LCAC configuration A can transport 1,500 gallons of fuel and 25 Marines while configuration B transports 125 Marines only). The parameter $config_{s,g,c}$ specifies how much of commodity *c* is transported by configuration *g* of connector type *s*. This value is derived from data a user enters to enumerate every configuration for each connector and the requisite quantity of commodity *c* each can transport. For example, if an LCAC can deliver fuel, water, and ammunition simultaneously in varying quantities, then the user could enumerate multiple separate configurations and input the amount of each commodity that could be transported in each configuration. This closely resembles the parameter $f_c$, which describes the single configuration used by our standardized CLPs to deliver supplies between land nodes. That is, $f_c$, is the amount of commodity *c* carried on one CLP delivery. While our focus is on the air and surface network, the CLPs are an important asset that push supplies within the land network. For simplicity we assume that there is only one type of CLP configuration used at various locations throughout the land network.

Like the original model, we use the decision variable $X_{i,j,t,c}$ to reflect the flow of a given commodity being delivered within the network. The only change to this variable is that it is now indexed by *c* to reflect the amount delivered by each commodity type. Formally, $X_{i,j,t,c}$ is the flow of commodity *c* originating at time *t* at node *i* going to node *j*. As discussed in Section III.A.1, we introduce the new decision variable $Z_{s,l,g,t}$ to capture runs. $Z_{s,l,g,t}$ reflects the flow of runs of configuration *g* of connector type *s* originating at time *t* and heading to node *l*. This variable allows us to separate and identify the different configurations used for each platform.

Figure 2 illustrates the connection and conversion between the flow of runs (*Z*) and the flow of commodities (*X*). Consider an LCAC transporting supplies to Beach 1 using configuration A at time period four ($Z_{LCAC,Beach1,A,4} = 1$). To determine the total amount of commodity flow *X* for each class of supply *c* eventually delivered, we use the $config_{LCAC,A,c}$ parameter, which is we multiple the sum of each connector configuration used against their configuration capacity. An illustration of this process appears in Figure 3 with two additional configurations. Our objective is to satisfy demand for supplies and this relates to the commodity flow (*X*). However, our final plan should efficiently use connector runs

($Z$). Therefore, we include a penalty term in the objective function related to $Z$. We discuss this in greater detail after the mathematical definition of the model.

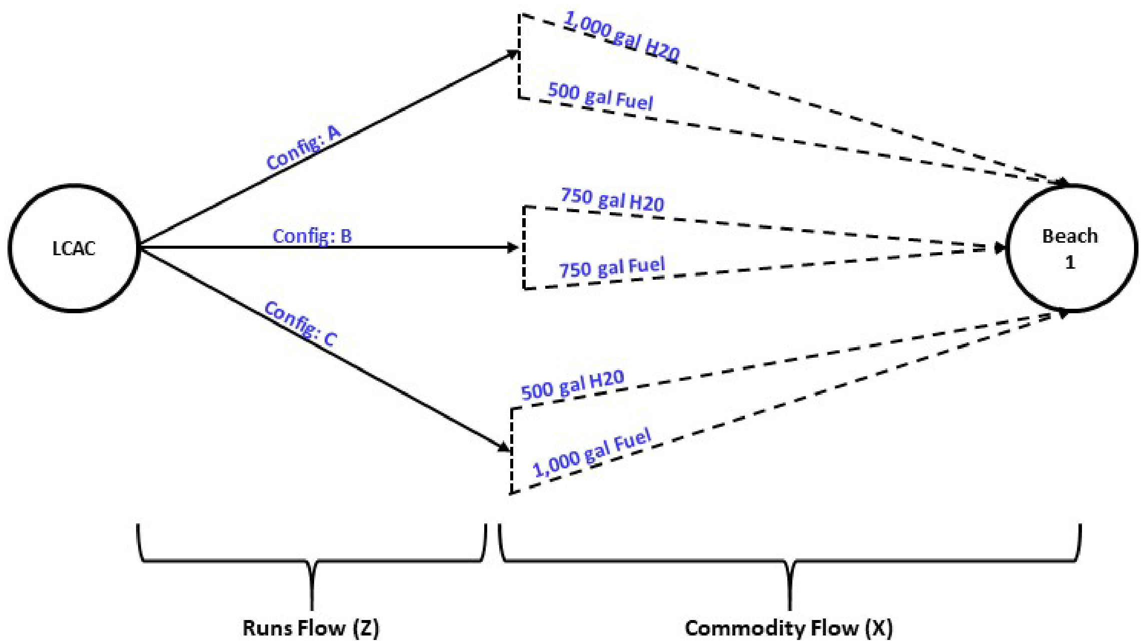



Figure 3. Example of Configurations Delivering Multiple Commodities

We define flow rate constraints for the variables $X_{i,j,t,c}$ and $Z_{s,l,g,t}$ using the parameters $u_l$ and $v_s$, respectively. The parameter $u_l$ is similar to the parameter used in the original MACS formulation, in that it places an upper bound on the amount of supplies that can flow via CLPs between land nodes at any given time. $v_s$ performs a similar function; however, this is a new parameter we introduce in our model to place an upper bound on the maximum number of connector runs of type $s$ that can depart from the seabase at any given time.

Two accounting variables are used to monitor the amount of runs available and the storage capacity at each land node. $Y_{n,t,c}$ remains largely unchanged from the original model and reflects the quantity of supply type $c$ stored at each land node at any given time. Lastly, the decision variable $W_{s,t}$ is introduced to capture the total number of possible runs remaining for each connector platform $s$ at time $t$. This value is initialized using $maxRuns_s$

and is updated at each time step, subtracting the number of newly initialized runs from the previous remainder.

We now present a detailed mathematical formulation of the Quickest Flow Model linear program, along with a description of each constraint.

**Indices and Sets**

| | |
|---|---|
| $t \in T$ | time periods |
| $s \in S$ | supply nodes |
| $l \in L$ | land nodes |
| $c \in C$ | commodity types |
| $g \in Gs_s$ | configurations of connector *s* |
| $ss \in superSink$ | single $ss \in superSink$ |
| $allSinks$ | all sink nodes: $allSinks = L \cup superSink$ |
| $i, j, n \in N$ | all nodes: $N = S \cup allSinks$ |
| $(i, j) \in A$ | arcs directed from node *i* to node $j \in N$ |

**Data [units]**

| | |
|---|---|
| $maxTime$ | total time available for operations [minutes] |
| $u_l$ | capacity on CLP flow leaving land node *l* for each time period [runs] |
| $v_s$ | capacity on flow of connecters *s* for each time period [runs] |
| $tau_{i,j}$ | time to travel on each arc $(i, j)$ [minutes] |
| $nodeCap_{l,c}$ | storage capacity at node *l* of commodity *c* [commodity units] |
| $maxRuns_s$ | total runs available per connector *s* [runs] |
| $demand_{l,c}$ | demand of commodity *c* at node *l* [commodity units] |
| $config_{s,g,c}$ | configuration *g* of connector $s \in Gs_s$ for commodity *c* [commodity units] |
| $f_c$ | capacity of commodity $c$ on convoy [commodity units] |
| α | weight to emphasize early delivery [unitless] |
| β | penalty value to minimize unused connector space [penalty units/run] |
| $wt_c$ | weight value to normalize the flow of all supplies [commodity units] |

**Decision Variables [units]**

| | |
|---|---|
| $X_{i,j,t,c}$ | flow of commodity *c* from node *i* to *j* at time *t* [commodity units] |
| $Y_{n,t,c}$ | amount of commodity *c* stored at node *n* at time *t* [commodity units] |
| $Z_{s,l,g,t}$ | flow of connector *s* to node *l* of configuration *g* at time *t* [runs] |

$W_{s,t}$ remaining runs of connector $s$ at time $t$ [runs]

**Formulation**

$$\max_{X,Y,Z,W} \quad z = \sum_{l \in L} \sum_{t \in T} \sum_{\substack{(l,ss) \in A, \\ ss \in SS}} \sum_{c \in C} wt_c \alpha^t X_{i,ss,t,c} - \beta \sum_{s \in S} \sum_{l \in L} \sum_{g \in Gs_s} \sum_{t \in T} Z_{s,l,g,t} \tag{3.1}$$

$s.t.$

$$W_{s,0} = maxRuns_s \qquad \forall s \in S \tag{3.2}$$

$$Y_{l,0,c} = 0 \qquad \forall l \in allSinks, \forall c \in C \tag{3.3}$$

$$Y_{l,t,c} \le nodeCap_{l,c} \qquad \forall l \in L, \forall t \in T, \forall c \in C \tag{3.4}$$

$$\sum_{\substack{l \in L, \\ g \in Gs_s}} Z_{s,l,g,t} \le v_s \qquad \forall\, s \in S, \forall t \in T \tag{3.5}$$

$$\sum_{t \in T} \sum_{\substack{ss \in SS, \\ (l,ss) \in A}} X_{l,ss,t,c} \le demand_{l,c} \qquad \forall l \in L, \forall c \in C \tag{3.6}$$

$$Y_{n,t+1,c} = Y_{n,t,c} + \sum_{\substack{i \in N, \\ (i,n) \in A}} X_{i,n,t-tau_{i,j},c} - \sum_{\substack{j \in N, \\ (n,j) \in A}} X_{n,j,t,c} \qquad \forall c \in C, \forall t \in \{T / \max(T)\}, \forall n \in N \tag{3.7}$$

$$W_{s,t+1} = W_{s,t} - \sum_{\substack{l \in L, \\ g \in Gs_s}} Z_{s,l,g,t} \qquad \forall s \in S, \forall t \in \{T / \max(T)\} \tag{3.8}$$

$$\sum_{\substack{j \in N, \\ (l,j) \in A}} X_{i,j,\max(T),c} \le Y_{l,\max(T),c} + \sum_{\substack{i \in N, \\ (i,l) \in A}} X_{i,l,\max(T)-tau_{i,j},c} \qquad \forall l \in L, \forall c \in C \tag{3.9}$$

$$\sum_{l \in L} \sum_{g \in Gs_s} Z_{s,l,g,t} \le W_{s,t} \qquad \forall s \in S, \forall t \in T \tag{3.10}$$

$$\sum_{t \in T} \sum_{s \in S} \sum_{\substack{j \in N, \\ (s,j) \in A}} X_{s,j,t,c} \le \sum_{t \in T} \sum_{l \in L} \sum_{\substack{ss \in SS, \\ (l,ss) \in A}} X_{l,ss,t,c} \qquad \forall c \in C \tag{3.11}$$

$$X_{s,l,t,c} \le \sum_{g \in Gs_s} Z_{s,l,g,t} config_{s,g,c} \qquad \forall s \in S, \forall t \in T, \forall l \in L, \forall c \in C \tag{3.12}$$

$$\sum_{\substack{j \in L, \\ (l,j) \in A}} X_{l,j,t,c} \le u_l f_c \qquad \forall l \in L, \forall t \in T, \forall c \in C \tag{3.13}$$

$$X_{i,j,t,c} \ge 0 \qquad \forall i \in N, \forall j \in N, \forall t \in T, \forall c \in C \tag{3.14}$$

$$Y_{n,t,c} \ge 0 \qquad \forall n \in N, \forall t \in T, \forall c \in C \tag{3.15}$$

$$Z_{s,l,g,t} \ge 0 \qquad \forall s \in S, \forall l \in L, \forall g \in Gs, \forall t \in T \tag{3.16}$$

$$W_{s,t} \ge 0 \qquad \forall s \in S, \forall t \in T \tag{3.17}$$

We now discuss each of the constraints in detail before elaborating further on the objective function.

1. Constraint (3.2) ensures that at time zero the number of available runs for each connector type equals the initial starting value $maxRuns_s$.

2. Constraint (3.3) ensures that there are no supplies on land at time 0.

3. Constraint (3.4) ensures that total supplies stored at any land node do not exceed the land storage capacity.

4. Constraint (3.5) limits the total number of runs initiated for each connector type across all configurations in each period.

5. Constraint (3.6) ensures that the sum of all the supplies consumed at any node does not exceed the amount demanded.

6. Constraint (3.7) is the classic flow balance equation ensuring that supplies being stored at each land node in a time period do not exceed the previous period's level plus the supplies flowing into the node minus the supplies flowing out of the node.

7. Constraint (3.8) updates the remaining available runs for all connectors by subtracting all the runs initialized in the previous time period.

8. Constraint (3.9) is the flow balance condition, similar to constraint (3.7), for the last time period.

9. Constraint (3.10) prevents the total number of runs initiated in a period over all configurations of one connector type from exceeding the max number of runs available.

10. Constraint (3.11) ensures that we only send supplies ashore that are consumed, preventing the delivery of unneeded supplies.

11. Constraint (3.12) converts run flow into commodity flow, which is the issue we discuss in Section III.A.1. The equation is not an equality as a connector

can depart for a shore run only partially loaded. This is discussed in detail below as we describe the objective function.

12. Constraint (3.13) restricts the amount of supplies flowing between any land nodes from exceeding the commodity capacity of a land connecter.

13. Constraint (3.14-17) ensure that all our decision variables are non-negative.

The objective function (3.1) has two terms, and is substantially different from the one in the original MACS model. With the inclusion of multiple supply commodities and several different connector platforms, we wish to not only maximize the delivery of supplies ashore, but also to find the most efficient means of doing so. Our primary objective is to maximize the flow of demanded commodities ashore. However, the units of our flow variables $X$ are commodity dependent (e.g., gallons vs. people vs. pallets) so we cannot directly sum our $X$ variables across commodity types. Thus, we include a weight parameter $wt_c$ to normalize all commodities. Our default value of $wt_c$ is $\frac{1}{\sum_{l \in L} demand_{l,c}}$. Thus, the maximum value of the objective function is $|C|$: the number of commodity types. If one wants to put a higher priority on certainty commodities, then the weight $wt_c$ could also capture that prioritization. Furthermore, in the case of priorities, we multiply $X_{i,ss,t,c}$ by $\alpha^t$, where $\alpha$ is slightly less than one (our default is 0.999), as suggested by the work completed by Strickland [15]. This modification encourages earlier deliveries.

The second term in the objective function is a penalty function $-\beta \sum_{s \in S} \sum_{l \in L} \sum_{g \in Gs_s} \sum_{t \in T} Z_{s,l,g,t}$, which tries to minimize the number of connector runs. The only constraint connecting the flow of runs to the flow of commodities is constraint (3.12). However, constraint (3.12) is an inequality because it might not be possible for all runs of all connectors to be loaded 100%. That is, constraint (3.12) allows for partially loaded runs. While partially loaded runs may be necessary to deliver all the supplies ashore, from a practical point of view we do not want many partially loaded runs. As our goal is to not only produce a schedule of amphibious deliveries ashore, but to do so as efficiently as possible, we want the connectors traveling to shore to have their cargo storage as full as possible. Without the penalty term in the objective function, constraint 3.12 may produce many more partially loaded runs than necessary. To address this issue, we introduce a penalty term for the total number of runs by using penalty weight $\beta$. In practice we could vary $\beta$ by connector type *s* or configuration *g* to skew the results to more heavily use certain connectors. This penalty term results in more efficient solutions with more fully loaded runs. In Chapter IV we perform sensitivity analysis on $\beta$ and examine how the schedule changes as we increase the penalty.

### 3. Output

The output of the Quickest Flow Model linear program provides the total number of runs required by each connector configuration to satisfy demand, and the total run count $\sum_{t \in T} Z_{s,l,g,t} \; \forall s \in S, \forall g \in Gs_s, \forall l \in L$. An example of this output appears in Figure 4. Of note, the Quickest Flow Model does not return integer values for the number of runs needed to satisfy demand. We need to perform an additional step to convert the runs to integer solutions. A conservative approach used by Christafore [3] rounds up the number of runs. We discuss another option in Section III.A.4 and go into much greater detail about rounding the runs in Appendix B.

```
Total CH53 runs: 6.784
  CH53 to MAIN requires 0.0 runs for config 1
  CH53 to MAIN requires 4.547 runs for config 2
  CH53 to MAIN requires 0.0 runs for config 3
  CH53 to LZ2 requires 0.0 runs for config 1
  CH53 to LZ2 requires 0.658 runs for config 2
  CH53 to LZ2 requires 0.0 runs for config 3
  CH53 to LZ1 requires 0.0 runs for config 1
  CH53 to LZ1 requires 1.579 runs for config 2
  CH53 to LZ1 requires 0.0 runs for config 3
```

```
Total MV22 runs: 12.05
  MV22 to MAIN requires 3.333 runs for config 1
  MV22 to MAIN requires 0.208 runs for config 2
  MV22 to MAIN requires 4.223 runs for config 3
  MV22 to LZ2 requires 1.905 runs for config 1
  MV22 to LZ2 requires 0.0 runs for config 2
  MV22 to LZ2 requires 0.0 runs for config 3
  MV22 to LZ1 requires 2.381 runs for config 1
  MV22 to LZ1 requires 0.0 runs for config 2
  MV22 to LZ1 requires 0.0 runs for config 3
```

Figure 4. Example of Connector Run Output

The second output of this model captures the total percentage of demand satisfied for each supply commodity. Furthermore, we perform a postprocessing step to determine the time when the total amount of supplies delivered for each commodity reaches 90 percent of demand. This is done to provide the user greater insight as to when they can expect the majority of their supplies. If this condition is not satisfied a message appears in the output indicating that less than 90 percent of required supplies could be delivered. The amount of supplies delivered ashore for each commodity is further broken down to reflect the amount being transported to each location and by what means. An example of this output is provided in Figure 5.

```
Total Amount of Fuel Delivered: 52 Thousand Gallons
Time to reach 90% for Fuel is: 90 minutes
  Fuel Demand Satisifed: 100.0 %
  Ship to Shore Deliveries:
    LCAC to B1 = 20.0
    LCAC to B2 = 15.0
    MV22 to MAIN = 7.0
    MV22 to LZ2 = 4.0
    MV22 to LZ1 = 5.0
  Land to Land Deliveries:
    MAIN to FWD = 6


Total Amount of Water Delivered: 35 Thousand Gallons
Time to reach 90% for Water is: 90 minutes
  Water Demand Satisifed: 100.0 %
  Ship to Shore Deliveries:
    LCAC to B1 = 22.5
    LCAC to B2 = 12.0
  Land to Land Deliveries:
    MAIN to FWD = 2
    B1 to MAIN = 3
    B1 to LZ1 = 5
    B2 to LZ2 = 2
```

Figure 5. Example of Supply Delivery Output

### 4. Mixed Integer Linear Program

The main output of Quickest Flow Model is the total run count during the day $\sum_{t \in T} Z_{s,l,g,t} \, \forall s \in S, \forall g \in Gs_s, \forall l \in L$. That is the number of runs of configuration g of connector type *s* to land node *l*. Unfortunately, as Figure 5 illustrates, the run count produced is likely non-integer. In Appendix B we examine an approach to convert the run count to integers. Another strategy is to build in the integrality constraint directly into the Quickest Flow Model. This requires the addition of a new decision variable and constraint to convert the model to a mixed integer linear program:

**Decision Variable**

$\hat{Z}_{s,g,l}$ Integer number of runs of configuration *g* of connector *s* to node *l* over the course of the entire operation.

**Constraint**

$$\sum_{t \in T} Z_{s,l,g,t} = \hat{Z}_{s,g,l} \qquad \forall s \in S, \forall g \in Gs_s, \forall l \in L \qquad (3.18)$$

Constraint (3.18) ensures that the total number of runs for each of a platform's configurations are equal to an integer value established by $\hat{Z}_{s,g,l}$.

While including decision variable $\hat{Z}_{s,g,l}$ and constraint (3.18) is arguably the "best" way to generate integer runs, there are potential computational issues with this mixed integer version of the model. If the mixed integer linear program is used for a complicated scenario involving large numbers of platform configurations or intricate amphibious networks, the run time may become exceptionally large and make this an unrealistic tool for planners. Therefore, we believe it is important to consider the non-integer version presented in Section III.A.2 on its own so we can generate schedules for scenarios where the mixed integer version fails to produce a solution in a reasonable amount of time. In Appendix B, we discuss rounding techniques for the non-integer model and compare the results to the mixed integer model.

## B. ASSIGNMENT HEURISTIC

As discussed in Section III.A, the main output of the Quickest Flow Model is the total number of connector runs by configuration that are required to satisfy shore-based demand. An example of this output is provided in Table 1 reflecting MV-22 resources required to meet demand at two LZs. This example includes three MV-22 configurations: J, K, and L.

Table 1. Example of MV-22 Resources Needed to Meet Demand

| LZ X | | LZ Y | |
|---|---|---|---|
| **Configuration** | **Number of Runs** | **Configuration** | **Number of Runs** |
| **J** | **3** | **J** | **2** |
| **K** | **2** | **K** | **1** |
| **L** | **1** | **L** | **0** |

The information in Table 1 is extremely useful. It identifies the number of runs for each platform variant needed to accomplish the mission. However, Table 1 is far from a complete schedule. The next step is to sort the runs in Table 1 to efficiently meet demand. For example, at LZ *X*, do we want all the configuration J connectors to arrive first, then K, then L? Should we intersperse deliveries of the three configurations? Furthermore, the connectors may originate from several different ships in the seabase, and thus we need to sequence when the connectors are dispatched from each ship. The Assignment Heuristic prioritizes the runs and generates a sequence of the runs from each ship to each land node.

Our first goal is to sort the configurations in Table 1, so we have a desired order for how we want the configurations to arrive to each land node. We start by prioritizing the commodities. For example, if the most pressing need of the GCE is ammunition, then the MV-22 runs presented in Table 1 should be organized in such a manner that the delivery of ammunition is prioritized above other commodities. We allow the user to specify priorities by connector type. For example, we may prioritize fuel on LCACs and people on CH-53s. More advanced schemes could also vary the prioritization by land node. Once we have the commodity priority for the connector type of interest, we sort the list of configurations in descending order of the capacity of the selected prioritized commodity. In the final step, we sort all the connector runs of the specified connector type to each of their delivery locations based on the configuration priority. An example of this process is presented in Figure 6, where the MV-22 runs presented in Table 1 are organized based on meeting ammunition requirements first. In this case configuration K has highest priority because it carries the largest amount of ammunition, followed by configuration L, then configuration J. The final output is a sorted list, based on commodity priority, of runs for each connector platform to each delivery location. Our desire is for the connectors to arrive to the two LZs in configuration order given by Figure 6.

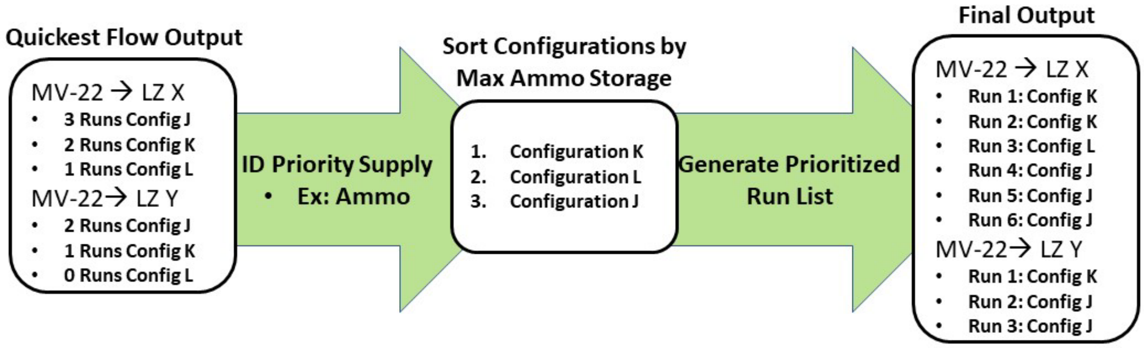



Figure 6. Example of MV-22 Runs Being Sorted by Priority

Now that we have a sorted list of runs by each configuration to each land node, we need to assign a connector run leaving the seabase to each of the nine LZ runs in Figure 6. This portion of the heuristic takes the form of a naïve planner. Using the output from Figure 6 a planner must determine the most efficient means of launching MV-22s from each of the ships to meet the required land node delivery sequence. Each ship within the ARG as well as the beaches, LZs, and FOBs have a limited number of landing locations (number of connectors that can arrive simultaneously based on a node's dimensional capacity) that a connector can conduct onload/offload. For example, if two LCACs are scheduled to deliver supplies to a beach with only one LCAC landing location, we need to determine which LCAC delivers first. If two landing spots are available on ship B and one on ship A, we need to determine where a returning MV-22s and CH-53 are sent to. Should runs be scheduled in a manner that reduce loiter time for a landing spot to become available, or do we schedule them in such a way that the demand with the largest deficit is addressed first? As the above questions address, we are dealing with a complicated assignment problem once all the connectors and different ARG ships are taken into consideration. An illustration of how the output from Figure 6 could be mapped to ship runs is depicted in Figure 7. The right side of Figure 7 corresponds to the right side of Figure 6.

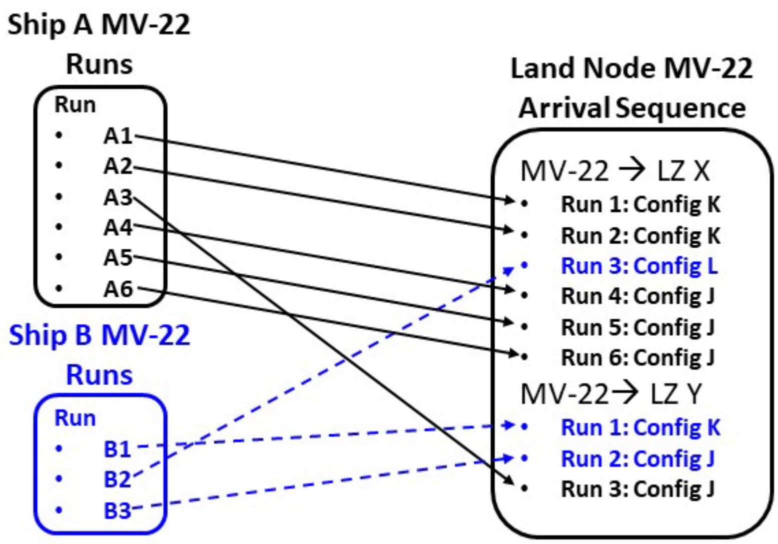



Figure 7. Example of Generating Ship Runs

One could develop a binary integer linear program to solve this matching problem to assign ship runs (left side of Figure 7) to LZ runs (right side of Figure 7). Christafore [3] presents a binary integer linear program, but it could take more than an entire day to solve even for a relatively small example. As we need a reliable tool that runs quickly, we opt for heuristics over a binary integer program to solve our assignment problem.

Another issue with the assignment matching is that while we have the land node run sequence list (right side of Figure 7), we do not have a ship run sequence list (left side of Figure 7). The example in Figure 7 assumes six connectors leave ship A and three leave ship B, but another schedule could have five leave A and four leave B. This complexity makes an integer program even more impractical. The next step in the heuristic determines the total number of runs and configuration by ship (left side of Figure 6) and provides the matching between ship runs and beach runs (the arrows in Figure 6).

The overall approach the heuristic takes is to create a discrete event simulation to mimic run scheduling operations in real time. The heuristic operates as though it is a naïve planner with the land node run sequence list (right side of Figure 7) at its disposal. Whenever an aircraft is ready for tasking aboard a ship the heuristic references the list, determines which land node is next on the list for scheduling, and then assigns the configuration and destination for the aircraft. The only decision the planner (heuristic) makes is which land node to assign the next available connector. Once a land node is chosen, the configuration follows immediately from the land node run sequence list (right side of Figure 7). There are a variety of policies a planner can use to determine which beach to send the next connector to. These policies include scheduling based on shortest distance, minimizing loiter time, and reducing relative deficit. The default policy is the reduction of relative deficit. These assignment operations continue with runs being scheduled in a myopic fashion until all land sequences have been satisfied. The final output is the final ship list and ship run sequence list similar to Figure 7.

The Assignment Heuristics remains largely unchanged from Christafore's work except for the update of two parameters, $loadTime_{s,g}$ and $unloadTime_{s,g}$. These two parameters address the specific loading and unloading time associated for each connector *s* and their respective configuration *g*. This is included to reflect the different time cost associated with various commodities being loaded on a connector and the time to transition between configurations. For example, an LCAC loaded with just food will take longer to load/offload when compared to one transporting vehicles, as vehicles can immediately debark while the food must be offloaded a pallet at a time using a forklift.

We believe the best way to describe this portion of the assignment heuristic is to walk through an example, which we provide in Appendix C. This appendix also contains the pseudocode for the entire Assignment Heuristic.

## C. SCHEDULER LINEAR PROGRAM

The final model of our MACS-MC tool is the Scheduler Linear Program. This model uses the output from the Assignment Heuristic in addition to user entered input parameters to create our detailed schedule of ship to shore deliveries. The Scheduler Linear Program is constructed as an optimization model that seeks to minimize the completion time of the schedule. The finalized output is a minute-by-minute schedule of each run. This model is largely similar to Christafore's work. The only difference is the load times and unload times now depend upon the configuration. We refer discussion to Appendix B as most of this is a replication.

The model tracks six events with each run:

1. Begin loading at the ship

2. Depart ship for beach or LZ

3. Arrive at Beach or LZ

4. Start unloading at beach or LZ

5. Depart beach or LZ for ship

6. Arrive back at ship

Most of the constraints for this model ensure the six events flow in a logical and consistent fashion. For example, a connector cannot leave the ship until it finishes loading; a connector cannot arrive until it finishes its transit; and a connector cannot begin unloading at a beach until a spot is available. If a connector arrives at a beach or landing zone and no spots are available it must loiter as it waits, which puts the connector in an operationally vulnerable position. This is tied to the Assignment Heuristic schedule that takes a naïve approach to scheduling our amphibious connectors. The Scheduler Linear Program formulation allows the user to control how much loitering occurs at beaches and LZs. If the user does not want loitering, the schedules are staggered to ensure a spot is open when the connector arrives, and no loitering occurs when the final schedule is produced. An example of the finalized schedule reflecting both configurations and the times for each of the six events is depicted in Table 2.

Table 2. CH-53 Example Produced by the Scheduler Linear Program

| | | | **Time Steps** | | | | | |
|---|---|---|---|---|---|---|---|---|
| **Run** | **Destination** | **Configuration** | **Start Load** | **Depart Ship** | **Arrive at LZ** | **Start Unload** | **Depart LZ** | **Arrive at Ship** |
| **1** | LZ2 | A | 0 | 20 | 50 | 50 | 75 | 105 |
| **2** | LZ1 | C | 0 | 45 | 75 | 75 | 100 | 130 |
| **3** | LZ2 | A | 105 | 125 | 155 | 155 | 180 | 210 |
| **4** | LZ2 | B | 130 | 150 | 180 | 180 | 205 | 235 |
| **5** | LZ2 | B | 210 | 230 | 292 | 292 | 317 | 380 |
| **6** | LZ1 | A | 235 | 255 | 317 | 317 | 342 | 405 |
| **7** | LZ1 | C | 380 | 400 | 462 | 462 | 487 | 550 |

For more information regarding the Scheduler Linear Program please see Appendix A, which includes the formulation for the model and an explanation for each of the constraints.

# IV. TRADEOFF BETWEEN COMMODITY DELIVERY AND CONNECTOR USAGE

The objective function in (3.1) has two terms. The first term tries to maximize the timely delivery of commodities, and the second term tries to minimize the number of connector runs. We use a parameter beta to regulate the relative importance between these conflicting objectives. This section studies this tradeoff numerically.

Our baseline Quickest Flow Model formulation includes the integer constraint in Chapter III.A.4 to ensure the number of runs is an integer. If the mixed integer linear program takes too long to solve, then we can use the linear program from Chapter III.A.2 and implement a rounding method to generate integer runs. In the interest of space, we defer discussion of rounding non-integer solutions to Appendix B.

## A. SCENARIO

For the remainder of this chapter we use the scenario in Figure 8. The seabase consists of one ARG with three ships LHD, LPD, and LSD. The air connectors have access to all land nodes except for the forward FOB we refer to as FWD; LCACs have access to both beaches; and the LCU is limited to just Beach 2. CLPs exist at both beaches and the Main to form our land network, with LZ1 being accessible only via air.

The requirements ashore consist of the following commodities:

- Fuel (in thousands of gallons)
- Water (in thousands of gallons)
- Ammunition (in pallets)
- PAX (in personnel)
- HMMWVs (in vehicles)

Demand at each of the nodes is represented in Table 3.

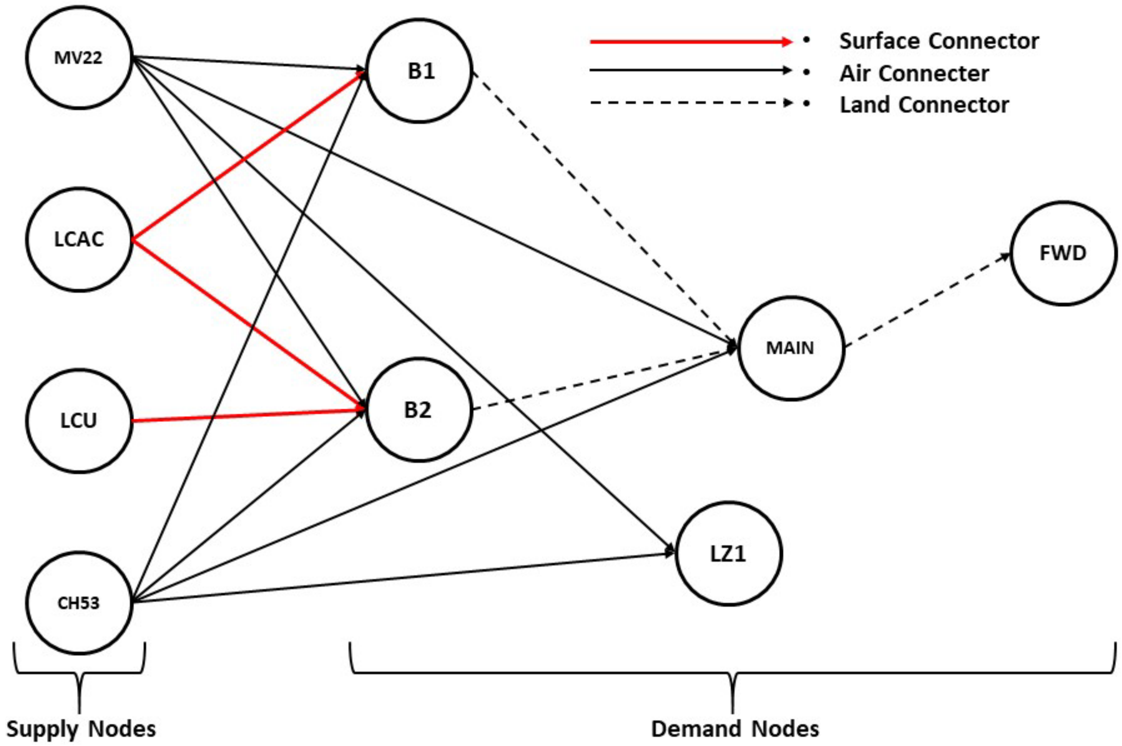



Figure 8. Example Amphibious Network

Table 3. Demand at Each Land Node

| | FUEL | WATER | AMMO | PAX | HMMWV |
|---|---|---|---|---|---|
| **B1** | 15 | 10 | 20 | 200 | 8 |
| **B2** | 2 | 3 | 6 | 50 | 4 |
| **LZ1** | 0.2 | 0.4 | 4 | 140 | 2 |
| **MAIN** | 3 | 3 | 12 | 80 | 10 |
| **FWD** | 3 | 2 | 8 | 160 | 4 |

There are 26 different configurations across all our amphibious connectors (six per surface connector and seven per air connector) as well as single one for the CLP to transport these commodities, as enumerated in Table 4.

Table 4. Configurations Utilized

| | Config | Fuel (gal) | Water (gal) | Ammo (pallet) | PAX (people) | HMMWV (vehicle) | loadTime | unloadTime |
|---|---|---|---|---|---|---|---|---|
| **MV22** | T | 0.5 | 0 | 0 | 0 | 0 | 15 | 5 |
| | U | 0 | 0 | 0 | 24 | 0 | 5 | 5 |
| | V | 0 | 0 | 0 | 0 | 1 | 19 | 40 |
| | W | 0 | 0 | 2 | 12 | 0 | 30 | 18 |
| | X | 0 | 0.5 | 0 | 0 | 0 | 15 | 5 |
| | Y | 0 | 0.1 | 1 | 12 | 0 | 20 | 10 |
| | Z | 0 | 0.2 | 2 | 0 | 0 | 15 | 20 |
| **CH53** | A | 0.5 | 0 | 0 | 0 | 0 | 15 | 5 |
| | B | 0 | 0 | 0 | 38 | 0 | 34 | 35 |
| | C | 0 | 0 | 4 | 18 | 0 | 18 | 38 |
| | D | 0 | 0 | 0 | 0 | 1 | 28 | 15 |
| | E | 0 | 0.5 | 0 | 0 | 0 | 15 | 5 |
| | F | 0 | 0.4 | 4 | 0 | 0 | 25 | 15 |
| | G | 0 | 0.4 | 0 | 18 | 0 | 30 | 10 |
| **LCAC** | A | 1.5 | 0.75 | 2 | 25 | 3 | 23 | 37 |
| | B | 0.75 | 1.5 | 3 | 25 | 3 | 31 | 27 |
| | C | 0 | 0 | 0 | 125 | 0 | 20 | 5 |
| | D | 0 | 0 | 0 | 25 | 11 | 40 | 15 |
| | E | 2 | 0 | 0 | 25 | 4 | 30 | 15 |
| | F | 0 | 2 | 10 | 25 | 2 | 40 | 20 |
| **LCU** | T | 0.4 | 0.2 | 3 | 50 | 8 | 44 | 35 |
| | U | 0.2 | 0.4 | 3 | 50 | 8 | 44 | 20 |
| | V | 0 | 0 | 0 | 300 | 0 | 15 | 5 |
| | W | 2 | 0 | 0 | 100 | 4 | 35 | 20 |
| | X | 0 | 2 | 20 | 50 | 4 | 35 | 40 |
| | Y | 0.8 | 0.8 | 8 | 100 | 6 | 30 | 35 |
| **CLP** | A | 2 | 2 | 10 | 20 | 4 | 150 | 40 |

The information in Table 4 represents how the user would input the different configurations for a mission. For example, MV-22 configuration Y transports 100 gallons of water, one pallet of ammunition, and 12 personnel. Furthermore, this configuration requires 20 minutes for loading at the seabase and 10 minutes to offload when it arrives at its destination. When we run this scenario with the mixed integer linear program variant of the Quickest Flow Model without a penalty in the objective function we generate the output depicted in Table 5 for MV-22s.

Table 5. MV-22 Output with no Objective Function Penalty

| Destination | Configuration | Required Runs |
|---|---|---|
| **B1** | **T** | 23 |
| | **W** | 8 |
| | **X** | 14 |
| **B2** | **X** | 1 |
| **MAIN** | **T** | 5 |
| | **V** | 114 |
| | **X** | 4 |
| **LZ1** | **T** | 1 |
| | **U** | 5 |
| | **V** | 2 |
| | **W** | 2 |
| **Total Required Runs:** | | **179** |

This output illustrates the issue from the introduction of this chapter: a large number of MV-22 runs needed to support the sustainment plan. Throughout this chapter as we modify our baseline model, we compare our updated results to the baseline output in Table 5 to highlight the effects our techniques have on the Quickest Flow Model output. Of note, configurations that do not require any runs have been omitted for ease of analysis. For example, Table 5 implies that the schedule requires zero configuration Z runs across all four land nodes. In this Chapter we examine how to satisfy the same amount of demand with fewer assets.

Of note, the analysis in this chapter relates only to the Quickest Flow Model delivery plan output, and not the final minute-by-minute schedule.

## B. OBJECTIVE FUNCTION PENALTY

In Sections IV.A and IV.B, we use Quickest Flow Model mixed integer linear program variant that produces integer runs. In our initial analysis, the objective function for the Quickest Flow Model in Chapter III.A.2 did not have the penalty term and took the form of

$$\max_{X,Y,Z,W} \sum_{l\in L} \sum_{t\in T} \sum_{\substack{(l,ss)\in A,\\ ss\in SS}} \sum_{c\in C} wt_c \alpha^t X_{i,ss,t,c} \; .$$

Without any penalty in the objective function, this formulation seeks to deliver supplies as quickly as possible without regard for economical connector use. It is with this formulation that we achieve the results in Table 5. From a planning perspective we also want to reduce the number of runs, but the only connection between the commodities being delivered ashore and the run count is constraint (3.12) from the Quickest Flow Model. This constraint is an inequality because if every run of every connector is packed full, the connectors could deliver more supplies ashore than are consumed. The inequality in constraint (3.12) allows us to have partial runs so that we deliver the precise amount of supplies ashore needed. Unfortunately, the inequality may also produce more partial runs than we want from an operational point of view. To counter these excess runs and improve configuration selection, we impose a penalty within the objective function for each connector run. This penalty takes the following form

$$-\beta \sum_{s \in S} \sum_{l \in L} \sum_{g \in Gs_s} \sum_{t \in T} Z_{s,l,g,t}.$$

If there are two solutions that generate the same delivery and consumption plan then this penalty term returns the plan with fewer runs as the optimal. Our penalty term has a constant parameter $\beta$. A more complex model would have $\beta$ depend upon connector type (*s*), delivery node (*l*), or configuration (*g*). This would provide the user with the option of biasing solutions to favor either fewer or more runs for specific connectors or runs to certain land nodes.

The penalty value is determined by the parameter $\beta$, which is a non-negative real number. As we vary $\beta$, the final delivery schedule can change significantly. Setting $\beta$ equal to a very small number does not change the final demand satisfied but improves packing efficiency. As we increase $\beta$, eventually we decrease the amount of delivered supplies, which reduces the amount of demand satisfied. This occurs because the penalty is too large to justify the additional deliveries. In the extreme for a very large $\beta$, there would be nothing delivered because of the large cost. The most interesting observations occur when we tune $\beta$ around small values where the amount of demand satisfied remains at its maximum. This can result in very different schedules and connector usages as we shall see later in this chapter.

The penalty term treats runs across the different connector types equivalently. Therefore, as we increase $\beta$ the updated solutions favor connectors that have greater supply transport capacity as this most effectively reduce the run count. This can have significant real-world implications as surface connectors carry many times the capacity of air connectors. The consequences of increasing $\beta$ include relying too heavily surface connectors, increasing delivery times, and placing a substantial burden on land connectors to push supplies forward.

In this thesis we fix $\beta$ to arbitrarily small values. However, further research could examine a more robust way to define $\beta$ for a given set of inputs. In practice we cannot just say $\beta$ =0.1 or 0.01 or 0.001 is a "good" choice for all problems, because the penalty term relates to commodities delivered and total run count, which are unknown ahead of time.

Continuing with our scenario, we analyze the effects of increasing $\beta$ by first determining when the amount of commodities consumed at any node changes. When the amount consumed (equivalently demand satisfied) changes, this implies that the penalty is adversely impacting delivery. We examine the *demand deficit* at each node and capturing when it changes. For this specific analysis we define demand deficit as the fraction of demand not satisfied for each commodity at each of the land nodes, or more specifically as

$$\frac{demand_{l,c} - \sum_{\substack{ss \in SS, \\ t \in T}} X_{l,ss,t,c}}{demand_{l,c}} \forall c \in C, \forall l \in L.$$

With this established, we run the Quickest Flow Model and vary $\beta$ from zero to 0.03 to measure any incurred demand deficit. The results in Table 6 indicate that when $\beta$ =0 the Quickest Flow Model satisfies all demand. As $\beta$ is incrementally increased to 0.02, the Quickest Flow Model returns plans that incur no deficit. Not until $\beta$ is increased to 0.03 does demand deficit accrue for certain commodities at different land nodes. For the remainder of this analysis we consider a range from zero to 0.01.

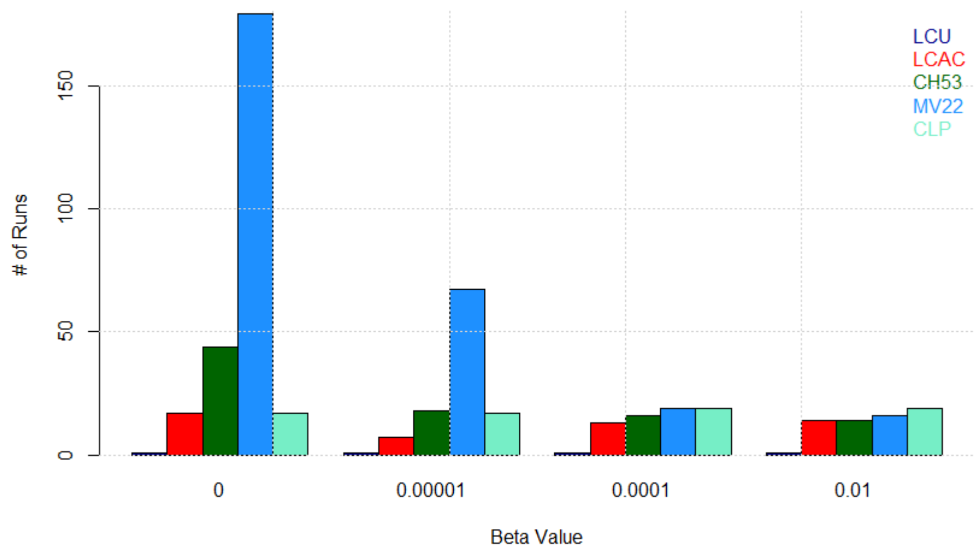
Table 6. Effect of Increased $\beta$ on Supply Deliveries

| | | **β=0** | **β=0.00001** | **β=0.0001** | **β=0.01** | **β=0.03** |
|---|---|---|---|---|---|---|
| Destination | Commodity | Deficit % | Deficit % | Deficit % | Deficit % | Deficit % |
| **B1** | Water | 0.00% | 0.00% | 0.00% | 0.00% | 0.00% |
| | Fuel | 0.00% | 0.00% | 0.00% | 0.00% | 0.00% |
| | AMMO | 0.00% | 0.00% | 0.00% | 0.00% | 0.00% |
| | PAX | 0.00% | 0.00% | 0.00% | 0.00% | 0.00% |
| | HMMWV | 0.00% | 0.00% | 0.00% | 0.00% | 0.00% |
| **B2** | Water | 0.00% | 0.00% | 0.00% | 0.00% | 0.00% |
| | Fuel | 0.00% | 0.00% | 0.00% | 0.00% | 0.00% |
| | AMMO | 0.00% | 0.00% | 0.00% | 0.00% | 0.00% |
| | PAX | 0.00% | 0.00% | 0.00% | 0.00% | 0.00% |
| | HMMWV | 0.00% | 0.00% | 0.00% | 0.00% | 0.00% |
| **MAIN** | Water | 0.00% | 0.00% | 0.00% | 0.00% | 0.00% |
| | Fuel | 0.00% | 0.00% | 0.00% | 0.00% | 0.00% |
| | AMMO | 0.00% | 0.00% | 0.00% | 0.00% | 0.00% |
| | PAX | 0.00% | 0.00% | 0.00% | 0.00% | 0.00% |
| | HMMWV | 0.00% | 0.00% | 0.00% | 0.00% | 0.00% |
| **LZ1** | Water | 0.00% | 0.00% | 0.00% | 0.00% | 0.00% |
| | Fuel | 0.00% | 0.00% | 0.00% | 0.00% | 100.00% |
| | AMMO | 0.00% | 0.00% | 0.00% | 0.00% | 0.00% |
| | PAX | 0.00% | 0.00% | 0.00% | 0.00% | 0.00% |
| | HMMWV | 0.00% | 0.00% | 0.00% | 0.00% | 0.00% |
| **FWD** | Water | 0.00% | 0.00% | 0.00% | 0.00% | 0.00% |
| | Fuel | 0.00% | 0.00% | 0.00% | 0.00% | 100.00% |
| | AMMO | 0.00% | 0.00% | 0.00% | 0.00% | 0.00% |
| | PAX | 0.00% | 0.00% | 0.00% | 0.00% | 0.00% |
| | HMMWV | 0.00% | 0.00% | 0.00% | 0.00% | 0.00% |

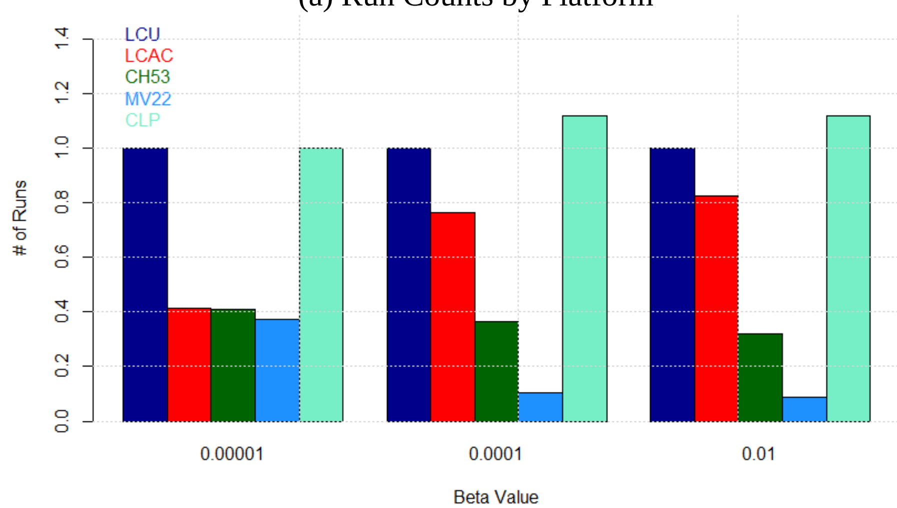
We also tabulate the number of runs generated as we vary $\beta$. We want the number of runs to be small. We use four $\beta$ values, (0, 0.00001, 0.0001, 0.01) and record the total run counts for each of the connectors and CLPs. The number of runs for surface and air connectors comes directly from the output to the Quickest Flow Model ($\sum_{l \in L} \sum_{g \in Gs_s} \sum_{t \in T} Z_{s,l,g,t} \; \forall s \in S$ ). To estimate the number of CLP runs between nodes we use

$$\max_c \left\lceil \frac{\sum_{(i,j) \in A} \sum_{t \in T} X_{i,j,t,c}}{f_c} \right\rceil \forall c \in C.$$

The numerator is the total amount of supplies of type *c* transported along the land network by CLPs. We divide that by the capacity of a CLP for commodity *c* to give an estimate of the number of CLPs to transport commodity c via land. The maximum over the commodity types provides the final estimate. Figure 9 provides the total run count across the different connector types for four values of $\beta$. Figure 9 (a) provides the total raw run counts. Figure 9 (b) provides the run counts relative to $\beta = 0$. A value of 1 in the Figure 9 (b) corresponds to the same number of runs in the $\beta = 0$ scenario.




(a) Run Counts by Platform




(b) Normalized Run Count

Figure 9. Run Counts and Normalized Runs. (a) reflects total number of absolute runs. (b) is relative to baseline $\beta = 0$, where one is equivalent to the baseline.

Setting $\beta$ =0.00001 reduces the number of connector runs by 148. There remains a heavy reliance on MV-22s (67 runs), and the total CLP count has remained the same at 17 indicating that the overall supply delivery plan remains largely unchanged. While the delivery plan remains unchanged, the decrease in runs by platform, represented by Figure 9 (b), translates to more efficient configurations being selected to meet demand.

For $\beta$ =0.0001, a more dramatic reduction in air connector runs occurs while the surface connector runs increase. MV-22 and CH-53 dropped to 10% and 40% of the original run values (Figure 9 (b)) suggesting that much of their delivery requirements has been assumed by the additional six LCAC runs. The dramatic reduction in the requirement for MV-22s and CH-53s occurs because, on average, a single LCAC's capacity is comparable to that of 10 MV-22s. This shift in focus to delivering supplies primarily through the two beaches places an increased burden on the ground network to push supplies forward to the outlying nodes. This has the added effect of increasing the time until demand is fully satisfied as supplies offloaded on the beach are reloaded and transported to their destination via CLPs, which when fully loaded have the approximate capacity of 15 MV-22s. This assumption is further supported by the CLP run count that increases by an additional two runs to 19. The increasing CLP run trend is an important aspect to keep track of as an amphibious planner must ensure that the land network does not become overburden with the requirement to push supplies inland.

Setting $\beta$ =0.01 results in the continued increase of surface connector runs at the expense of air connectors. LCUs and LCACs have used the maximum number of available runs (recall the constraint 3.12 from Chapter III, $\sum_{l \in L} \sum_{g \in Gs_s} \sum_{t \in T} Z_{s,l,g,t} = supply_s \quad s \in (LCAC, LCU)$ ) and CH-53's and MV-22's runs now represent 35% and 9% of their initial values. If the LCACs and LCUs did not reach their maximum allowable it is likely that we would have seen an even greater increase in surface usage. These results indicate that the surface connector runs are now fully loaded with supplies to compensate for the significant decrease in aircraft runs. CLPs remain at their highest run value, 19, an overall increase of two runs from the original results reflecting the extra supplies being delivered through the beaches.

Observing Table 7 we see the effects that increasing $\beta$ has on the delivery of supplies to each of the land nodes. The values in Table 7 represent the amount of supplies delivered directly from the seabase (across all four connector types) to the land node specified by the row. For the values considered in Table 7, the delivery plan satisfies all demand. Therefore, the columns sum to the total demand: 18.4 water, 23.2 fuel, 630 PAX (see Table 3). In each of the three commodities seen in Table 7, we see that there is only a slight change in the amount of supplies delivered to each of the nodes when we move from $\beta$=0 to $\beta$=0.0001. This occurs because when we set $\beta$ equal to a very small number, we do not change the delivery plan much, but just improve our use of connectors. However, when we transition to $\beta$=0.01 there is an increase in deliveries of fuel and PAX at the two beach nodes and a drop at the MAIN, which can only be accessed from the seabase by air connectors This highlights that more supplies are delivered through the two beaches and then pushed further inland via the use of CLPs.

Table 7. Effects of $\beta$ Value on Supply Delivered Directly from the Seabase to a Land Node

| Water Delivery vs β Value | | | | |
|---|---|---|---|---|
| | **0** | **0.00001** | **0.0001** | **0.01** |
| **B1** | 10.1 | 10 | 10.5 | 9.8 |
| **B2** | 3.1 | 3.2 | 3.7 | 3.5 |
| **Main** | 4.8 | 4.8 | 3.9 | 4.7 |
| **LZ1** | 0.4 | 0.4 | 0.4 | 0.4 |

| Fuel Delivery vs β Value | | | | |
|---|---|---|---|---|
| | **0** | **0.00001** | **0.0001** | **0.01** |
| **B1** | 15.1 | 15.5 | 14.8 | 14.2 |
| **B2** | 2.5 | 2.5 | 4.2 | 5.2 |
| **Main** | 5.3 | 5 | 4 | 3.5 |
| **LZ1** | 0.2 | 0.2 | 0.2 | 0.2 |

| PAX Delivery vs β Value | | | | |
|---|---|---|---|---|
| | **0** | **0.00001** | **0.0001** | **0.01** |
| **B1** | 317 | 244 | 290 | 274 |
| **B2** | 70 | 145 | 127 | 143 |
| **Main** | 102 | 100 | 72 | 72 |
| **LZ1** | 140 | 140 | 140 | 140 |

We finish this section by examining the consumption time distribution for each commodity. Consumption occurs when demand is satisfide at each node. Specifically the amount of commodity $c$ consumed at time $t$ is $\sum_{(l,ss)\in A} X_{l,ss,t,c}\ \forall t \in T, \forall c \in C.$ The consumption time distribution appears in Figure 10 in boxplot form. The box represents the middle 50 percent of the distribution and the black bar represents the median. Figure 10 confirms our intuition regarding reliance on air vs surface connectors as we vary $\beta$. When $\beta$=0.0001 the commodity consumption times are nearly identical to the $\beta$=0 case, with slightly heavier tails in PAX and HMMWV. This illustrates that the $\beta$=0.0001 produces essentially the identical delivery and consumption plan as when $\beta$=0 with approximately 148 fewer connector runs. This is tied to more efficient use of connector configurations. Furthermore, we observe that when $\beta$=0.01 the overall consumption time is dramatically slowed across almost all commodities. This reinforces the notion that a heavier reliance on the surface connectors places an increased burden on the CLPs to push supplies inland at a slower pace.

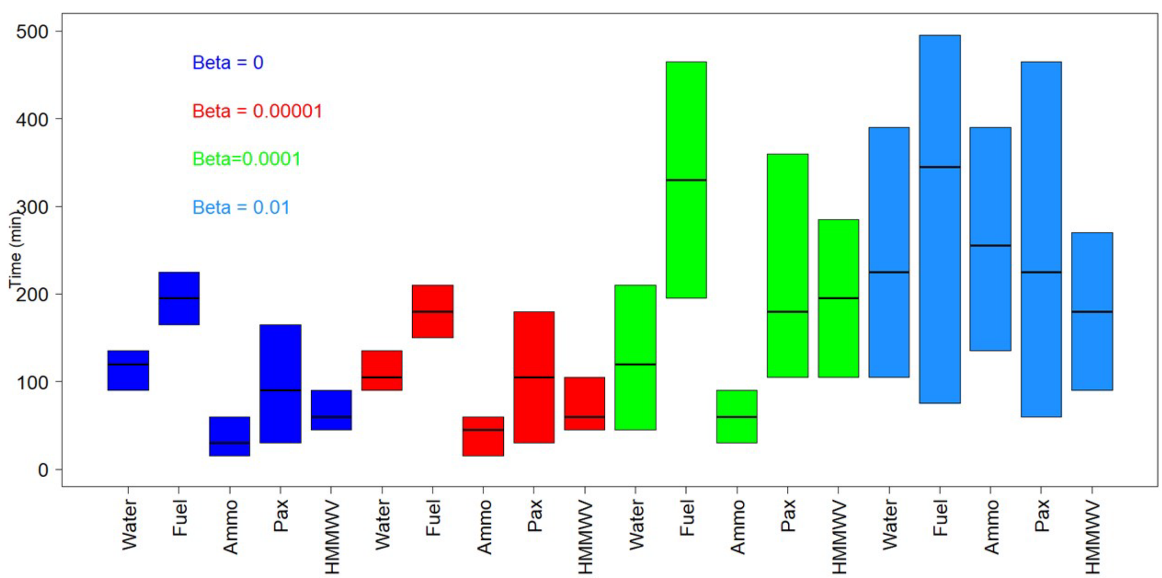



Figure 10. Effect of $\beta$ on Consumption Time by Commodity

We use $\beta$ =0.01 to run the Quickest Flow Model, and display the results for MV-22s in Table 8. Of note is the dramatic reduction in the overall run count when compared to Table 1 (dropping from 179 to 16 runs) and the concentration of runs into fewer configurations. Furthermore, 14 of the 16 runs are to non-beaches. This is a result of surface connectors increasing throughput at each of the two beaches, allowing MV-22s to focus deliveries to land nodes unreachable by LCACs and LCUs.

Table 8. Updated MV-22 Output for $\beta$ =0.01

| Destination | Configuration | Required Runs |
|---|---|---|
| **B1** | **X** | 1 |
| **B2** | **X** | 1 |
| **MAIN** | **T** | 7 |
| | **X** | 3 |
| **LZ1** | **U** | 2 |
| | **V** | 2 |
| **Total Required Runs:** | | **16** |

# V. A CASE STUDY

In this chapter, we apply the updated MACS-MC tool to a realistic scenario and discuss its output. This scenario, which we refer to as our base scenario, was designed to stress the MACS-MC tool by including many connector configurations and an amphibious network that is geographically separated by greater distances. We include this greater operational range to more accurately reflect real world operations where an ARG would operate from over the horizon (OTH). Similarly, we perform sensitivity analysis to assess how we might alter our initial scheme of maneuver (ARG positioning, composition, or timeline) to more effectively satisfy demand. All user defined inputs are read into the model using comma separated value (CSV) files.

Furthermore, we apply our model to a historical amphibious landing that took place during the Vietnam War. This analysis highlights the strengths of MACS-MC and reflects its accuracy and efficiency. In the interest of space, we to defer the specific details of this analysis to Appendix F.

## A. BASE SCENARIO OVERVIEW

In this section, we present the scenario and rationale behind the corresponding the input parameters.

### 1. Scenario Parameters

Our scenario is based on the Naval Postgraduate School's (NPS) Joint Campaign Analysis (OA4602, Professor Jeff Kline) mini-study project assigned to students in the Operations Research Department [16]. More detail regarding this project can be found in Appendix D. This scenario entails the amphibious assault of an embarked MEU deploying from a three ship ARG to seize an enemy held island. The island in question has a limited number of usable beaches and LZs, making the efficient delivery of sustainment ashore both difficult to accomplish and realistic.

The premise of our scenario is that an ARG had been dispatched to the western Pacific to conduct an amphibious assault on the island of Natuna-besar. We adjust our ARG composition from the OA4602 mini-study prompt to reflect our standard seabase ships (LPD, LHD, LSD), and we assume we have sufficient sustainment embarked to support 15 days of operations. At this point in time, the bulk of the MEU's combat power has been landed ashore and the priority for delivering commodities ashore has transitioned to sustainment operations except for a final tank and light armored vehicle (LAV) company. The amphibious network consists of three beaches and three LZs as outlined in Figures 11 and 12. Of note Figure 11 only contains the connections from the seabase to land. For a detailed overview of the land network see Figure 12.

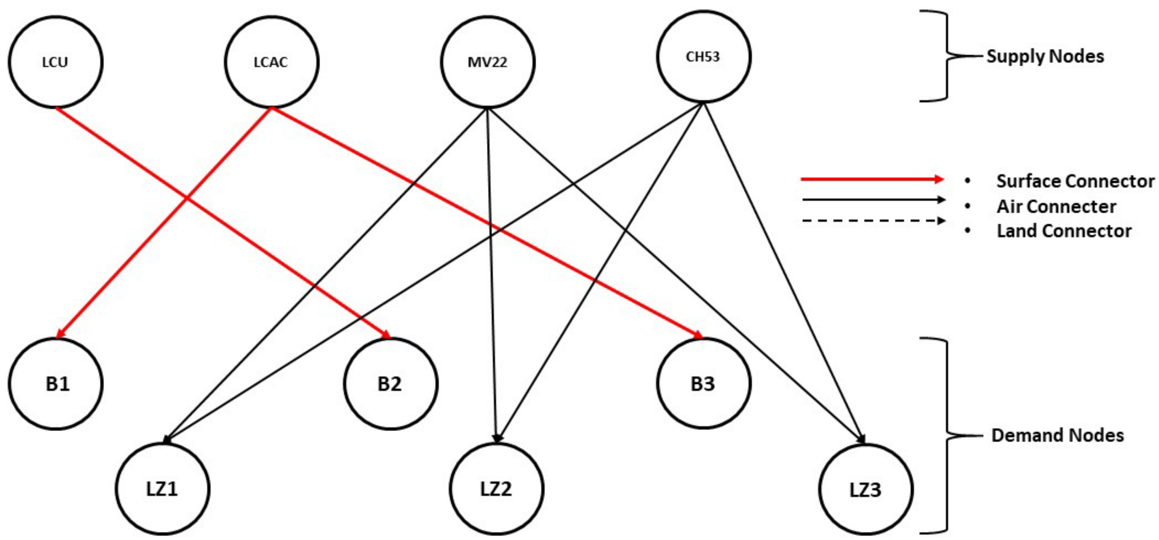



Figure 11. Base Scenario Amphibious Network

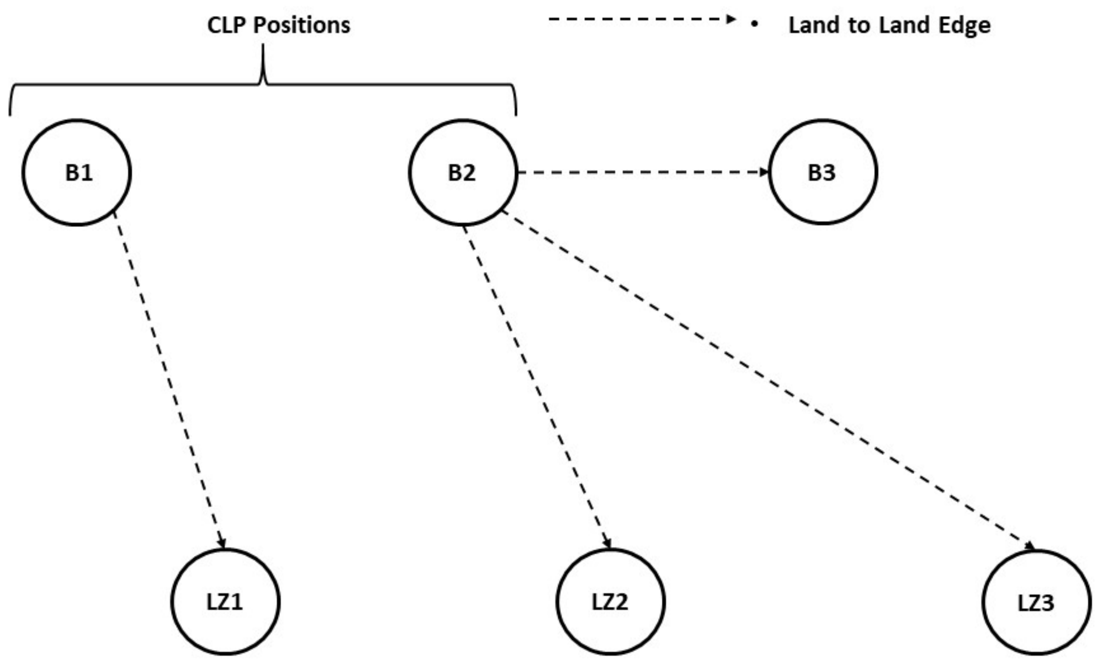



Figure 12. Base Scenario Land Network

LZs and beaches are located relatively close to one another as there is only one segment of the island suitable for landing with a thick jungle canopy preventing the deeper insertion of friendly forces. This prevents any nodes from being further inland as there were no suitable aircraft landing spots or any existing infrastructure to push forces inland via CLPs. The *(i,j)* edges represented in Figure 11 are tied to the information provided in Table 9, which define where each of our connectors are located. Of note, CLPs are positioned at B1 and B2 (Figure 12), as these two sites are the most beneficial for pushing supplies through the network and still far enough away from anticipated fighting (closer to LZ3) that they would not become vulnerable to attack. The CLP at B1 was only capable of delivering supplies to LZ1. B2’s CLP is connected to the remaining LZs and B3, allowing it to travel to the end of the amphibious network to deliver supplies as needed.

Table 9. Disposition of Connectors within the Network

| Connector Disposition (Surface, Air, & Land) | | | | | |
|---|---|---|---|---|---|
| **Node** | **LCAC** | **LCU** | **CH53** | **MV22** | **CLP** |
| **LHD** | 3 | 0 | 2 | 8 | 0 |
| **LPD** | 0 | 1 | 1 | 2 | 0 |
| **LSD** | 2 | 0 | 2 | 2 | 0 |
| **B1** | 0 | 0 | 0 | 0 | 1 |
| **B2** | 0 | 0 | 0 | 0 | 1 |

We position the three ships of our ARG relatively close together and OTH. The rationale for the ARGs placement is to keep all our shipping out of sight from enemy forces and thus harder to target. Generally speaking, something is OTH if it is greater than 21 nautical miles away, placing it out of sight of enemy forces [14]. These distances, as well as the distances between our land nodes are reflected in Table 10. It is important to point out that the distance between the land nodes (beaches and LZs) are not as the crow flies, but instead reflect the travel distance between each of the nodes on a local roadway. This is why the distances between land nodes was so precise, as it reflects the exact distance measured from a map. Additionally, the information reflects exact distances between all the land nodes; however, this does not mean that an edge exists in our model. Figure 12 provides the edges that are used in this scenario.

Table 10. Distances between Nodes (miles)

| | **LHD** | **LPD** | **LSD** | **B1** | **B2** | **B3** | **LZ1** | **LZ2** | **LZ3** |
|---|---|---|---|---|---|---|---|---|---|
| **LHD** | | 4 | 1 | 25 | 25 | 26 | 26 | 27 | 26 |
| **LPD** | 4 | | 5 | 29 | 28 | 26 | 30 | 27 | 25 |
| **LSD** | 5 | 1 | | 24 | 25 | 27 | 25 | 26 | 28 |
| **B1** | 25 | 29 | 24 | | 2.54 | 8.31 | 1.16 | 4.94 | 9.49 |
| **B2** | 25 | 28 | 25 | 2.54 | | 5.77 | 1.38 | 2.4 | 6.95 |
| **B3** | 26 | 26 | 27 | 8.31 | 5.77 | | 7.15 | 3.37 | 1.18 |
| **LZ1** | 26 | 30 | 25 | 1.16 | 1.38 | 7.15 | | 3.78 | 8.33 |
| **LZ2** | 27 | 27 | 26 | 4.94 | 2.4 | 3.37 | 3.78 | | 4.55 |
| **LZ3** | 26 | 25 | 28 | 9.49 | 6.95 | 1.18 | 8.33 | 4.55 | |

This scenario requires the delivery of nine separate supply commodities outlined below. Of note there are two ammunition categories as these two types of ammunition have been deemed to be incompatible with one another, and therefore cannot travel concurrently. In practice this is performed to separate small arms ammunition with explosives.

1. Fuel (in thousands of gallons)
2. Water (in thousands of gallons)
3. PAX (in personnel)
4. HMMWV (in individual trucks)
5. AMMO1 (in pallets)
6. AMMO2 (in pallets)
7. Tank (in individual tanks)
8. LAV (in individual LAVs)

Table 11 contains the demand for each of these commodities broken down by location. Demand for fuel, water, and ammunition have been designed to reflect the consumption of the MEU forces operating ashore for a single day, in addition to surplus bulk fluids to establish a land-based stock pile. The requirement for ammunition reflects the need for stockpiled ammunition and does not reflect consumption. Personnel, tank, LAV, and HMMWV numbers all reflect the need to deliver remaining combat power and support personnel ashore. The numbers generated to represent the water and MRE demand reflect tropical conditions (each Marine consuming three gallons of water and three MREs per day). Additionally, tank and LAV numbers reflect the accurate composition of a Marine tank and LAV company. We expect combat to occur around B3 and LZ3; however, the demand for tanks and LAVs in Table 11 show the bulk of these assets coming ashore at B1 and B2. This was purposefully done to model these assets landing in a relatively safe area to stage and collect their combat power before departing to engage the enemy. Additionally, while an edge does not exist from B1 to LZ1 and the remainder of the land network, it is assumed that these assets, once fully formed, will move independently of any land connectors.

Table 11. Demand Required at each Node

| Node | FUEL | WATER | AMMO1 | AMMO2 | HMMWV | PAX | LAV | TANK | MRE |
|---|---|---|---|---|---|---|---|---|---|
| **B1** | 8 | 1.5 | 15 | 10 | 8 | 75 | 12 | 10 | 10 |
| **B2** | 3 | 1 | 10 | 4 | 2 | 30 | 6 | 4 | 6 |
| **B3** | 0.5 | 0.5 | 4 | 1 | 8 | 30 | 6 | 4 | 4 |
| **LZ1** | 0 | 1 | 4 | 0 | 2 | 100 | 0 | 0 | 3 |
| **LZ2** | 0 | 0.6 | 2 | 2 | 2 | 50 | 0 | 0 | 3 |
| **LZ3** | 0 | 0.6 | 2 | 0 | 4 | 24 | 0 | 0 | 1 |

### 2. General Parameters

Based on the daily consumption of certain commodities and the need to complete the combat power delivery of our forces ashore, we establish the following priorities for each of our connectors:

- LCU: Tank
- LCAC: LAV
- MV-22: MRE
- CH-53: Water

Tanks were chosen for LCU's priority because the LCU has the capability of delivering two tanks simultaneously while the LCAC can only deliver one. LCACs were assigned the priority of transporting LAVs as they cannot go ashore via air connector and with there being more LAVs to transport than tanks it was important to use the surface connector that has the most assets available. The air connectors were selected to prioritize water and MREs to facilitate the more rapid replenishment of daily consumables, which in real life can bring an operation to a halt if not efficiently satisfied. While assigning priorities to each of the connectors does not ensure they are delivered by that platform, it does speed their delivery if the model chooses to send them with that craft. For instance, tanks being a priority for LCU's does not guarantee that tanks will be delivered by LCUs. However, it does guarantee that if the Quickest Flow Model selects tanks to move via LCU then tanks will be delivered to shore before any other commodity.

For this scenario, we set $\beta = 0.000001$. We chose this value after conducting sensitivity analysis, similar to what we employed in Section IV.B.1, with the value being the largest allowable amount before any additional demand deficit began to accrue. This value also has a minimal impact on the overall run count compared to a situation with no penalty, which is beneficial as our scenario involves longer distances (requiring more runs).

The model used a 10-hour day length as this most accurately reflects the maximum continuous amount of time well deck and air operations can take place. We use a total of 108 different configurations: 21 MV-22 configurations, 21 CH-53 configurations, 20 LCAC configurations and 46 LCU configurations. We choose these configurations based on the author's experience in how supplies are commonly packaged and transported aboard each of the platforms. The LCU has a greater number of configurations as their additional cargo storage area allows for more permutations of the supply commodities than the LCAC. For a complete listing of all the parameters used in this scenario, please see Appendix E.

## B. BASE SCENARIO IMPLEMENTATION AND RESULTS

We use the Quickest Flow Model mixed integer linear program as our configuration count is relatively low and the network is not overly complicated. MACS-MC produced results after a total run time of 72 seconds when run from a Microsoft Surface Pro with an Intel Core i5 processor and 16GB of RAM. Since we use the mixed integer linear program version, we did not implement a rounding technique. This method produces the final run count depicted in Figure 13 and Table 12.

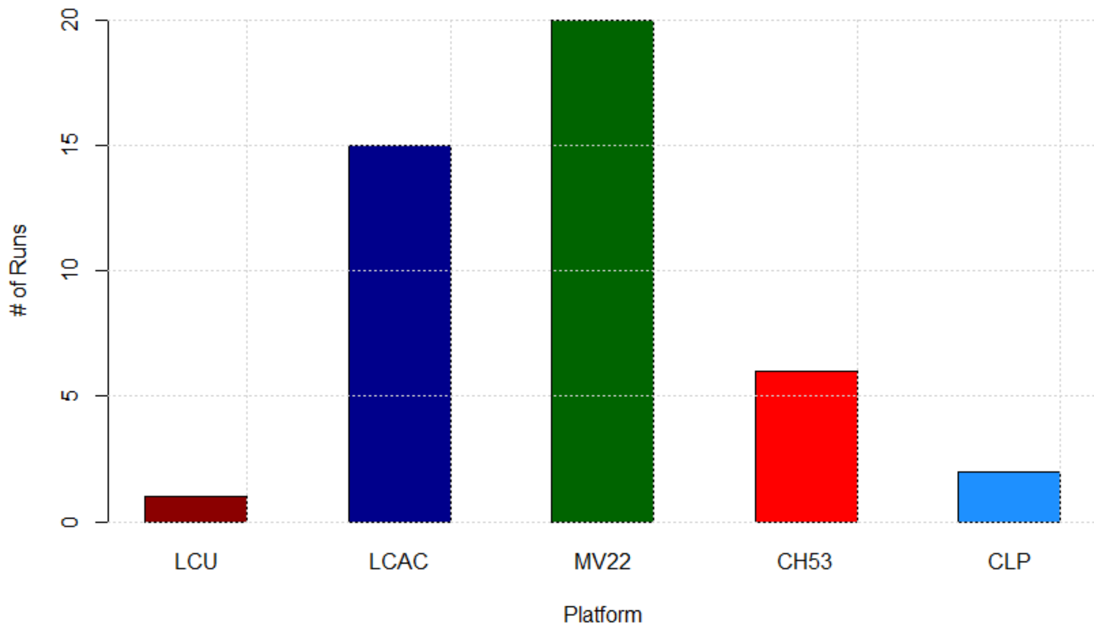



Figure 13. Base Scenario Run Count

Table 12. Runs by Connector to each Node

| | LCU | LCAC | MV-22 | CH-53 |
|---|---|---|---|---|
| **B1** | 0 | 12 | | |
| **B2** | 1 | 0 | | |
| **B3** | 0 | 3 | | |
| **LZ1** | | | 6 | 3 |
| **LZ2** | | | 6 | 3 |
| **LZ3** | | | 8 | 0 |

An important aspect to note is the relatively low run count for the entire scenario. This reflects our decision to position the MEU OTH in a defensive posture, which equates to longer travel distances and fewer round trips per connector. This assumption is confirmed when we compare the number of surface connector runs to their maximum possible number of runs. LCU and LCAC runs equal their maximum allowable value, demonstrating that the longer travel distances impose a restriction on the number of runs that can be used, and in turn limits deliveries. For example, the time requirement for a single LCU roundtrip was approximately 450 minutes. With a 600-minute upper bound on our operational window we can only have one LCU run. Similarly, LCACs require approximately 150 minutes (including onload and offload time) to complete a round trip, translating to three runs per LCAC or 15 total.

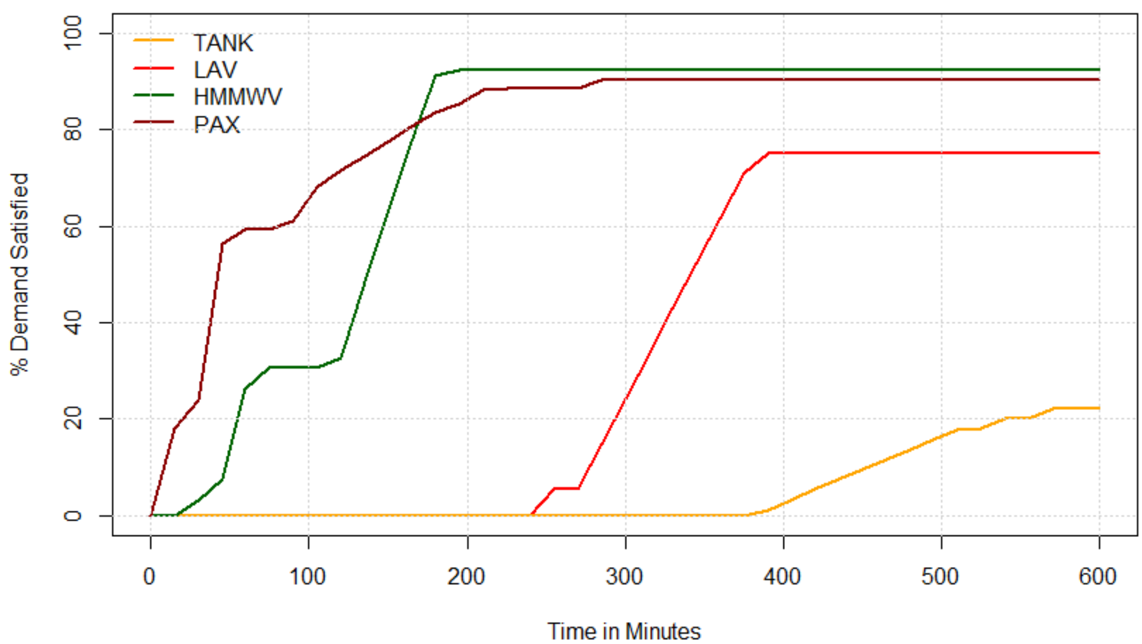
Furthermore, we see a surprisingly small number of air platforms being used in relation to their total supply of runs (40 and 168 runs for CH-53s and MV-22s, respectively). This suggests that demand at the three LZs is satisfied relatively quickly when compared to the beach nodes. This argument is supported by the fact that only two CLP runs are dispatched, and this is done to deliver supplies from B1 to LZ1 and B2 to B3. These CLP runs consist of relatively small amounts of ammunition, MREs and PAX. If we had any existing deficit at the LZs we would expect to see more air connector runs or CLP runs from the beaches to each of the LZs. Translating these run counts into demand satisfied (see Table 13) provides more useful information on the efficiency of our delivery plan and confirms many of our initial assumptions.

Table 13. Demand Satisfied by Commodity and Node

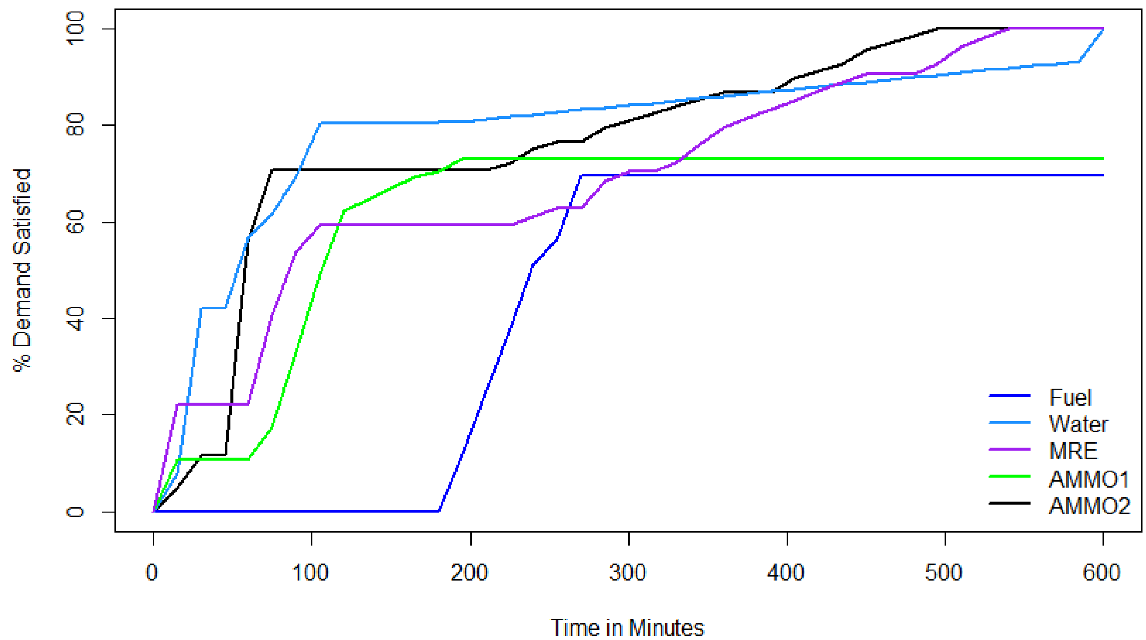
| Node | FUEL | WATER | AMMO1 | AMMO2 | HMMWV | PAX | LAV | TANK | MRE |
|---|---|---|---|---|---|---|---|---|---|
| **B1** | 100.00% | 100.00% | 100.00% | 100.00% | 100.00% | 100.00% | 100.00% | 50.00% | 100.00% |
| **B2** | 0.00% | 100.00% | 0.00% | 100.00% | 0.00% | 0.00% | 0.00% | 0.00% | 100.00% |
| **B3** | 0.00% | 100.00% | 100.00% | 100.00% | 100.00% | 100.00% | 100.00% | 0.00% | 100.00% |
| **LZ1** | | 100.00% | 100.00% | | 100.00% | 100.00% | | | 100.00% |
| **LZ2** | | 100.00% | 100.00% | 100.00% | 100.00% | 100.00% | | | 100.00% |
| **LZ3** | | 100.00% | 100.00% | | 100.00% | 100.00% | | | 100.00% |
| **Total:** | **69.57%** | **100.00%** | **72.97%** | **100.00%** | **92.31%** | **90.29%** | **75.00%** | **27.78%** | **100.00%** |

The deficit presented in Table 13 points to several interesting observations. First, the deficit is very asymmetric. Demand at the LZs is very easy to satisfy while it is much more difficult to meet beach requirements via surface connectors. This occurred because B2 is only accessible from the seabase by LCUs. With the ARG being placed OTH, this restricted the LCU to the one run, which is why we see such great deficits across most of the commodities at this node. Furthermore, there are no CLPs that could access B2, preventing supplies from being pushed from other locations.

Second, we observe that the hardest commodity to satisfy is tanks. There are a total of 14 tanks demanded at both B1 and B3 and we are limited to 15 LCAC runs. Only five LCAC runs are used to transport tanks because there are many other supplies that LCACs need to bring to those beaches. Similarly, with four tanks demanded at B2 and only one LCU run available, the solution elects to send one run fully loaded LCU with consumables to avoid incurring a larger deficit. These results help illustrate the difficulties faced when operating OTH. We present Figure 14 to demonstrate the relationship between time and the delivery of all our commodities.



(a) Combat Power Assets




(b) Consumable Supplies

Figure 14. Demand Satisfied per Commodity against Time

As we observe the relationship between time and percentage of demand delivered per commodity, we notice that the first commodities did not begin to reach 100% delivery completion until after 450 minutes. Figure 14 (a) demonstrate the relatively large transportation times associated with moving both tanks and LAVs via surface connectors and suggests the significant impact that operating OTH imposes on delivering large items. Consumable commodities shown in Figure 14 (b) reflect steadier delivery rates as they can move continuously via air connectors with limited turnaround time. However, when we look at the delivery rates for AMMO1 and fuel, we notice both commodities reach their maximum limit at 180 and 280 minutes, respectively, demonstrating that LZ demand has been satisfied while the beaches struggled to deliver tanks and LAVs. Both graphs however, point to the need of increasing the rate of delivery for all commodities to efficiently meet all the desired demand.

Should the senior decision maker not like the outcomes presented in Table 13, the following three options are possible ways to improve demand satisfaction. We examine these options in Section V.D:

1. Place the ARG closer to shore

2. Change the disposition of our land connectors

## C. FINAL SCHEDULE

Before performing the sensitivity analysis suggested at the end of Section V.B, we present the final schedule of deliveries for each of our connectors. We review the process from Chapter III for how the final schedule is produced. First the Quickest Flow Model produces the number of runs for each configuration of each connector to each land node. Table 12 presents a consolidated version of this run-count by aggregating over configurations. This run-count information is passed to the Assignment Heuristic (see Chapter III.B), where we associate each of the land node runs with a run originating from a ship in the ARG. This sequencing of connectors runs from ship to land sequences are passed through the Scheduler Linear Program (see Chapter III.C) to produce our final schedule for each run departing their respective ships. The final minute-by-minute schedule for each connector appears in this section. All times reflected in Table 14–17 are in minutes.

### 1. LCU Schedule

The ARG composition consists of one LCU. Given the delivery time constraints associated with making runs OTH, there is only sufficient time to make one LCU round trip. Additionally, LCUs are the only connector that can reach B2  With no CLPs available to deposit supplies at this node, this single run constitutes the only supplies sent to B2 throughout the entire scenario. What we see in Table 14 is that this single run, configuration QQ, consists entirely of consumables. Or more precisely, 1,000 gallons of water, 20 pallets of MREs, and 10 pallets of AMMO2. These are the only supplies that are 100% satisfied at B2 in Table 14. If the LCU were transporting only vehicles it would only be able to satisfy one type of demand (tanks or LAVs). These large vehicles are hard to pack, consume most of the LCUs cargo capacity, and make it difficult to transport any other commodities simultaneously.

Table 14. Final LCU Schedule

| Ship | Run | Config | Destination | Start Load | Depart Ship | Arrive at LZ | Start Unload | Depart LZ | Arrive at Ship |
|---|---|---|---|---|---|---|---|---|---|
| LPD | 1 | QQ | B2 | 0 | 25 | 235 | 235 | 250 | 460 |

### 2. LCAC Schedule

The final LCAC schedule in Table 15 reflects 15 different runs using six different configurations. All runs depart from the LHD and LSD with their final destinations being the two beaches accessible by this craft, B1 and B3. Of interest is the choice of configurations C, T and B that correspond to fully loaded deliveries of LAVs, HMMWVs, and tanks, respectively. The run departure for configuration C are front loaded in the schedules for both ships. This reflects the decision to have LAVs be the priority for LCACs, explaining why the delivery of tanks does not occur until later in the schedule. The remaining configurations (F, L & P) correspond to configurations loaded with consumables. These runs account for only five of the total of 15 runs. With only fifteen runs to satisfy demand across multiple commodities, the need to transport combat power cuts into the limited number of runs needed to deliver the remainder of the consumables ashore. However, these five runs of consumables are still effective at delivering all the required consumables to B1 and B3 (only nodes LCAC can access) except for fuel at B3 (see Table 13). So overall, while transporting combat power consumes a majority of the LCAC transportation capacity, the final schedule is still able to meet most demand, including five tanks and 18 LAVs (12 to B1 & six to B3).

Table 15. Final LCAC Schedule

| Ship | Run | Config | Destination | Start Load | Depart Ship | Arrive at LZ | Start Unload | Depart LZ | Arrive at Ship |
|---|---|---|---|---|---|---|---|---|---|
| LHD | 1 | C | B1 | 0 | 45 | 95 | 95 | 105 | 155 |
| LHD | 2 | T | B1 | 0 | 45 | 95 | 95 | 105 | 155 |
| LHD | 3 | P | B1 | 45 | 75 | 125 | 125 | 140 | 190 |
| LHD | 4 | F | B1 | 155 | 190 | 240 | 240 | 255 | 305 |
| LHD | 5 | B | B1 | 155 | 190 | 240 | 240 | 245 | 295 |
| LHD | 6 | B | B1 | 190 | 235 | 285 | 285 | 290 | 340 |
| LHD | 7 | C | B3 | 295 | 340 | 392 | 392 | 402 | 454 |
| LHD | 8 | T | B3 | 305 | 350 | 402 | 402 | 441 | 493 |
| LHD | 9 | B | B1 | 340 | 375 | 425 | 425 | 430 | 493 |
| | | | | | | | | | |
| LSD | 1 | C | B1 | 0 | 45 | 93 | 93 | 103 | 151 |
| LSD | 2 | P | B1 | 45 | 75 | 123 | 123 | 138 | 186 |
| LSD | 3 | L | B1 | 151 | 191 | 239 | 239 | 264 | 312 |
| LSD | 4 | B | B1 | 191 | 226 | 274 | 274 | 279 | 347 |
| LSD | 5 | B | B1 | 312 | 347 | 395 | 395 | 400 | 448 |
| LSD | 6 | L | B3 | 347 | 387 | 441 | 441 | 466 | 520 |

### 3. MV-22 Schedule

The final MV-22 schedule has 20 runs (Table 16) departing from all three ships and destined for all three LZs. There is an assortment of different configurations, seven total, to deliver supplies ashore. Most of the configurations deal with mixed loads of consumables and personnel. However, there are seven runs of configuration C, which deliver HMMWVs exclusively. These seven HMMWV runs account for 87.5% of LZ HMMWV requirements, with a CH-53 delivering the final vehicle. The MV-22 finish their final deliveries eight hours before the 10-hour operational window close. This is because all the demand at each of the LZs is satisfied. Furthermore, there are no ground movements, CLPs, departing from any of the LZs that could requiring more runs to be made.

Table 16. Final MV-22 Schedule

| Ship | Run | Config | Destination | Start Load | Depart Ship | Arrive at LZ | Start Unload | Depart LZ | Arrive at Ship |
|---|---|---|---|---|---|---|---|---|---|
| LHD | 1 | L | LZ3 | 0 | 10 | 18 | 18 | 23 | 31 |
| LHD | 2 | N | LZ1 | 0 | 10 | 18 | 18 | 23 | 31 |
| LHD | 3 | D | LZ3 | 0 | 20 | 27 | 27 | 42 | 70 |
| LHD | 4 | C | LZ3 | 0 | 34 | 42 | 42 | 52 | 70 |
| LHD | 5 | C | LZ3 | 24 | 44 | 52 | 52 | 72 | 80 |
| LHD | 6 | N | LZ1 | 10 | 44 | 52 | 52 | 57 | 65 |
| LHD | 7 | N | LZ2 | 39 | 44 | 53 | 53 | 58 | 66 |
| LHD | 8 | C | LZ1 | 34 | 54 | 62 | 62 | 72 | 80 |
| LHD | 9 | C | LZ3 | 44 | 64 | 72 | 72 | 82 | 106 |
| LHD | 10 | C | LZ3 | 54 | 74 | 82 | 82 | 93 | 111 |
| LHD | 11 | C | LZ2 | 65 | 85 | 93 | 93 | 98 | 106 |
| LHD | 12 | B | LZ1 | 70 | 85 | 93 | 93 | 98 | 106 |
| LHD | 13 | B | LZ3 | 70 | 85 | 93 | 93 | 98 | 111 |
| | | | | | | | | | |
| LPD | 1 | N | LZ1 | 4 | 9 | 18 | 18 | 23 | 32 |
| LPD | 2 | I | LZ2 | 4 | 44 | 53 | 53 | 93 | 101 |
| LPD | 3 | N | LZ2 | 32 | 44 | 53 | 53 | 58 | 101 |
| | | | | | | | | | |
| LSD | 1 | N | LZ3 | 0 | 5 | 13 | 13 | 18 | 26 |
| LSD | 2 | T | LZ2 | 0 | 20 | 27 | 27 | 42 | 50 |
| LSD | 3 | C | LZ1 | 26 | 55 | 62 | 62 | 72 | 80 |
| LSD | 4 | B | LZ2 | 50 | 85 | 85 | 93 | 98 | 106 |

### 4. CH-53 Schedule

The CH-53 schedule of six runs includes deliveries from all three ships using four different configurations (Table 17). The four different configurations represent the delivery of predominately personnel intermingled with small amounts of consumable commodities (water, fuel and MREs) and one HMMWV (configuration A). An interesting difference between this schedule and that for the MV-22s is the fewer number of runs. The CH-53 is a larger platform and thus requires more space to land. For example, an LZ that can accommodate a single CH-53 could fit two MV-22s, which is why we see a diminished run count for this platform. Similarly, the MV-22 is a much faster platform and can complete almost two round trip deliveries in the time it takes a CH-53 to complete one.

Table 17. Final CH-53 Schedule

| Ship | Run | Config | Destination | Start Load | Depart Ship | Arrive at LZ | Start Unload | Depart LZ | Arrive at Ship |
|---|---|---|---|---|---|---|---|---|---|
| LHD | 1 | F | LZ2 | 0 | 20 | 31 | 31 | 46 | 58 |
| LHD | 2 | O | LZ1 | 0 | 35 | 46 | 46 | 76 | 87 |
| LPD | 1 | F | LZ2 | 0 | 20 | 31 | 31 | 46 | 58 |
| LSD | 1 | A | LZ2 | 0 | 15 | 26 | 26 | 31 | 50 |
| LSD | 2 | R | LZ1 | 15 | 50 | 60 | 60 | 90 | 101 |
| LSD | 3 | F | LZ1 | 50 | 70 | 80 | 80 | 95 | 106 |

## D. SENSITIVITY ANALYSIS

At the end of Section V.B we propose three ways to potentially improve our delivery schedule. In this section we conduct sensitivity analysis to determine what change could improve demand satisfaction. This involves re-running the scenario three separate times and analyzing the total run count by connector and the demand satisfied by commodity and in totality. These adjustments are:

1. Placing the ARG 10 nautical miles closer to shore (35% reduction)

2. Adjusting the placement of the CLPs

### 1. Closer Placement of the ARG

By placing the ARG 10 nautical miles closer to shore, the ARG is roughly 35% percent closer, which substantially reduces the travel time for all connectors and increases surface throughput. These effects are apparent when we observe the total run count in Figure 15 and Table 18. LCU runs increase twofold while LCACs saw an increase of 10 runs (two per LCAC). The overall CLP count decreases from two to one. This single CLP run consists of one pallet of ammunition and four pallets of MREs being pushed from B2 to B3.

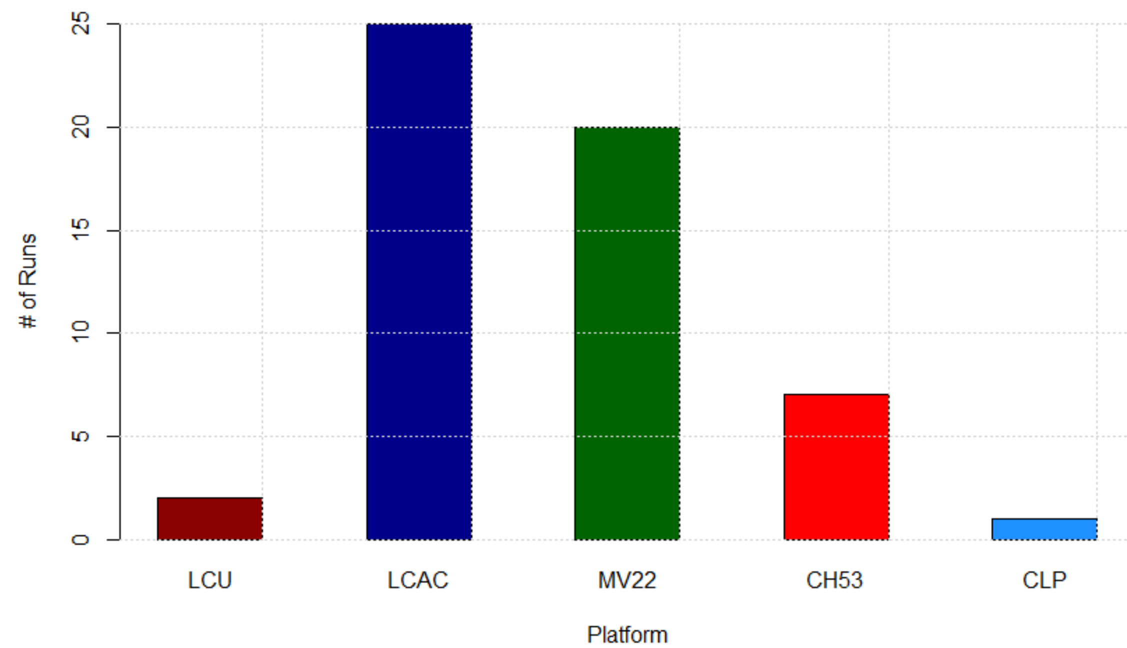



Figure 15. Final Run Count with the ARG Closer to Shore

Table 18. Runs by Connector to each Node

| | LCU | LCAC | MV-22 | CH-53 |
|---|---|---|---|---|
| **B1** | 0 | 17 | | |
| **B2** | 2 | 0 | | |
| **B3** | 0 | 8 | | |
| **LZ1** | | | 6 | 5 |
| **LZ2** | | | 6 | 2 |
| **LZ3** | | | 8 | 0 |

The effects of these extra runs are reflected in Table 19 with a greater percentage of the overall demand satisfied. Notable improvements include the delivery of tanks and LAVs, with 14 and 21 of each coming ashore. This is significant as it represents the bulk of both company's combat power, translating to their ability to begin combat operations. Despite the drop in CLP runs, the delivery plan still manages to satisfy demand at all the LZs as the use of air connectors making direct deliveries has increased by one platform. A notable change is the eight LCAC runs going to B3. With additional runs available, LCACs shift their delivery focus to delivering LAVs and tanks (one and four runs, respectively), leaving the three remaining runs to satisfy consumable demand.

Table 19. Delivery Satisfaction with the ARG Closer to Shore

| Node | FUEL | WATER | AMMO1 | AMMO2 | HMMWV | PAX | LAV | TANK | MRE |
|---|---|---|---|---|---|---|---|---|---|
| **B1** | 100.00% | 100.00% | 100.00% | 100.00% | 100.00% | 100.00% | 100.00% | 100.00% | 100.00% |
| **B2** | 0.00% | 100.00% | 100.00% | 100.00% | 0.00% | 0.00% | 50.00% | 0.00% | 100.00% |
| **B3** | 100.00% | 100.00% | 100.00% | 100.00% | 100.00% | 100.00% | 100.00% | 100.00% | 100.00% |
| **LZ1** | | 100.00% | 100.00% | | 100.00% | 100.00% | | | 100.00% |
| **LZ2** | | 100.00% | 100.00% | 100.00% | 100.00% | 100.00% | | | 100.00% |
| **LZ3** | | 100.00% | 100.00% | | 100.00% | 100.00% | | | 100.00% |
| | | | | | | | | | |
| **Total:** | **73.91%** | **100.00%** | **100.00%** | **100.00%** | **92.31%** | **90.29%** | **87.50%** | **77.78%** | **100.00%** |

### 2. Repositioning the CLPs

One of the trends we notice in the base scenario and in all the subsequent sensitivity analysis is the relatively low number of air connector and CLP runs. If we could use these assets more, surplus supplies could be delivered to the LZs and then transported overland to those nodes still requiring them. However, the position of the CLPs (at B1 and B2) is not well suited to meet this requirement as the CLPs can only push assets out from their assigned node. As such, we reposition our two CLPs to LZ1 and LZ2 as a means of addressing this shortcoming (Figure 16).

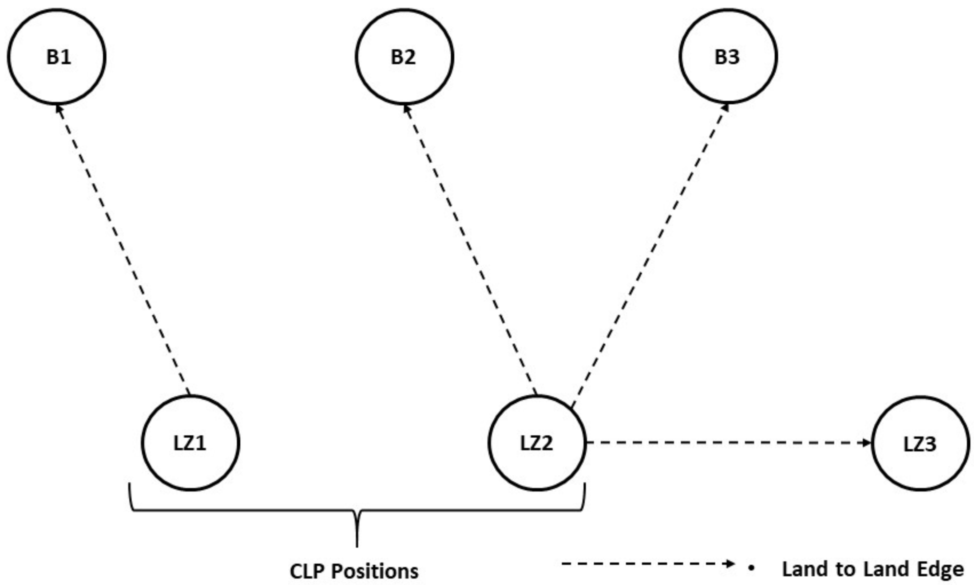



Figure 16. Land Network after Repositioning CLPs

The results of this repositioning, presented in Figure 17 and Table 20 demonstrate dramatic changes. The number of air connector runs increase sharply (65 MV-22 runs and 18 CH-53 runs) to deliver surplus supplies at LZ1 and LZ2. This translates to nine total CLP runs delivering supplies to B1, B2, and B3. Surface connector use remains high; however, an interesting change occurs with the LCU. Instead of trying to get as many commodities to B2 as possible to increase the cumulative supplies ashore, the only run now consists of an LCU transporting six LAVs. This is a direct reflection of consumable supplies being delivered to B2 via CLP from LZs instead of needing to be transported via surface connector.

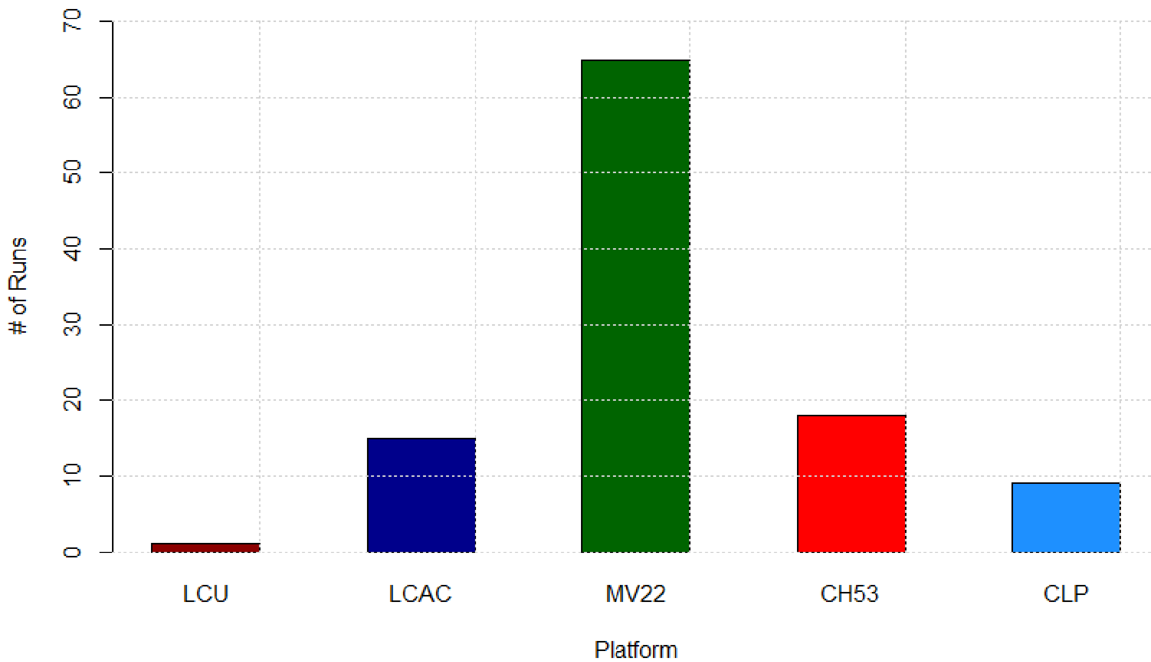



Figure 17. Final Run Count with New CLP Positioning

Table 20. Runs by Connector to each Node

| | LCU | LCAC | MV-22 | CH-53 |
|---|---|---|---|---|
| **B1** | 0 | 12 | | |
| **B2** | 1 | 0 | | |
| **B3** | 0 | 3 | | |
| **LZ1** | | | 34 | 11 |
| **LZ2** | | | 25 | 5 |
| **LZ3** | | | 6 | 2 |

The effect these additional air and CLP runs have on the delivery plan appear in Table 21. We see an almost complete reduction in all deficits for every commodity. Indeed, the only commodity still requiring delivery is tanks at both B2 and B3. As we have mentioned in previous analysis, this outlier in deliveries is tied to the lack of required surface connectors to deliver these large assets ashore. Tanks cannot be delivered via air. However, despite this shortcoming, allowing the CLPs to operate from LZ1 and LZ2 has dramatic improvements on the flow of supplies throughout the network. This availability allows greater quantities of supplies to be dropped off at the LZs to be then pushed to the beaches. This facilitates the surface connectors to concentrate more exclusively on the delivery of combat power.

Table 21. Delivery Satisfaction with New CLP Positioning

| Node | FUEL | WATER | AMMO1 | AMMO2 | HMMWV | PAX | LAV | TANK | MRE |
|---|---|---|---|---|---|---|---|---|---|
| **B1** | 100.00% | 100.00% | 100.00% | 100.00% | 100.00% | 100.00% | 100.00% | 100.00% | 100.00% |
| **B2** | 100.00% | 100.00% | 100.00% | 100.00% | 100.00% | 100.00% | 100.00% | 0.00% | 100.00% |
| **B3** | 100.00% | 100.00% | 100.00% | 100.00% | 100.00% | 100.00% | 100.00% | 50.00% | 100.00% |
| **LZ1** | | 100.00% | 100.00% | | 100.00% | 100.00% | | | 100.00% |
| **LZ2** | | 100.00% | 100.00% | 100.00% | 100.00% | 100.00% | | | 100.00% |
| **LZ3** | | 100.00% | 100.00% | | 100.00% | 100.00% | | | 100.00% |
| | | | | | | | | | |
| **Total:** | **100.00%** | **100.00%** | **100.00%** | **100.00%** | **100.00%** | **100.00%** | **100.00%** | **66.70%** | **100.00%** |

### 3. Final Observations

To compare the variants in Section V.C we normalize the delivery values for each trial so that we have a common metric across scenarios. We perform this normalization by summing together the total quantity delivered divided by the demand per commodity or mathematically as:

$$\sum_{c \in C} \sum_{\substack{l \in L, \\ ss \in SS, \\ t \in T}} \frac{X_{l,ss,t,c}}{demand_{l,c}}.$$

This method produces a number between zero and one for each commodity, reflecting percent of demand satisfied. When all these commodity values are summed together we achieve a number between zero and nine, with nine being the maximum possible value and indicative that all demand has been satisfied. A potential downfall of performing this normalization is that it does not include a weight for certain commodities should their delivery be more desired than others. Regardless, for our purposes this is still an effective means of communicating the best means of satisfying demand, as reflected in Figure 18.

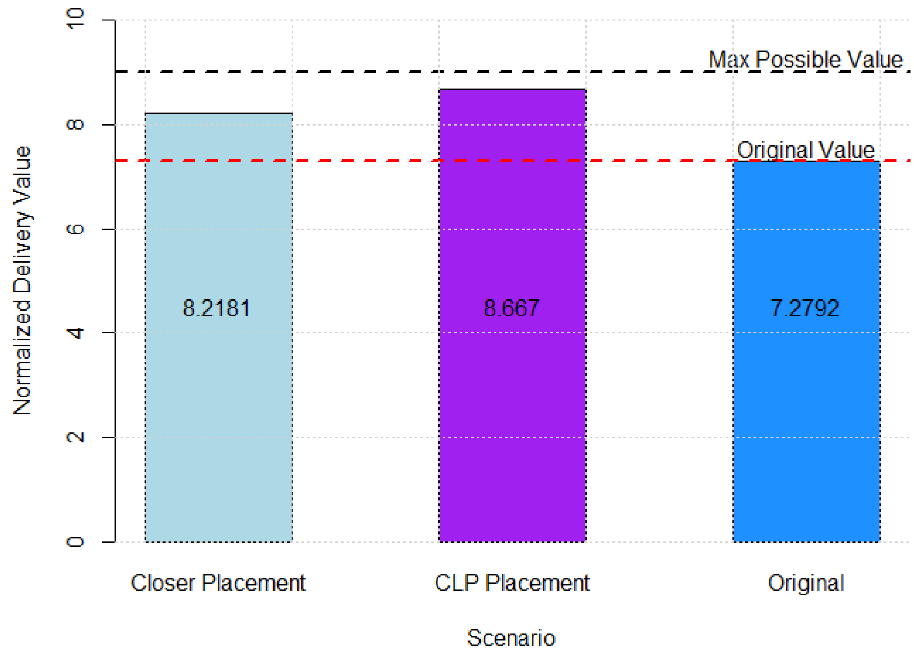



Figure 18. Normalized Delivery Values for each Method Tested

Analyzing the normalized delivery values for each of our methods, it becomes strikingly clear that the repositioning of our CLPs produces the best results. Repositioning of the ARG has improved the overall delivery of supplest and remains a viable alternative. Based on these results our final recommendation is to reposition the CLPs at LZ1 and LZ2 as this does not require the ARG to come any closer to shore and takes advantage of the ARGs ability to dispatch numerous aircraft runs.

# VI. CONCLUSIONS

This thesis improves the MACS in Christafore [3] to allow amphibious connectors to transport multiple commodities rather than just fuel. Improvements also include a new version of the Quickest Flow Model that is an mixed integer linear program and ideal for small amphibious networks or low configuration counts. This inclusion forces MACS-MC to return integer values for our runs by platform configuration whereas the original linear program would return a continuous flow for the same output. For larger amphibious networks, or when the number of configurations exceeds 200, we may need to revert to the linear program version of the Quickest Flow Model with one of several rounding mechanisms we created to convert the float run values to integer. Both the Assignment Heuristic and Scheduler Linear Program remain largely unchanged from their original model; however, small adjustments have been made to account for the incorporation of platform runs by configuration.

To improve connector efficiency, we develop the capability to generate multiple schedules by adjusting the objective function penalty. This feature allows a user to produce MACS-MC schedules that could favor faster deliveries, surface or air connectors, and the reliance to place on the ground network infrastructure. The flexibility this provides is extremely useful for amphibious planners as it allows them to adjust schedules to meet mission specific needs.

Our research demonstrates the versatility and efficiency MACS-MC can provide for sustainment operations by analyzing several notional scenarios. These demonstrations reflect the flexibility of our algorithms to accurately account for supply compatibility restrictions, vast numbers of configuration permutations, multiple supply commodities, and the ability to accurately produce a schedule of amphibious connector deliveries. Finally, we provided a comparison of our model output against an historical example (Appendix F) to demonstrate the improvements MACS-MC can provide to planners for the rapid and accurate development of a ship to shore landing schedule.

There are three areas of research that could enhance the accuracy of the MACS toolkit and make it a more reliable asset for planners.

1. Model Improvements

To date there remains no means of automating our value for the penalty parameter $\beta$, forcing the user to conduct sensitivity analysis to measure the effects of different penalty functions on the outcome. Additional research could codify a mechanism to use user inputs to automatically create a penalty function value to achieve the effects of either improving loading efficiency or shifting results to favor either air vs surface connector dominated delivery plans. These improvements would prevent the user from running MACS-MC multiple times to find their desired $\beta$ value.

More work should be conducted to improve the user interface. In its current form, MACS-MC requires the user to complete 10 separate CSV files and then manually adjust certain starting parameters in Python before running a scenario. Many of these actions could be streamlined into a single Microsoft Excel user interface. This interface could allow a user to input all information required for the generation of the 10 CSVs, place them in their folder, update the Python starting parameters, and then run the model in Python. This update would provide the user with a less cumbersome and time-consuming method of running any of their desired scenarios.

Another important contribution to the model would be the creation and incorporation of new rounding mechanisms. In its current form the MACS-MC has only two mechanisms to round non-integer run counts with the Quickest Flow Model linear program. The two approaches presented in this thesis (see Appendix B) may not provide the user with a wide enough range of options. Further research in this area could produce several additional rounding techniques that could more efficiently round run values to further minimize excess runs and reduce delivery deficit. Additionally, the incorporation of new techniques that can consider the users desire to round based on some preferred methodology (rounding by commodity, delivery nodes, delivery time, etc.) would generate greater versatility in the user's ability to run the MACS-MC tailored to their personal preferences.

Finally, the priorities given to each of our connectors should be changed from a single commodity to a prioritized list across all commodities. Currently the user defines one commodity per connector that, if the Quickest Flow has decided to transport that commodity with that connector, forces the scheduler to dispatch the configuration carrying that commodity before any others. As our results have demonstrated, once connectors have satisfied their priority commodity, or if they are not selected by Quickest Flow for transport, the scheduler begins to randomly dispatch connector configurations until all required runs have been satisfied. With a relatively small adjustment, this single priority can be converted to a prioritized list. This improvement would allow for each of the connectors to focus on subsequent tiers of priority commodities vice focusing on one exclusively.

### 2. Include AAVs for a Holistic Model

Both Christafore's [3] research and our work aim to develop the MACS-MC as a sustainment centric model. In this sense we assume that an assault force has already secured a beach and any subsequent deliveries involve the delivery of supplies, personnel, and equipment via means other than AAVs. However, with small adjustments to the models, the use of AAVs could be included as a one-way delivery of personnel to conduct the initial assault. This would produce a final output reflecting an entire amphibious operation through both the assault and sustainment phases. As we have already demonstrated, the MACS-MC tool is robust enough to handle any type of supply commodity and can easily model the delivery of all combat power assets required in the beginning of an operation.

### 3. Combine with Strickland's Research

Finally, the model could be made more robust if combined with the concurrent work that Strickland [15] has undertaken as a separate expansion of Christafore's [3] model. Strickland took the approach of making the MACS a more robust model by accounting for uncertainty in weather, enemy impact, and mechanical failure. These uncertainties are applied to every iteration of the MACS process that produces a final schedule that incorporates buffer runs to account for potential connector attrition, or slower delivery rates [15]. Furthermore, adjustments were made within the Quickest Flow Model that allowed certain land nodes to be assigned the priority for resupply as well as a mechanism to ensure that equity exists between node consumption. In its current form, Strickland's model can only transport one commodity, fuel, similar to Christafore's [3] approach. However, if both Strickland's model and the work presented in this thesis are combined, then MACS-MC would consist of a model that could simultaneously push multiple commodities through an amphibious network while building a robust delivery schedule accounting for potential delays and connector attrition.

# APPENDIX A. SCHEDULER LINEAR PROGRAM

The Scheduler Linear Program remains largely unchanged from Christafore's [3] original model (Chapter III.C). We present the entire formulation below but refer the reader to Christafore's [3] for the full details. The only aspect that differs between Christafore's original formulation and our modification is that the load time and unload times now depend upon the configuration. The parameter $loadTime_{s,g}$ now depends upon both the ship *s* and configuration *g*. The parameter $unloadTime_{b,g}$ now depends upon the beach/LZ *b* and configuration *g*. To incorporate this modification, we add one binary parameter and tweak two constraints.

The binary parameter $d_{s,rs,g}$ associates ship departure runs with the configuration type used for each respective platform. That is $d_{s,rs,g}$ =1 if run *rs* from ship *s* has configuration *g*, otherwise it is zero. This is a synthesis of the prioritized data frame produced by the *prioritizer* and the rough schedule produced by the *Assignment Heuristic*. For example, if the heuristic dictates that the first run departing the LHD is an LCAC destined to Red Beach, and the *prioritizer* has indicated that the first run of an LCAC to Red Beach should be configuration A, then $d_{LHD,1,A} = 1$. It is in this manner that we map the configurations used to each of the runs scheduled. The $d_{s,rs,g}$ parameter is an output of the Assignment Heuristic.

A detailed mathematical formulation of the updated *Scheduler Linear Program* [3] is presented below, along with a description of each constraint.

**Indices and Sets**

| | |
|---|---|
| $g \in Gs_s$ | set of configurations |
| $s \in S$ | ships |
| $b \in B$ | land nodes that can be directly reached by this connector type |
| $i \in L$ | 6 time points for each run |
| $rs \in Rx_s$ | set of runs from each ship |
| $rb \in Ry_b$ | set of runs from each land node |

**Data [units]**

$numConn_s$ $numConn_s$ number of connectors per ship *s*

$eunsShip_s$ $runsShip_s$ total number of runs from each ship s

$runsBeach_b$ $runsBeach_b$ total number of runs to each land node *b*

$spotsShip_s$ total number of welldeck or flightdeck spots for each ship *s*

| | |
|---|---|
| $spotsBeach_b$ | total number of landing spots at each land node *b* |
| $transTime_{s,b}$ | transit time from ship *s* to land node *b* [minutes] |
| $loadTime_{s,g}$ | time to load connector on ship *s* for configuration *g* [minutes] |
| $unloadTime_{b,g}$ | time to unload connector at land node *b* for connector *g* [minutes] |
| $dayLength$ | day length of welldeck/flight crew day [minutes] |
| $onWSlack$ | (on water slack) slack time value to limit time connectors are vulnerable to waiting to unload |
| $a_{s,rs,b,rb}$ | Binary parameter that associate ship departure runs with beach runs: $a_{s,rs,b,rb} = 1$ if the $rs$ departure run of ship *s* corresponds to the $rb$ run of land node $b$ |
| $c_{b,rb,s,rs}$ | Binary parameter that associate beach runs with sup arrival runs: $c_{b,rb,s,rs} = 1$ if the *rb* run of land node *b* corresponds to the *rs* return run of ship $s$. |
| $d_{s,rs,g}$ | Binary parameter that associate ship departure runs with configuration type: $d_{s,rs,g} = 1$ if the *rs* run of ship *s* is configuration *g*. |

**Decision Variables [units]**

| | |
|---|---|
| $X_{i,s,rs}$ | the time of the *ith* event of the *rs* departing run from ship *s* |
| $Y_{i,b,rb}$ | the time of the *ith* event of the *rb* run of beach *b* |
| $Z_{i,s,rs}$ | the time of the *ith* event of the *rs* returning run to ship *s* |
| $welldeckStart_s$ | time to start welldeck/flight deck on ship *s* |
| $welldeckEnd_s$ | time to end welldeck/flight deck on ship *s* |

**Formulation**

$$\min_{\substack{x,y,z,\\ welldeckStart,\\ welldeckEnd}} \sum_{s \in S} \sum_{rs \in Rx_s} X_{6,s,rs} \qquad ( .19)$$

$s.t.$

$$welldeckStart_s \le X_{i,s,rs} \le welldeckEnd_s \qquad \forall i \in I, \forall s \in S, \forall rs \in Rx_s \qquad ( .20)$$

$$welldeckEnd_s = welldeckStart_s + dayLength \qquad \forall s \in S \qquad ( .21)$$

$$X_{2,s,rs} \ge X_{1,s,rs} + \sum_{g \in Gs_s} d_{s,rs,g} loadTime_{s,g} \qquad \forall s \in S, \forall rs \in Rx_s \qquad ( .22)$$

$$X_{3,s,rs} \ge X_{2,s,rs} + \sum_{b \in B} \sum_{rb \in Ry_b} a_{s,rs,b,rb} transTime_{s,b} \qquad \forall s \in S, \forall rs \in Rx_s \qquad ( .23)$$

$$Y_{4,b,rb} \ge Y_{3,b,rb} \qquad \forall b \in B, \forall rb \in Ry_b \qquad ( .24)$$

$$Y_{5,b,rb} \ge Y_{4,b,rb} + \sum_{s,rs \in Rx_s} \sum_{g \in Gs_s} a_{s,rs,b,rb} d_{s,rs,g} unloadTime_{b,g} \qquad \forall b \in B, \forall rb \in Ry_b \qquad ( .25)$$

$$Y_{6,b,rb} \ge Y_{5,b,rb} + \sum_{s \in S} \sum_{rb \in Ry_b} a_{s,rs,b,rb} transTime_{s,b} \qquad \forall b \in B, \forall rb \in Ry_b \qquad ( .26)$$

$$X_{4,s,rs} - X_{z,s,rs} \le onWSlack \sum_{b \in B} \sum_{rb \in Ry_b} a_{s,rs,b,rb} transTime_{s,b} \qquad \forall s \in S, \forall rs \in Rx_s \qquad ( .27)$$

$$X_{i,s,rs} \ge X_{i-1,s,rs} \qquad \forall i \in I, \forall s \in S, \forall rs \in Rx_s \qquad ( .28)$$

$$Z_{i,s,rs} \ge Z_{i-1,s,rs} \qquad \forall i \in I, \forall s \in S, \forall rs \in Rx_s \qquad ( .29)$$

$$Y_{i,b,rb} \ge Y_{i-1,b,rb} \qquad \forall i \in I, \forall b \in B, \forall rb \in Ry_b \qquad ( .30)$$

$$X_{1,s,rs} \ge X_{2,s,rs-spotsShip_s} \qquad \forall s \in S, \forall rs \in Rx_s \qquad ( .31)$$

$$Y_{4,b,rb} \ge Y_{5,b,rb-spotsBeach_b} \qquad \forall b \in B, \forall rb \in Ry_b \qquad ( .32)$$

$$X_{1,s,rs} \ge Z_{6,s,rs-numConn_s} \qquad \forall s \in S, \forall rs \in Rx_s \qquad ( .33)$$

$$X_{2,s,rs} \ge X_{2,s,rs-1} \qquad \forall s \in S, \forall rs \in Rx_s \qquad ( .34)$$

$$Y_{4,b,rb} \ge Y_{4,b,rb-1} \qquad \forall b \in B, \forall rb \in Ry_b \qquad ( .35)$$

$$Z_{6,s,rs} \ge Z_{6,s,rs-1} \qquad \forall s \in S, \forall rs \in Rx_s \qquad ( .36)$$

$$Y_{i,b,rb} = \sum_{s \in S} \sum_{rs \in Rx_s} a_{s,rs,b,rb} X_{i,s,rs} \qquad \forall i \in I, \forall b \in B, \forall rb \in Ry_b \qquad ( .37)$$

$$Z_{i,s,rs} \ge \sum_{b \in B} \sum_{rb \in Ry_b} c_{b,rb,s,rs} Y_{i,b,rb} \qquad \forall i \in I, \forall s \in S, \forall rs \in Rx_s \qquad ( .38)$$

$$X_{i,s,rs} \geq 0 \qquad \forall i \in I, \forall s \in S, \forall rs \in Rx_s \qquad ( .39)$$

$$Y_{i,b,rb} \geq 0 \qquad \forall i \in I, \forall b \in B, \forall rb \in Ry_b \qquad ( .40)$$

$$Z_{i,s,rs} \geq 0 \qquad \forall i \in I, \forall s \in S, \forall rs \in Rx_s \qquad ( .41)$$

$$welldeckStart_s \geq 0 \qquad \forall s \in S \qquad ( .42)$$

$$welldeckEnd_s \geq 0 \qquad \forall s \in S \qquad ( .43)$$

We copy below the explanation of the components of the model from Christafore [3]. Only (A.4) (the load constraint) and (A.7) (the unload constraint) differ from the model in Chapter III.3 of Christafore.

1. Objective function A.1 seeks to minimize the time associated with the final departing run for each connector. This forces the model to produce a schedule that minimizes the overall time associated with the ship to shore schedule.

2. Constraints A.2 and A.3 ensure all the connectors are scheduled within the user prescribed operational time window.

3. Constraint A.4 prevents connectors from departing the seabase until they have been loaded with their supplies. **Note that the load time depends upon the configuration and requires knowledge of the binary parameter *d*.**

4. Constraint A.5 prevents a connector from arriving at its destination until it has departed the seabase and completed its transit.

5. Constraint A.6 prevents a connector from offloading its cargo until it arrives at its destination.

6. Constraint A.7 holds connectors at the delivery node until their offload has been completed. Note that the unload time depends upon the configuration and requires knowledge of the binary parameter *d*.

7. Constraint A.8 ensures that connectors have departed the delivery node and completed their transit before returning to the seabase.

8. Constraint A.9 limits the amount of time a connector can sit idle off the coast as they wait to deliver their supplies. This prevents unwanted congestion at the land nodes and produces a deconflicted schedule and efficient delivery schedule.

9. Constraints A.10-A.12 ensure that the proper sequencing of events is maintained as connectors are dispatched from the seabase to complete their deliveries for the three decision variables *X*, *Y* and *Z*.

10. Constraint A.13 prevents a connector from loading aboard a ship until there is a connector position available.

11. Constraint A.14 prevents a connector from offloading at a beach or landing zone until a spot for that connector has become available.

12. Constraint A.15 prevents a connector from reloading at the seabase until after it has completed its previous delivery.

13. Constraints A.16-18 ensures that the proper order of events is maintained when connectors are dispatched to complete their deliveries.

14. Constraint A.19 connects every outgoing run from a ship to a land node.

15. Constraint A.20 connects every outgoing run from a beach to a ship.

16. Constraints A.21-25 ensure that the decision variables and well deck start/end times are non-negative.

THIS PAGE INTENTIONALLY LEFT BLANK

# APPENDIX B. ROUNDING TECHNIQUES AND CONFIGURATION FIDELITY

For the results presented in the main text, we use the mixed integer linear program discussed in Section III.A.4, which ensures integrality for the number of runs generated by the Quickest Flow model. However, depending on the complexity of the problem and the planning timeline the mixed integer linear program may not solve quickly enough. In these cases, we can use to the linear program version from Section III.A.2 to generate a schedule in a reasonable amount of time. While the linear program model can solve relatively quickly, it may produce fractional runs, which we must convert to integer runs to create an operational schedule. Consequently, in Appendix B.B we focus on ensuring that we have an integer number of runs for each configuration-land node pair. The conversion from fractional to integer runs can result in either more or fewer supplies delivered to a node than the Quickest Flow delivery plan desires, which leads to inefficiencies. Before presenting our rounding methods, we present an analysis on potential excess supplies delivered ashore. We specifically focus on how the fidelity of the configuration list developed by the planners impacts the delivery excess.

For the remainder of this appendix we shall use the same example from Chapter IV.A; however, we use the linear program version of the Quickest Flow model to demonstrate the separate rounding algorithms we create. Examples of MV-22 run counts appear in Table 22 and 23 reflecting the effects of a non-penalized and penalized objective function.

Table 22. Linear Program MV-22 Output with no Penalty or Rounding

| Destination | Configuration | Required Runs |
|---|---|---|
| **B1** | **T** | 21.179 |
| | **U** | 0.055 |
| | **W** | 6.599 |
| | **X** | 10.3 |
| | **Y** | 1.859 |
| **B2** | **T** | 7.237 |
| | **U** | 0.216 |
| | **W** | 0.217 |
| | **X** | 3.784 |
| | **Y** | 0.223 |
| **MAIN** | **T** | 10.051 |
| | **U** | 0.22 |
| | **V** | 8.458 |
| | **W** | 4.871 |
| | **X** | 6.028 |
| | **Y** | 4.829 |
| **LZ1** | **T** | 4.9 |
| | **U** | 2.656 |
| | **V** | 2 |
| | **W** | 2 |
| **Total Required Runs:** | | **97.682** |

Table 23. Linear Program MV-22 Output Reflecting $\beta$ =0.01

| Destination | Configuration | Required Runs |
|---|---|---|
| **LZ1** | T | 0.4 |
| | V | 2 |
| **MAIN** | T | 6.103 |
| | U | 0.164 |
| | X | 2.443 |
| **Total Required Runs:** | | **11.11** |

## A. CONFIGURATION FIDELITY AND EXCESS

In Appendix B.B we convert the fractional runs from Table 22 and 23 to integers to determine the final runs allocation. Before turning to this analysis, we discuss in this section how we evaluate the effectiveness of a runs-allocation in terms of what commodities the plan delivers ashore.

One of our main outputs of interest in our analysis is the consumption plan. That is the total amount of commodity $c$ consumed at node $l$ during the time period of interest $\sum_{\substack{t \in T, \\ ss \in SS}} X_{l,ss,t,c} \; \forall c \in C, \forall l \in L$ .

In Chapters IV and V, we consider how this consumption compares to the total demand to derive the demand deficit. Note the amount consumed cannot be greater than demand and hence the deficit cannot be negative.

For the remainder of this Appendix, we turn our focus to the delivery plan specified by Quickest Flow. That is the total amount of commodity $c$ delivered to node $l$ directly from the seabase during the time period of interest

$$\sum_{\substack{s \in S, \\ t \in T}} X_{s,l,t,c} \; \forall c \in C, \forall l \in L \; .$$

Because we use the linear program version of Quickest Flow in this Appendix, the delivery plan in Table 24 is slightly different than the delivery plan in Table 7 of Chapter IV This is the desired delivery plan.

Table 24. Effects of  Value on Supply Delivered Directly from the Seabase to a Land Node

| Water Delivery vs β Value | | | | |
|---|---|---|---|---|
| | **0** | **0.00001** | **0.0001** | **0.01** |
| **B1** | 10 | 10.025 | 10.14 | 12.002 |
| **B2** | 3 | 3 | 3.025 | 5.143 |
| **Main** | 5 | 4.975 | 4.834 | 0.855 |
| **LZ1** | 0.4 | 0.4 | 0.4 | 0.4 |

| Fuel Delivery vs β Value | | | | |
|---|---|---|---|---|
| | **0** | **0.00001** | **0.0001** | **0.01** |
| **B1** | 15.12 | 15.737 | 15.136 | 15.538 |
| **B2** | 2.129 | 2.167 | 4.209 | 4.118 |
| **Main** | 5.751 | 5.096 | 3.655 | 3.524 |
| **LZ1** | 0.2 | 0.2 | 0.2 | 0.2 |

| PAX Delivery vs β Value | | | | |
|---|---|---|---|---|
| | **0** | **0.00001** | **0.0001** | **0.01** |
| **B1** | 248.655 | 278.86 | 266.251 | 294.32 |
| **B2** | 86.26 | 111.8 | 139.634 | 157.199 |
| **Main** | 155.085 | 98.34 | 84.115 | 38.481 |
| **LZ1** | 140 | 140 | 140 | 140 |

Given a final run allocation *Z*, we can determine how much is delivered to each land node. We compare the actual delivery plan to the desired delivery plan; the actual delivery may be more or less than the desired delivery plan from Quickest Flow.

We define the delivery excess (deficit if negative) as

$$\frac{\sum_{(s,l)\in A}\sum_{g\in Gs_s}\sum_{t\in T} Z_{s,l,g,t}\, config_{s,g,c} - \sum_{s\in S}\sum_{t\in T} X_{s,l,t,c}}{\sum_{(s,l)\in A}\sum_{t\in T} X_{s,l,t,c}} \quad \forall c\in C, \forall l\in L \ .$$

The denominator is the desired delivery plan from Quickest Flow. The first term in the numerator is the actual amount delivered using a given *Z* delivery plan, assuming all runs are fully loaded. Table 25 specifies this delivery excess or deficit for the Quickest Flow plans for four values of β assuming all connectors are fully loaded when they make their run. In practice connectors would be full when making delivers ashore. We use the fractional runs directly outputted by the Quickest Flow model (see Table 22 and 23), and therefore by construction there can be no deficit in Table 25. Table 25 illustrates that fully loaded connectors could result in sizeable excesses. While operationally we want to load as many connectors as completely as possible, to do so across the board would result in wasteful deliveries. Although excess commodity delivery is not as concerning as deficit, it remains an inefficiency within our current model that we must consider. Looking more closely at Table 25 it becomes apparent that there are some commodities that experience extreme excess at some of the nodes. We cannot put much stock in the $\beta=0$ results because it is likely the optimal solution generates many partial runs. However, for positive $\beta$, there are several examples of excess well over 100%. This is a result of there being unused cargo space aboard many of the connectors as certain commodities begin to become completely satisfied. For example, any LCAC configuration can always transport at least 25 Marines. Even after satisfying demand for PAX at a beach node, all subsequent LCAC runs will generate excess.

Table 25. Delivery Excess (if all loads depart fully loaded)

| | | **β=0** | **β=0.00001** | **β=0.0001** | **β=0.01** |
|---|---|---|---|---|---|
| Destination | Commodity | Excess % | Excess % | Excess % | Excess % |
| **B1** | Water | 120.00% | 100.00% | 100.00% | 120.00% |
| | Fuel | 120.00% | 106.67% | 100.00% | 100.00% |
| | AMMO | 185.00% | 125.00% | 155.00% | 140.00% |
| | PAX | 707.00% | 140.00% | 148.00% | 147.00% |
| | HMMWV | 237.50% | 225.00% | 437.50% | 437.50% |
| **B2** | Water | 100.00% | 100.00% | 100.00% | 166.67% |
| | Fuel | 250.00% | 100.00% | 200.00% | 200.00% |
| | AMMO | 133.33% | 133.33% | 150.00% | 300.00% |
| | PAX | 614.00% | 238.00% | 280.00% | 314.00% |
| | HMMWV | 200.00% | 250.00% | 325.00% | 350.00% |
| **MAIN** | Water | 200.00% | 166.67% | 166.67% | 33.33% |
| | Fuel | 200.00% | 166.67% | 133.33% | 133.33% |
| | AMMO | 250.00% | 141.67% | 150.00% | 0.00% |
| | PAX | 261.25% | 122.50% | 105.00% | 47.50% |
| | HMMWV | 80.00% | 80.00% | 0.00% | 0.00% |
| **LZ1** | Water | 0.00% | 0.00% | 0.00% | 0.00% |
| | Fuel | 1000.00% | 0.00% | 0.00% | 0.00% |
| | AMMO | 100.00% | 100.00% | 100.00% | 100.00% |
| | PAX | 100.00% | 100.00% | 100.00% | 100.00% |
| | HMMWV | 100.00% | 100.00% | 100.00% | 100.00% |

We next illustrate that these delivery excesses are not a flaw with our model, but a result of low fidelity in the number of configurations inputted by the user. Consider an example when Beach 2 has a fuel demand of two units and a water demand of zero units. We have an LCAC with one configuration to carry 0.5 units of both fuel and water. To satisfy demand for fuel requires four runs, which results in two units of excess water delivered if the LCAC departs fully loaded. It would be better to dispatch two LCACs that would be fully loaded with only fuel. This results in both fewer runs and less (in this case 0) excess. This issue does not stem from a problem within the model but is instead a result of insufficient realistic configurations for each of the connectors. This section explores the idea of configuration fidelity in relation to excess capacity.

As the above example illustrates, the number of configurations used can have a direct impact on total run count and the amount of deficit/excess capacity delivered ashore. A greater number of well thought out configurations for each of the platforms provides more options for the Quickest Flow Model to choose from, resulting in more efficient ship-to-shore deliveries. Having enough configurations enumerated prior to running the model can largely reduce the total number of excess runs.

The objective of our loading process is to ensure that every connecter departs as fully loaded as possible. For the purposes of demonstrating the impact of configuration fidelity, we focus exclusively on excess capacity generated when comparing a scenario with few vs many configurations.

To demonstrate the impact that configuration fidelity can have on the total excess we re-ran our Chapter IV scenario 10-times using increasing numbers of configurations per platform. To generate large numbers of configurations we ran a script that created 2,350 total configurations across all our connectors by permuting each of the commodity values by 20 percent increments of their max capacity per platform. For example, if an LCAC's max capacity for fuel was one, then we would vary it by 0.2 and fill the unused space by another commodity until every combination had been enumerated. An example of this configuration generation is presented in Table 26.

Table 26. Example of Configuration Generation

| Connector | Config | WATER | FUEL | AMMO | PAX | HMMWV | loadTime | unloadTime |
|---|---|---|---|---|---|---|---|---|
| CH53 | A | 0 | 0 | 0 | 0 | 1 | 35 | 25 |
| | B | 0 | 0 | 0 | 37 | 0 | 35 | 25 |
| | C | 0 | 0 | 1 | 29 | 0 | 35 | 25 |
| | D | 0 | 0 | 2 | 25 | 0 | 35 | 25 |
| | E | 0 | 0 | 3 | 22 | 0 | 35 | 25 |
| | F | 0 | 0 | 4 | 18 | 0 | 35 | 25 |
| | G | 0 | 0 | 5 | 11 | 0 | 35 | 25 |
| | H | 0 | 0 | 6 | 7 | 0 | 35 | 25 |
| | I | 0 | 0 | 7 | 3 | 0 | 35 | 25 |
| | J | 0 | 0 | 8 | 0 | 0 | 35 | 25 |

While this is useful for exploring the effects of configuration fidelity on excess, this does not represent realistic packing plans for our platforms. For example, in practice ammo and fuel should not be transported together. With the full list of 2,350 configurations generated in this manner, we then create 10-additional lists of configurations by reducing the total configuration count by randomly deleting configurations for each of our connectors (ensuring each connector has the same number of final configurations) until our lists ranged from 20 to 1412 configurations. We then ran the Quickest Flow Model 10-times with our different configuration lists to measure the impact configuration fidelity has on the resulting delivery plans.

For every trial we calculate the total delivery excess across all nodes for each supply commodity. These values were then normalized to the quantity of a fully loaded MV-22 and summed together to reflect total excess across all commodities. To convert the excess values by commodity into MV-22 loads we use the values presented in Table 27. For example, if our trial indicated that we had two excess fuel units, three water units, and 36 PAX this would equate to four, six, and 1.5 MV-22 runs, respectively.

Table 27. Maximum MV-22 Load per Commodity

| Water | Fuel | AMMO | PAX | HMMWV |
|---|---|---|---|---|
| 0.5 | 0.5 | 4 | 24 | 1 |

Figure 19 illustrates the total excess in equivalent MV-22 units for the different configuration trials. We see a significant drop of excess capacity occurring when the total configuration count is 40 and gradually approaches zero at 400.

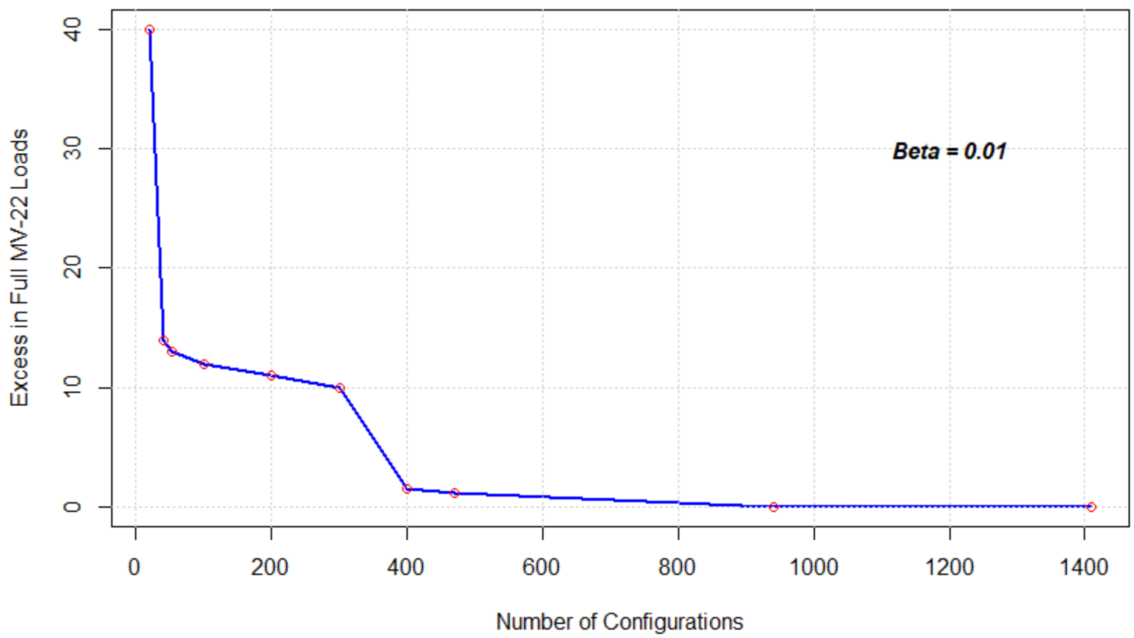



Figure 19. Effects of Configuration Count on Excess Capacity.

We only generate one random configuration list for a fixed configuration size in Figure 19. As discussed earlier the configurations are not operationally realistic. Our purpose with Figure 19 is just to illustrate that improved configuration fidelity can significantly reduce or even eliminate excess.

The results in Figure 19 support the argument that accurately enumerating many realistic configurations per platform produces schedules with diminished excess capacity, making the final schedule more efficient. This requires an upfront cost to the user in terms of time needed to accurately record realistic configurations. However, once these configurations are transferred from unit Standard Operating Procedures and subject matter expertise, a user will have a reliable list of configurations that they can use for future scenarios. For the remainder of this chapter, we shall focus on a smaller number of configurations (26) to ease our analysis. This results in some delivery excess; however, this is not concerning as we know we can always reduce that quantity with a greater number of configurations. As we turn to converting the fractional run plan to an integer run plan, our primary focus is on delivery deficits and creating an integer run plan that has a small amount of delivery deficit and a relatively small number of additional runs to reduce that deficit.

## B. ROUNDING FUNCTIONS

The desired Quickest Flow Model includes integer variables (see Chapter III.A.4) to produce a plan with an integer number of runs. This integer linear program implementation is more straightforward and mathematically efficient than other rounding techniques to generate an integer number of runs. We prefer to use this integer linear program version where possible because the output returns integer values for our run counts. However, due to potentially long computational run times, we may have to use the linear program model that produces fractional run values. We then must convert these fractional values to integers to create a legitimate schedule. In this section we discuss these rounding techniques when we use the linear program version. We then compare these rounding techniques to the preferred integer linear program results.

Christafore's 2017 work addresses the issue of non-integer runs by rounding all the values up to the nearest integer [3]. This rounding-up approach may create problems for us because our model can consist of several hundred configurations, whereas Christafore had only one configuration per platform (100% fuel). To illustrate consider the following trivial example. The linear program Quickest Flow specifies a plan to send LCACs to Beach 1 via five configurations: A, B, C, D, E. The output of the model is 0.1 runs for each of the five configurations. If we round all five run values up, this generates a total of five runs to Beach 1. However, the amount of supplies desired by Beach 1 is less than one fully-loaded LCAC, so rounding up results in excessive utilization of LCAC runs and a great deal of unused space. This notion of excess runs and delivery capacity becomes more complicated as the total number of configurations increases to non-trivial quantities.

At this point in our analysis we have a fixed $\beta$ value (0.01) and a fixed configuration list. However, the output of the Quickest Flow Model remains non-integer (see example in Table 23 and 28) and we now focus on converting these fractional amounts to integer values. We aim to create an integer run plan whose actual delivery plan roughly corresponds to the desired delivery plan produced by the Quickest Flow using fractional runs.

Table 28. LCAC B2 Non-Integer Run Requirements

| Configuration | Required Runs |
|---|---|
| D | 0.86 |
| G | 4.72 |
| F | 2.06 |

We employ two greedy rounding techniques and measure the performance based on the amount of delivery deficit they produce and number of total runs. We began by analyzing the delivery deficit for each commodity at each of the land nodes and then transition to total delivery deficit across all land nodes for each commodity. We ignore excess delivery for most of our analysis in this section; our primary objective for this section is to identify the rounding technique that produces the lowest delivery deficit. Deficit is calculated via the same method presented in Appendix B.A Utilizing these metrics, we analyze the performance of the following rounding techniques:

- Rounding by Fraction
- Rounding by Deficit

### 1. Rounding by Fraction

In this section we explore the fractional rounding technique by walking through a concrete example presented in Table 29. As a reminder, arbitrarily rounding all our run values up would produce an excess of runs. If we round up all the run values in Table 29, this would produce 12 total runs, which is over two more runs than we need.

Table 29. Example of LCAC Requirements

| LCAC to B1 | |
|---|---|
| **Configuration** | **Runs** |
| **A** | 1.37 |
| **B** | 0 |
| **C** | 2.278 |
| **D** | 1.03 |
| **E** | 4.964 |
| **Total:** | **9.642** |

The fractional rounder represents a naïve approach and focuses on the runs required per platform to deliver supplies to a single node. For a given *(s,l)* pair, we first compute the total number of runs across all configurations of the given supply type *s* ( $\sum_{t \in T} \sum_{g \in Gs_s} Z_{s,l,g,t} \; \forall s \in S, \forall l \in L$ ). In general, this is non-integer. For example, in Table 29 this is 9.642. We want to round all the specific configurations for this *(s,l)* pair so that the sum of these rounded values is within one of the fractional total. For example, in Table 29 the integer sum is 10. This maximum number of total runs is determined by summing all the fractional values (Table 31) and taking the ceiling function.

To determine the final run-allocation, we first take the floor values produced by the linear Quickest Flow (Table 29) to produce Table 30.

Table 30. Floor Values of LCAC Requirements

| LCAC to B1 | |
|---|---|
| **Configuration** | **Runs** |
| **A** | 1 |
| **B** | 0 |
| **C** | 2 |
| **D** | 1 |
| **E** | 4 |
| **Total:** | **8** |

We make modifications of the baseline allocation in Table 30 to produce the final allocation. We next determine the maximum number of runs available for further allocation. We subtract the number of runs currently allocated (eight from Table 20) from the maximum number of possible runs (10 from taking the ceiling in Table 29), which yields a maximum of two runs to further allocate. To allocate the remaining runs we sort the configurations by their fractional part. That is, we take each quantity in Table 29 and we subtract its floor (Table 30). The sorted configurations appear in Table 31.

Table 31. Sorted LCAC Requirements

| LCAC to B1 | |
|---|---|
| **Configuration** | **Runs** |
| **E** | 0.964 |
| **A** | 0.37 |
| **C** | 0.278 |
| **D** | 0.03 |
| **B** | 0 |
| **Total:** | **1.642** |

We then proceed down the sorted list of configurations, adding one to that configuration's floor value (Table 30), until we reach the maximum number of runs for this *(s,l)* pair (i.e., two extra runs). Table 32 presents the final allocation of runs. This process yields one extra run to the two configurations with the highest fractional value, in this case E and A.

Table 32. Final LCAC Run Requirements

| LCAC to B1 | |
|---|---|
| **Configuration** | **Runs** |
| **A** | 2 |
| **B** | 0 |
| **C** | 2 |
| **D** | 1 |
| **E** | 5 |
| **Total:** | **10** |

There is a small issue when using the above approach. If we apply this approach to every node accessible to an LCAC, then the final run-count for each node becomes the ceiling of the run-count generated by Quickest Flow. For the above example, Quickest Flow produced 9.642 runs and our approach generates a final run-count of 10 and allows us to allocate two additional runs from the baseline in Table 30. However, if we take the ceiling across many nodes, then the final total run-count across all nodes might exceed the amount allowable. The total allowable run count corresponds to the parameter $supply_s$ in Chapter III, see constraint 3.12 in the Quickest Flow model, $\sum_{l \in L} \sum_{g \in Gs_s} \sum_{t \in T} Z_{s,l,g,t} \leq supply_s \; \forall s \in S$. Thus, we perform an additional check across all nodes, similar to the one described for the one node case above, to ensure we do not allocate more runs than allowed. We describe this check now with a concrete example based on the information in Table 33. The values in Table 33 correspond to the initial run allocation across three beaches (similar to Table 30) and the fractional remainder (similar to Table 31).

Table 33. LCAC Run Floor Values with Available Runs

| | **B1** | **B2** | **B3** |
|---|---|---|---|
| **Runs Floor Value** | 8 | 6 | 2 |
| **Available Runs** | 1.642 | 1.1 | 0.2 |

If we apply the procedure described at the beginning of this section, then we would allocate 21 total runs (10 to B1, eight to B2, three to B3). If the maximum number of possible LCACs runs across all nodes (i.e., $supply_s$ ) is 22 runs, then the original procedure is valid, and we can round up all available runs to get the final number of runs to allocate. However, if the maximum possible run count is 20, then rounding up all available runs would produce an infeasible allocation. If taking the ceiling of available runs produces an infeasible allocation, then we generate a feasible allocation in the following way. First, we add to the floor value in Table 33, the floor of the fractional available runs. Next, we sort the land nodes by the remaining fractional runs available. This transforms Table 33 to Table 34.

Table 34. Sorted Land Nodes Based on Fractional Available Runs

| Sorted Land Node | Fractional Available Run | Updated Runs |
|---|---|---|
| **B1** | 0.642 | 9 |
| **B3** | 0.2 | 2 |
| **B2** | 0.1 | 7 |

We then proceed down the sorted list adding one additional run to the final count for each beach until we reach the maximum possible runs across all beaches. For our example if 20 is the maximum possible run count, then our update in 24 gives us 18 total runs. Thus, we can add two additional runs. We add one run to B1 and one run to B3. The final run count for each beach appears in Table 35. Once we have this final count for each node, we can follow the steps described above to move from Table 30 to 32 to generate the final run-count by configuration.

Table 35. Final Run Count by Node

| | B1 | B2 | B3 |
|---|---|---|---|
| **Final Run Count** | 10 | 7 | 3 |

### 2. Rounding by Deficit

The Deficit Rounding technique begins by determining the maximum number of runs available for each *(s,l)* pair in the same manner we describe for the fractional rounder in Appendix B.B.1. Table 36 presents similar results to Table 29–30 for a more involved example using LCACs and LCUs and two beaches. Table 36 contains the fractional runs and Table 37 applies the floor function. A notable change from the fractional rounder is that the deficit reducer considers all platforms that can deliver to a single land node and determines the connector *s* and configuration *g* that best satisfies demand.

Table 36. LCAC Fractional Run Values

| **LCAC to B1** | | **LCAC to B2** | | **LCU B2** | |
|---|---|---|---|---|---|
| **Configuration** | **Runs** | **Configuration** | **Runs** | **Configuration** | **Runs** |
| **A** | 1.07 | **A** | 0 | **T** | 0.89 |
| **B** | 0 | **B** | 3.769 | **U** | 0 |
| **C** | 2.13 | **C** | 1.28 | **V** | 0 |
| **D** | 1.21 | **D** | 1.56 | **W** | 0 |
| **E** | 4.03 | **E** | 2.36 | **X** | 0 |
| **Total:** | 8.44 | **Total:** | 8.969 | **Total:** | 0.89 |
| **Available** | **1** | **Available** | **2** | **Available** | **1** |

Table 37. LCAC Run Floor Values and Available Runs

| **LCAC to B1** | | **LCAC to B2** | | **LCU B2** | |
|---|---|---|---|---|---|
| **Configuration** | **Runs** | **Configuration** | **Runs** | **Configuration** | **Runs** |
| **A** | 1 | **A** | 0 | **T** | 0 |
| **B** | 0 | **B** | 3 | **U** | 0 |
| **C** | 2 | **C** | 1 | **V** | 0 |
| **D** | 1 | **D** | 1 | **W** | 0 |
| **E** | 4 | **E** | 2 | **X** | 0 |
| **Total:** | 8 | **Total:** | 7 | **Total:** | 0 |
| **Available** | **1** | **Available** | **2** | **Available** | **1** |

After we generate the initial run count in Table 37, we calculate the delivery deficit that would incur for each of the commodities at each node. We define $T_{s,l,g}$ as an arbitrary run-plan. That is the number of integer runs of configuration *g* of type *s* to node *l*. An example of such a run plan is configuration runs listed in Table 25. Given this run-plan we can compute how much of each commodity is delivered to node l. We refer to this as the *actual delivery plan.* We then compare this actual plan to the desired delivery plan generated by the Quickest Flow Model. That is, we define the delivery deficit as:

$$\sum_{s\in S}\sum_{t\in T} X_{s,l,t,c} - \sum_{\substack{s\in S,\\ g\in Gs_s}} T_{s,l,g} config_{s,g,c} \ \forall l \in L, \forall c \in C \, .$$

For our example if we define $T_{s,l,g}$ as the plan in Table 37, then the delivery deficit appears in Table 38.

Table 38. Incurred Delivery Deficit at Beaches

| | **B1** | **B2** |
|---|---|---|
| **FUEL** | 1 | 0.8 |
| **WATER** | 0 | 2 |
| **PAX** | 26 | 15 |
| **HMMWV** | 2 | 1 |
| **AMMO** | 8 | 4 |

As with the Fractional Rounder, for each *(s,l)* we have a total number of extra runs to allocate to that pair as we iterate through this algorithm (Table 37 Available Runs). We want to allocate these additional runs to reduce the delivery deficit shown in Table 38. Framed from a different perspective, when we assign these additional runs we want the connector to have as much of its volume filled with supplies with an outstanding deficit in Table 38. If an *(s,l)* pair has unallocated runs available, we compute the percentage of volume filled if we next assign configuration *g* of type *s* on a run to node *l*. We only include the volume if we can apply it to outstanding deficit in Table 38. For example, if we are considering configurations to send to B1, then any space allocated to water on the configuration would not count toward our total effective volume because there is no water deficit at B1. In the following table we present the effective volume for each configuration across the two land nodes (Table 39).

Table 39. Effective Volume by Configuration

| LCAC to B1 | | LCAC to B2 | | LCU B2 | |
|---|---|---|---|---|---|
| **Configuration** | **Volume %** | **Configuration** | **Volume %** | **Configuration** | **Volume %** |
| **A** | 0.63 | **A** | 0.54 | **T** | 0.25 |
| **B** | 0.19 | **B** | 0.14 | **U** | 0.34 |
| **C** | 0.05 | **C** | 0.15 | **V** | 0.62 |
| **D** | 0.27 | **D** | 0.32 | **W** | 0.41 |
| **E** | 0.89 | **E** | 0.72 | **X** | 0.38 |

We then allocate the *(s,g)* configuration to go the land node *l* such that *(s,g)* configuration has the greatest percentage of its volume filled. In our example this would be configuration E dispatched to B1. The updated run-plan appears in Table 40.

Table 40. Updated LCAC Run Values and Available Runs

| LCAC to B1 | | LCAC to B2 | | LCU B2 | |
|---|---|---|---|---|---|
| **Configuration** | **Runs** | **Configuration** | **Runs** | **Configuration** | **Runs** |
| **A** | 1 | **A** | 0 | **T** | 0 |
| **B** | 0 | **B** | 3 | **U** | 0 |
| **C** | 2 | **C** | 1 | **V** | 0 |
| **D** | 1 | **D** | 1 | **W** | 0 |
| **E** | **5** (+1) | **E** | 2 | **X** | 0 |
| **Total:** | 9 | **Total:** | 7 | **Total:** | 0 |
| **Available** | **0 (-1)** | **Available** | **2** | **Available** | **1** |

After this new allocation, we recompute the remaining delivery deficit at the nodes. That is, we update Table 40 after including configuration E to B1, which results in Table 41. We then determine the next (*s*,*l*,*g*) combination that has the largest volume of effective supplies (similar to Table 39) and continue the process. This iterative procedure ends either when no deficit remains, or we have assigned all unallocated runs.

Table 41. Updated Deficit at Beaches

| | B1 | B2 |
|---|---|---|
| **FUEL** | 0.5 | 0.8 |
| **WATER** | 0 | 2 |
| **PAX** | 1 | 15 |
| **HMMWV** | 0 | 1 |
| **AMMO** | 0 | 4 |

### 3. Analysis of Rounding Techniques

Utilizing these rounding mechanisms, we run the linear program Quickest Flow Model for our scenario and observe the results for the two rounding methods in Table 42. We also run the mixed integer Quickest Flow variant, which has zero deficit by construction. Both rounding methods have large deficits of AMMO at B2. The fractional rounder has fewer nodes with deficits, but its deficits at B2 are much larger than the deficits of the deficit rounder. The two rounding heuristics are simple and quickly round all configurations for a relatively small price in deficit. These rounding heuristics would be suitable for larger problems where many configurations are used. The mixed integer linear program incurs no deficit but is an integer linear program and as a result can generate excessive computational time if the number of configurations becomes large. For example, to generate the results in Table 42 when $\beta = 0.01$ the mixed integer linear program required a total of 10 minutes of computational time.

Table 42. Deficit per Class of Supply Utilizing Different Rounding Techniques

| Node | Supply | Demand | Unrounded Output X Delivered | Fractional Rounder | Deficit Rounder | QF ILP |
|---|---|---|---|---|---|---|
| B1 | Water | 10 | 12.002 | -0.10% | -6.26% | 0.00% |
| | Fuel | 15 | 15.358 | 0.00% | -5.59% | 0.00% |
| | AMMO | 20 | 27.775 | 0.00% | -6.39% | 0.00% |
| | PAX | 200 | 294.32 | 0.00% | -6.56% | 0.00% |
| | HMMWV | 8 | 15.336 | 0.00% | 0.00% | 0.00% |
| B2 | Water | 3 | 5.143 | -27.09% | 0.00% | 0.00% |
| | Fuel | 2 | 4.118 | 0.00% | -4.08% | 0.00% |
| | AMMO | 6 | 18.225 | -43.90% | -23.18% | 0.00% |
| | PAX | 50 | 157.199 | 0.00% | -4.58% | 0.00% |
| | HMMWV | 4 | 10.664 | 0.00% | 0.00% | 0.00% |
| MAIN | Water | 3 | 0.855 | 0.00% | 0.00% | 0.00% |
| | Fuel | 3 | 3.524 | 0.00% | 0.00% | 0.00% |
| | AMMO | 12 | 0 | 0.00% | 0.00% | 0.00% |
| | PAX | 80 | 38.481 | 0.00% | 0.00% | 0.00% |
| | HMMWV | 10 | 0 | 0.00% | 0.00% | 0.00% |
| LZ1 | Water | 0.4 | 0.4 | 0.00% | 0.00% | 0.00% |
| | Fuel | 0.2 | 0.2 | 0.00% | 0.00% | 0.00% |
| | AMMO | 4 | 4 | 0.00% | 0.00% | 0.00% |
| | PAX | 140 | 140 | 0.00% | -1.43% | 0.00% |
| | HMMWV | 2 | 2 | 0.00% | 0.00% | 0.00% |

Exploring the information further in Table 42 we see that the fractional and deficit rounding mechanisms produce very different results, even though their total run counts are the same. Table 43 contains the differences in the two heuristics for B2. The fractional rounder has one LCAC configuration A run and two for B while the deficit reducer places all three runs for LCAC configuration B. Similarly, the deficit reducer has chosen a different LCU than what the model original selected.

Table 43. B2 Run Requirements Comparison

| | LCAC Configurations | |
|---|---|---|
| | A | B |
| Fractional | 1 | 2 |
| Deficit | 0 | 3 |
| | LCU Configurations | |
| | T | W |
| Fractional | 0 | 1 |
| Deficit | 1 | 0 |

Another facet to consider is the total deficit across all land nodes for each of the commodities. In some circumstances deficit generated from amphibious connectors can be alleviated by the delivery of excess commodities from other nodes if an edge exists to support this. Figure 20 illustrates the total deficit or excess across all nodes. Note this is an imperfect global picture as deficit at one node might be offset by excess at another node. Consequently, Figure 20 presents an overly optimistic picture regarding deficit, but it still provides relevant information for analysis purposes. As a reminder, for the purposes of identifying the best performing rounding technique we are not concerning ourselves with the impacts each technique has on generating excess.

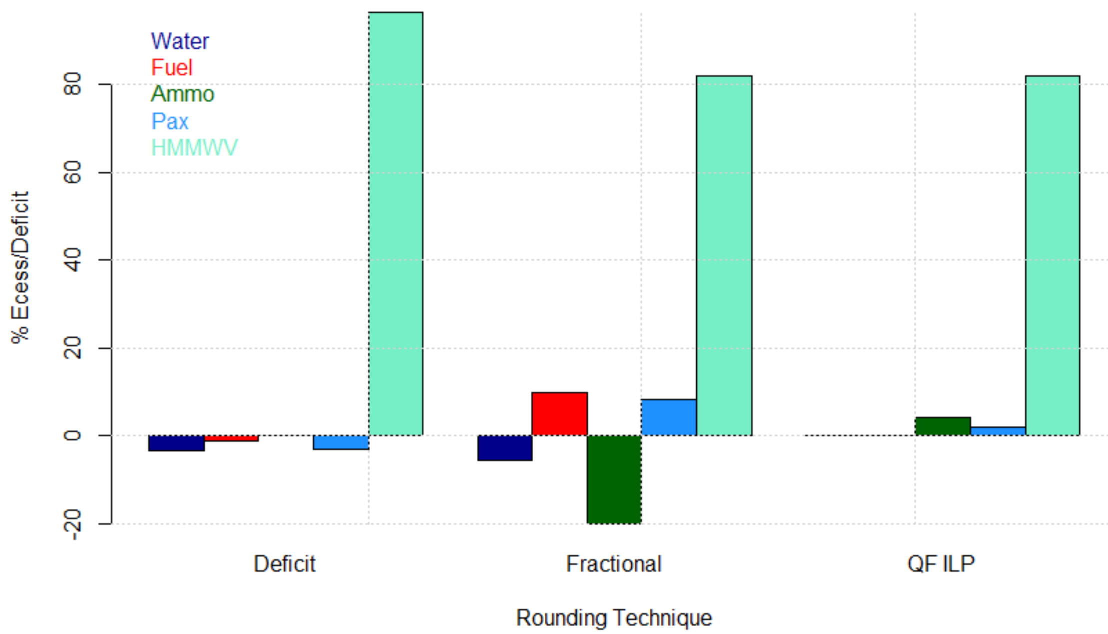



Figure 20. Rounding Techniques Compared by Total Deficit

An important aspect to consider in Figure 20 is that the fractional rounder produces the highest amount of overall deficit when compared to the other two methods. Furthermore, it also has the highest amount of excess capacity for FUEL and PAX deliveries. When we look more closely at the results as they pertain to the amount of demand satisfied ashore when we only concern ourselves with commodities that are correctly delivered (Figure 21), we see a much larger amount of deficit as it pertains to the fractional rounding method. We then conclude that if a user needs to run the linear program Quickest Flow Model due to computational time concerns it would be best to use the deficit reducer as it produces the most accurate results. Future work could perform a more systematic comparison of these two rounding techniques and propose more sophisticated rounding approaches.

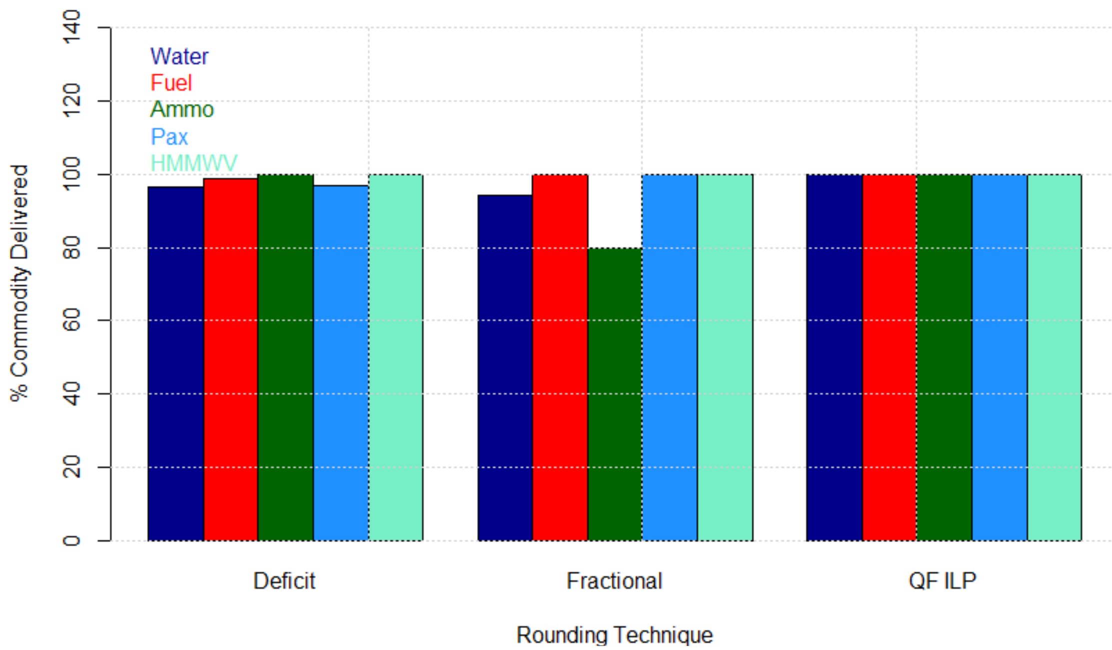



Figure 21. Percent of Demand Satisfied by Commodity

## C. CONCLUSIONS

We conclude that applying a penalty to the objective function and using the mixed integer linear program Quickest Flow model when practical or the deficit reducing rounding technique for the linear program are the most effective means of enhancing efficiency and producing integer runs. These conclusions are supported by the graph in Figure 22, which demonstrates the reduced run count these techniques produce.

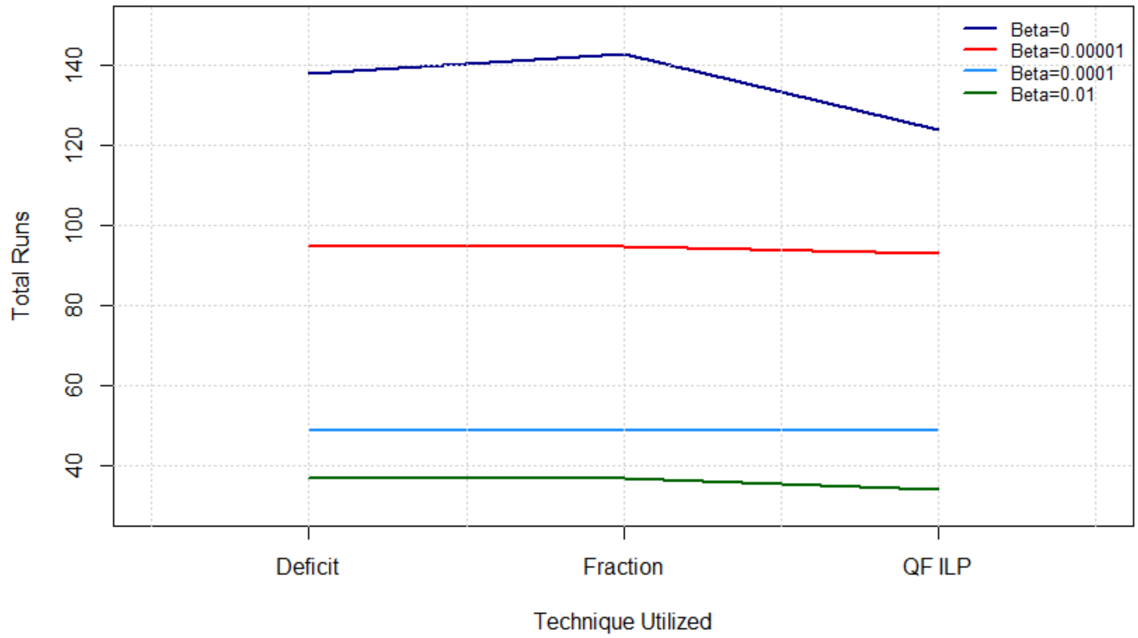



Figure 22. Effects of β and Rounding Techniques on Run Count

Table 44 contains the total run counts for each of our rounding methods when using β =0.00001. As we can see there is significant parity between the three methods; however, the mixed integer linear program demonstrates the best performance for reducing the overall run count, incurring no deficit and minimizing excess (Figure 20).

Table 44. Run Count by Rounding Mechanism

| | **Deficit** | **Fractional** | **QF ILP** |
|---|---|---|---|
| **B1** | 47 | 47 | 46 |
| **B2** | 7 | 7 | 5 |
| **Main** | 31 | 31 | 32 |
| **LZ1** | 10 | 10 | 10 |
| **Total** | **95** | **95** | **93** |

For this analysis we primarily used β =0.01; however, that is not a global value appropriate for all scenarios. Further analysis is required to quantify the relationship of β to the total run count and number of commodities being used. In practice, it is useful from a decision support perspective to vary beta and present the commander with several options.

In a similar light, while the mixed integer linear program is the preferred method for generating integer runs, its computational time will increase as scenarios become more complicated. Should run times for the mixed integer linear program become impractical for planning purposes the user can opt to use the fractional reducer for a marginal cost in deficit. Reflected in Table 45 is the final MV-22 output for our scenario using $\beta$ =0.01 and the mixed integer linear program Quickest Flow Model. These mechanisms had the effect of reducing the overall run count by 88 while concentrating runs to five configurations.

Table 45. Final MV-22 Output Utilizing the LP Rounder and $\beta$ =0.01.

| Destination | Configuration | Required Runs |
|---|---|---|
| **LZ1** | V | 2 |
| **MAIN** | T | 5 |
| | X | 1 |
| **B1** | T | 1 |
| **Total Required Runs:** | | **9** |

THIS PAGE INTENTIONALLY LEFT BLANK

# APPENDIX C. ASSIGNMENT HEURISTIC EXAMPLE

In Chapter III.B, we describe the Assignment Heuristic. To aid in understanding the logic, we present an example in this Appendix to walk through the steps. Section B of this Appendix provides the pseudo-code for this heuristic.

## A. EXAMPLE WALKTHROUGH

The heuristic tracks all available connectors. At any point in time a connector is in one of three states:

- Ready to Load at ship (status 1)
- Ready to Unload at LZ (status two)
- Ready to depart back to beach (status three)

We track the state of each connector and the current time and then process connectors in chronological order. For this example, we shall make the following assumptions using data from Figure 6 and 7 from Chapter III.B:

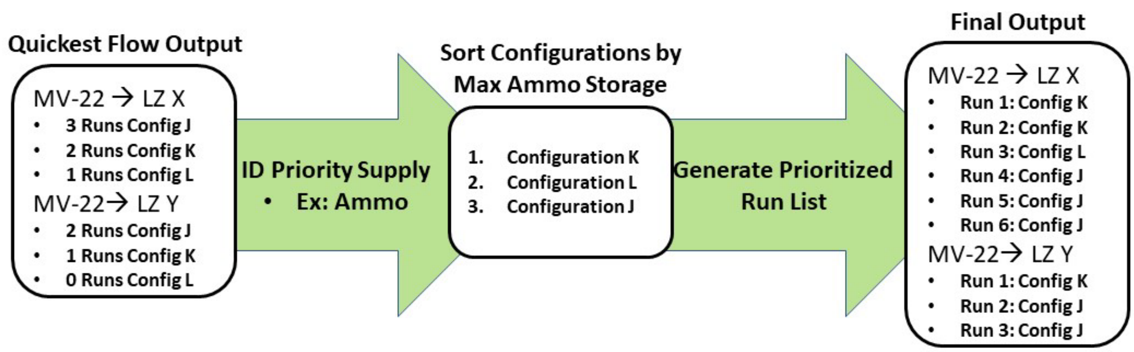



Figure 23. Example of MV-22 Runs being Sorted by Priority

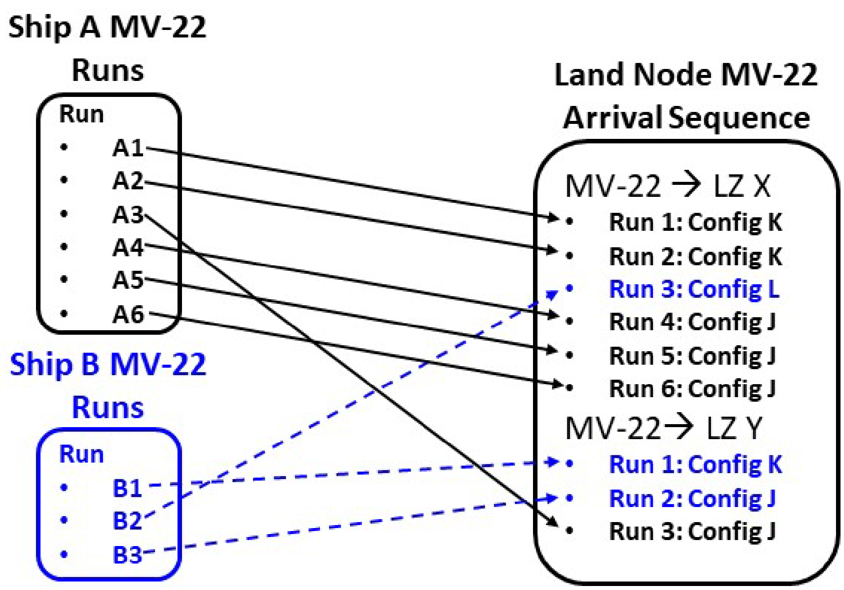



Figure 24. Example of Generating Ship Runs

- Two available ships (A and B) with two landing spots available on each ship
- Five MV-22s available for tasking (three on A and two on B)
- Two landing zones (*X* and *Y*) with two landing spots each
  1. 20-minute travel time from Ship A to LZ *X*
  2. 30-minute travel time from Ship A to LZ *Y*
  3. 25-minute travel time from Ship B to LZ *X*
  4. 15-minute travel time from Ship B to LZ *Y*
- Three different configurations (J, K, and L) available for the air connector
  1. J has on onload/offload time of 5min
  2. K has an onload/offload time of 15min
  3. L has an onload/offload time of 10 min

The scheduling begins by placing each of the five aircraft in a waiting-to-load aboard the ship status (status 1). This status remains the same until the planner assigns the connector to an LZ. The planner must make the decision as to what aircraft will go to what LZ and what configuration will be selected. We refer to the land run sequence list in Figure 6 to make this decision. The connectors are processed in chronological order according to the "Time" column. To avoid ties, we initialize the first time to a small random number; for illustration purposes these starting times are in 0.1-minute increments, as Table 46 depicts.

Table 46. All Aircraft begin Loading

| Ship | Connector ID | Configuration | Status | Time | Departure Run | Return Run | LZ ID | LZ Run |
|---|---|---|---|---|---|---|---|---|
| A | A1 | - | 1 | 0.1 | - | - | - | - |
| A | A2 | - | 1 | 0.3 | - | - | - | - |
| A | A3 | - | 1 | 0.5 | - | - | - | - |
| B | B1 | - | 1 | 0.2 | - | - | - | - |
| B | B2 | - | 1 | 0.4 | - | - | - | - |

The first connector the planner assigns is connector A1. Both LZ *X* and LZ *Y* have 100% deficit, so LZ *X* is chosen. Referring to the land run sequence in Figure 23, the first run assigned to LZ *X* is configuration K. The planner next assigns B1 to LZ *Y* because LZ *Y* still has 100%. Figure 23 states the first run to LZ *Y* is configuration K. After A1 and B1 are assigned their status updates to two (waiting to unload at LZ). The time updates to the time the connectors arrive at the LZ, which includes loading (15 minutes) and transit (20 minutes for A1 and 15 for B1). Table 47 shows these updates.

Table 47. First MV-22s are Assigned and Depart

| Ship | Connector ID | Configuration | Status | Time | Departure Run | Return Run | LZ ID | LZ Run |
|---|---|---|---|---|---|---|---|---|
| A | A1 | K | 2 | 35.1 | 1 | - | X | - |
| A | A2 | - | 1 | 0.3 | - | - | - | - |
| A | A3 | - | 1 | 0.5 | - | - | - | - |
| B | B1 | K | 2 | 30.2 | 1 | - | Y | - |
| B | B2 | - | 1 | 0.4 | - | - | - | - |

With the first two runs scheduled we refer to Figure 23 to determine how the next two aircraft should be scheduled. A2 and B2 are the next two aircraft to leave the seabase (see Time column in Table 47). Because the ships have two spots, A2 and B2 begin loading at the times in Table 4. To determine where to send A2 and B2, we refer to Figure 23, LZ *X* has relative deficit 5/6 and LZ *Y* has relative deficit 2/3. Consequently, both A2 and B2 are sent to LZ *X*. According to Figure 23, A2 should be configuration K because A2 begins loading before B2. B2 uses configuration L. Similar to the process in Table 47, statuses, times, departure runs and LZ IDs updates for both aircraft in Table 48. Of note, aircraft A1, A2, and B2 arrive almost simultaneously to LZ *X*; however, since there are only two landing spots per LZ, aircraft B2 must remain in a loiter status.

Table 48. Next Two MV-22s are Scheduled and Depart

| Ship | Connector ID | Configuration | Status | Time | Departure Run | Return Run | LZ ID | LZ Run |
|---|---|---|---|---|---|---|---|---|
| A | A1 | K | 2 | 35.1 | 1 | - | X | - |
| A | A2 | K | 2 | 35.3 | 2 | - | X | - |
| A | A3 | - | 1 | 0.5 | - | - | - | - |
| B | B1 | K | 2 | 30.2 | 1 | - | Y | - |
| B | B2 | L | 2 | 35.4 | 2 | - | X | - |

With four of our five aircraft now scheduled and awaiting offload we can now schedule aircraft A3. Aircraft A3 is sent to LZ *Y* because it has a greater relative deficit (2/3) compared to LZ *X* (3/6). According to Figure 23, A3 uses configuration J. With only two loading spots aboard each of the ships, A3 must await the departure of aircraft A1 to begin its loading process. As a result, the time stamp associated with A3 in Table 49 reflects a 15-minute waiting to load, 5-minute load, and 25-minute transit time. All aircraft have now been assigned an LZ and configuration and currently awaiting offload at their respective locations as reflected in Table 49. Additionally, with all aircraft now at their respective LZs we update the “LZ Run” column to reflect the sequence that the aircraft arrive to each location.

Table 49. All MV-22s Depart Shipping

| Ship | Connector ID | Configuration | Status | Time | Departure Run | Return Run | LZ ID | LZ Run |
|---|---|---|---|---|---|---|---|---|
| A | A1 | K | 2 | 35.1 | 1 | - | X | 1 |
| A | A2 | K | 2 | 35.3 | 2 | - | X | 2 |
| A | A3 | J | 2 | 50.1 | 3 | - | Y | 2 |
| B | B1 | K | 2 | 30.2 | 1 | - | Y | 1 |
| B | B2 | L | 2 | 35.4 | 2 | - | X | 3 |

A1, A2, A3 and B1 begin offloading immediately upon arrival as there are two spots at each LZ. We update the time stamp for these four aircraft in Table 50 based on the unload times of the configurations.

Table 50. MV-22s begin Return Movement to Ships

| Ship | Connector ID | Configuration | Status | Time | Departure Run | Return Run | LZ ID | LZ Run |
|---|---|---|---|---|---|---|---|---|
| A | A1 | K | 3 | 50.1 | 1 | - | X | 1 |
| A | A2 | K | 3 | 50.3 | 2 | - | X | 2 |
| A | A3 | J | 3 | 55.1 | 3 | - | Y | 2 |
| B | B1 | K | 3 | 45.2 | 1 | - | Y | 1 |
| B | B2 | L | 2 | 35.4 | 2 | - | X | 3 |

As soon as A1 departs LZ *X* at time 50.1, a landing spot opens, which allows aircraft B2 to land and begins its offload process. After unloading, B2 enters status three and we update the information in Table 51.

Table 51. All Aircraft Complete Offload

| Ship | Connector ID | Configuration | Status | Time | Departure Run | Return Run | LZ ID | LZ Run |
|---|---|---|---|---|---|---|---|---|
| A | A1 | K | 3 | 50.1 | 1 | - | X | 1 |
| A | A2 | K | 3 | 50.3 | 2 | - | X | 2 |
| A | A3 | J | 3 | 55.1 | 3 | - | Y | 2 |
| B | B1 | K | 3 | 45.2 | 1 | - | Y | 1 |
| B | B2 | L | 3 | 60.1 | 2 | - | X | 3 |

All five of our assigned aircraft have now completed their offload at their respective LZs and next return to their respective ship. Once an aircraft returns to a ship, it transitions to status one, as the aircraft is now ready to load again. We update the status and the time stamp to incorporate the return travel time in Table 52. Table 52 also includes the “Return Run,” which specifies the order that the aircraft return to the ship. Aircraft A1 and A2 essentially perform the exact same sortie as they both head from ship A to LZ *X* with configuration K. Once back in status one, the aircraft are ready for further tasking in accordance with the land run sequence list in Figure 23.

Table 52. All Aircraft Returned

| Ship | Connector ID | Configuration | Status | Time | Departure Run | Return Run | LZ ID | LZ Run |
|---|---|---|---|---|---|---|---|---|
| A | A1 | K | 1 | 70.1 | 1 | 1 | X | 1 |
| A | A2 | K | 1 | 70.3 | 2 | 2 | X | 2 |
| A | A3 | J | 1 | 85.1 | 3 | 3 | Y | 2 |
| B | B1 | K | 1 | 60.2 | 1 | 1 | Y | 1 |
| B | B2 | L | 1 | 85.1 | 2 | 2 | X | 3 |

After a connector returns to a ship and transitions back to status one, we save all the information about the run. This corresponds to the data in Table 52. This information allows us to construct the assignment depicted in Figure 25. Below we show a partial construction of this assignment through the first five runs.

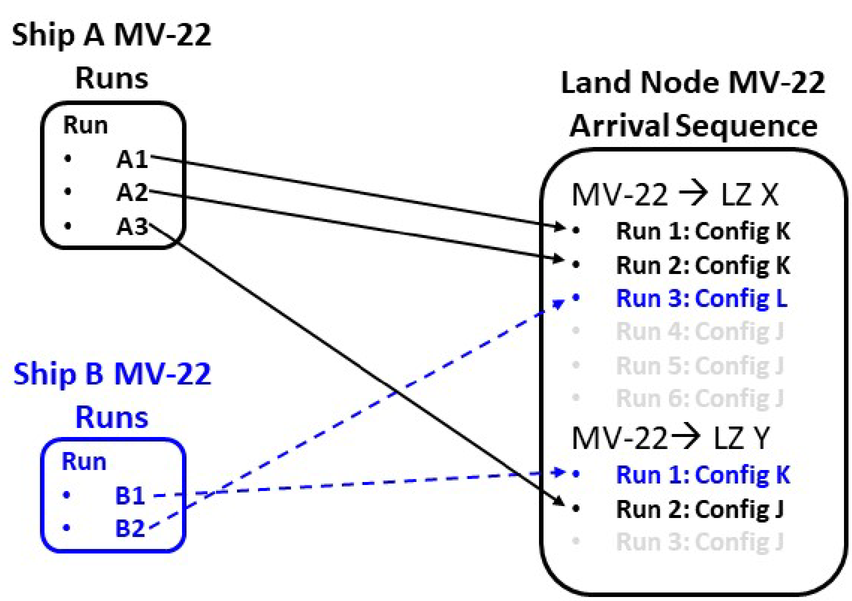



Figure 25. Scheduled Ship Runs from Heuristic Walkthrough

As illustrated, we still have four additional runs to complete. The assignment and scheduling would continue to operate as described above until all runs have been assigned to a ship. For example, connector B1 would be the next connector loaded and assigned to LZ *X* with configuration J as LZ *X* still has 3/6 deficit when compared to LZ *Y* with 1/3. Likewise, aircraft A1 would next be dispatched to LZ *X* next with configuration J for the same rationale. The information depicted in Table 52 (and illustrated in different fashion in Figure 25) is exactly the information that we want to capture to generate both our ship list and ship run sequence list. It is by this naïve approach to organizing and sequencing amphibious connectors that a rough schedule is developed for the entire MEU sustainment plan.

## B. PSEUDO-CODE

This section provides the logic of the Assignment Heuristic. The Assignment Heuristic runs separately for each connector type. The input corresponds to the left-hand side of Figure 23: the number of runs to each land node by configuration. For example, in Figure 23 there are three MV-22J runs to LZ X and one MV-22K runs to LZY. The first step in the Assignment Heuristic is to sort these configuration-runs by priority. The pseudo-code for this appears in Subsection 1. The algorithm used for prioritizing connector configurations is a new addition to the MACS-MC tool and corresponds to the middle portion of Figure 23. The second step of the Assignment Heuristic is the actual assignment of ship runs to beach runs (right-hand side of Figure 23) and remains largely unchanged from Christafore's [3]. This second part appears in Subsection 2.

### 1. Sorting by Priority Pseudo-Code

As discussed above, the input (run_dict) to this function is the left-hand side of Figure 23: the number of runs to each land node by configuration.

\

```
function SORT_RUNS_BY_PRIORITY(run_dict)

    priority={}
    for s in S do

        cur_pri = connector_pri [s]          #get priority for connector from input CSV
        for g in G[s] do
            priority[s].append( [config[s,g,cur_pri],g])     #track config and capacity of
                                                             #priority item
        end for
        priority[s] = sort_by_column(priority[s],1)
                                             #sort by capacity of high priority item
    end for

    prioritized_runs_dict = {}                    #store prioritized list of configs
    for s in S do:                                #loop over all connectors
        for l in L do:                            #loop over all land
    connectors
            if (s, l) in A:                       #only consider valid (s,l) pairs
                for key in priority[s] do:        #loop over prioritized configs
                 g = key[2]                #config is 2nd element in tuple in priority
                 num_runs = run_dict[s][l][g]     #number of total runs by this config
                 cur_list = rep_value(g,num_runs)         #repeat g, num_runs times
                 config_list.append(cur_list)             #append to complete list
                end for
                prioritized_runs_dict[s][l] = config_list
            end if
        end for
    end for

return prioritized_runs_dict
end function
```

### 2. Assignment Heuristic Pseudo-Code

Most of the logic in this subsection remains the same as in Christafore's Appendix B [3]. The one key difference is we need to assign the connector to a land node before loading it because we need to know how long it takes to load.

```
function RUN_ASSIGNMENT_HEURISTIC
    b_runs ←get_beach_runs                                  #from Quickest Flow
    num_con←get_num_connectors                              #from input CSVs
    sb_load ←get_seabase_loadtime                           #from input CSVs
    sb_spots ←get_seabase_spots                             #from input CSVs
    b_unload ←get_beach_unloadtime                          #from input CSVs
    b_spots ←get_beach_spots                                #from input CSVs
    trav_time←get_travel_time                               #from input CSVs
    prioritized_runs_dict ←SORT_RUNS_BY_PRIORITY(run_dict)
                                                            #previous subsection
    runs_config =  prioritized_runs_dict[connector]         #one connector at a time

    sb_obj ←init_seabaseobj(sb_spots,num_con)               #tracks info about each ship
    beach_obj ←init_beachobj(b_spots)                       #tracks info about each beach
    conn_pq ←init_connectors(num_con)             #tracks connector info in Priority Q

    while runs remain do
        next_event = conn_pq.get()            #pull next connector event from Pri Q
        PROC_NEXT_EVENT(b_runs,next_event,sb_load,b_unload,b_spots,
         trav_time,beach_objs,sb_obj,runs_config)
        conn_pq.get().put(next_event)    #put updated connector info back to Pri Q
    end while
end function

function INIT_SEABASEOBJ(sb_spots,num_con)
```

```
    sb_obj =[]                                      #store info on each ship

    for i in 1:num_ships do
                                                         #first element ship id
                                              #2nd element is next run to depart
                                               #3rd element is next run to arrive
                           #4th element is the next time a spot is available to load
        sb_obj[i]←[i,1,(1−num_con[i]),0]
    end for
    return sb_obj
end function

function INIT_BEACHOBJ(b_spots)
    beach_obj =[]                                   #store info on each beach

    for i in1:num_beaches do
                                                        #first element beach id
                                        #2nd element is next run to arrive to beach
                             #3rd element is the next time a spot is available to unload
                       #4th element is number connectors in transit or offloading at beach
                          #5th element is number connectors already assigned to beach
        beach_obj[i]←[i,1,0,0,0]
    end for
    return beach_obj
end function

function INIT_CONNECTORS(num_con)
    conn_pq =Init_PriorityQueue           #store connector info in Priority Queue

    for i in1:num_ships do
```

```
        cur_num←num_con[i]                          #number of connectors on Ship i

        for j in1: cur_num do
                                          #1st element: time associated with current task
                                                             #2nd element: current task
                                                                   #3rd element: ship ID
                                                            #4th element: connector ID
                                                 #5th element: departure ship run
                                             #6th element: beach ID for this run
                                                                 #7th element: beach run
                                                          #8th element: return ship run
                                                         #9th element: configuration

            con_info←[0,1,i, j,−1,−1,−1,−1,-1]
            conn_pq.put(con_info)       #put connector info into Priority Queue
        end for
    end for
    return conn_pq
end function

functionPROC_NEXT_EVENT(b_runs,next_event,sb_load,b_unload,b_spots,trav_time
    beach_obj,sb_obj)                                      #additional function inputs
    cur_task ←next_event[2]                        #2nd element in connector info list

    if cur_task ==1 then
        process_sb_event(next_event,sb_load,b_runs,
        b_unload,b_spots,trav_time,
        sb_obj,beach_obj, runs_config)
    else if cur_task ==3 then
        process_b_event(next_event,b_unload,beach_obj)
```

```
    else
        process_return_to_sb(next_event,trav_time,beach_obj)
    end if
end function

function PROCESS_SB_EVENT(next_event,sb_load,b_runs,
b_unload,b_spots,trav_time,sb_obj,beach_obj, runs_config)
                                                        #additional function inputs
    save_completed_run          #before initializing next run, save previous run info
    arrival_time←next_event[1]
    ship_id ←next_event[3]
    next_spot ←get_next_free_sbspot(sb_obj,ship_id)     #which load spot next free
    time_spot_avail ←get_time_next_free_sbspot(sb_obj,ship_id)
                                                        #time next spot free
    start_time←max(arrival_time,time_spot_avail)     #time connector starts loading
    next_beach←assign_next_beach(b_runs,beach_obj)          #which beach to go to
    cfg_id← next_cfg(next_beach, config_runs, beach_obj)   #next cfg to given beach
    splash_time← start_time + sb_load[ship_id][cfg_id]   #time connector leaves ship
    beach_time← splash_time + trav_time[ship_id][next_beach]
                                                        #time arrive at beach

    next_event[2]←3                   #new connector task: ready to unload at beach
    next_event[1]← beach_time                   #new connector time: arrive at beach

    next_event[6]←next_beach                          #store beach assigned to run
    next_event[5]←ship_depart_run(sb_obj,ship_id)                 #get run number
    UPDATE_SB_OBJECT(sb_obj,ship_id)          #spot availability, number of runs
    UPDATE_BEACH_OBJECT(beach_obj,beach_id)
                                                #number of runs assigned to beach
end function
```

```
function PROCESS_B_EVENT(next_event,b_unload,beach_obj)
        arrival_timenext_event[1]
        beach_id ←next_event[6]
        cfg_id←next_event[9]
        next_spot ←get_next_free_bspot(beach_obj,beach_id)
                                                        #which load spot next free
        time_avail ←get_time_next_free_bspot(beach_obj,beach_id)
                                                        #time next spot free
        start_time←max(arrival_time,time_avail)        #time connector starts unloading
        splash_time← start_time + beach_unload[beach_id][cfg_id]
                                                        #time connector leaves beach
        next_event[2]←4                                 #new connector task: returning to ship
        next_event[1]← splash_time                      #new connector time: departure from beach
        next_event[7]←beach_run_num(beach_obj,beach_id)          #beach run number
        UPDATE_BEACH_OBJ (beach_obj,beach_id,splash_time)
                                                        #spot availability, number of runs
end function

function PROCESS_RETURN_TO_SB(next_event,trav_time,beach_obj)
        splash_time←next_event[1]
        ship_id ←next_event[3]
        beach_id ←next_event[6]
        return_time← splash_time + trav_time[ship_id][beach_id]       #time back at ship
        next_event[2]←1                                 #new connector task: ready to load at ship
        next_event[1]←return_time                       #new connector time: arrival back to ship
        update_beach_object(beach_obj,beach_id)         #one fewer connector at beach
end function

function ASSIGN_NEXT_BEACH(b_runs,beach_obj)
```

```
    max_deficit ←0                    #assign next run to beach with largest run deficit
    next_beach←−1                                                        #ID of next beach

    for i in1:num_beaches do
        tot_runs ← b_runs[i]                                         #total runs required
        runs_assign← beach_obj[i][5]                            #runs assigned to beach
        cur_deficit ←tot_runs−runs_assign                               #remaining runs

        if cur_deficit > max_deficit then                    #found new best candidate
            max_deficit ←cur_deficit
            next_beach←i
        end if
    end for
    return best_beach
end function
```

# APPENDIX D. CHAPTER V BASE SCENARIO PROMPT

The following appendix consists of an assignment prompt the author completed for one of his Masters courses and was the baseline for the scenario in Chapter V. The prompt, in its entirety, was taken from the Naval Postgraduate School's Joint Campaign Analysis Course (OA4602) winter 2018 segment, instructed by Professor Jeffery Kline (CAPT USN, RET) [16]. As the prompt demonstrates, the assignment required the development of a scheme of maneuver for an amphibious assault on an island in the western Pacific. The MACS-MC was used to provide insight and determine how best to conduct the amphibious assault based on delivery times and overall demand satisfaction per commodity. This same scenario was then modified and expanded to form the base scenario and subsequent sensitivity analysis detailed in Chapter V.

## A. JOINT CAMPAIGN ANALYSIS TEAM MINI STUDY PROMPT

Analytical Tasking for Maritime War of 2030

Joint Campaign Analysis

Winter 2018

Operating Environments and Assumptions

### 1. South China Sea Current Situation Year 2032 Full Combat

The war started late in 2030 with China's rapid and successful occupation of Natuna Besar, Indonesia and Palawan, Philippines. No immediate response was offered by either country, except for appeals to the United Nations, and in the Philippine's case, directly to the United States. Chinese PLAN and Maritime Patrol ships began to stop and inspect all merchant traffic through the South China Sea, which brought demarche protests from Japan, South Korea, Australia, Singapore, Vietnam, and the United States. The United States, honoring the mutual defense treaty with the Philippines and Japan, and entering a defense agreement with the other protesting countries, began stopping and inspecting Chinese flag ships world-wide in proportional response while mobilizing forces. During one such inspection in the Philippine Sea, a U.S. DDG was torpedoed by an unknown submarine. War was declared by all participants. North Korea allied itself with China.

The conflict quickly evolved to a maritime war of attrition with China's sea control threatened by allied submarines inside the first island chain, and allied sea control threatened by PLAN submarines, ballistic missiles, and cruise missiles around and outside the first island chain.

Now, in early 2032, all sides have lost from 10–15% of their submarines, ships, aircraft, and crew. Weapon inventories are down to 50% for allies and 70% for PLAN and North Korea. Although under threat of ballistic missile attack, allied expeditionary air fields are operating in the area of operations from Dong Tac, Vietnam; Kumejima Airport in Japan; Clark airfield in the Philippines; Singapore; Nangapinoh airfield, Borneo, Indonesia. Allied submarine forces have focused on intercepting and sinking re-supply convoys to Natuna Besar and Palawan for the past eight months. Long range strikes have occurred against PLA forces on both islands. The allies are now planning on moving inside the first island chain to retake first Natuna Besar and then Palawan. PLAN submarines and long range strike bombers still threaten allied sea lines of communications.

### 2. Marine Corps Forces, Pacific

Given the Blue forces above are fixed, develop a concept for retaking Natuna Besar. Consider a distributed MAGTF concept to challenge enemy ISR (http://www.mccdc.marines.mil/MOC/Primacy-Maneuver-Warfare/MAGTF/). Start with the tactical scenario which is provided.

Since taking Natuna Besar with the 1st Marine Brigade, the Chinese have reinforced the island with four to eight YJ-62 mobiles surface missile systems, two to four SA-500 surface to air missile systems, and 15 J-20 fighter aircraft. Per the scenario above, an allied undersea campaign against Chinese logistic lines and long-range strike campaign against the islands is ongoing. For your concept assess:

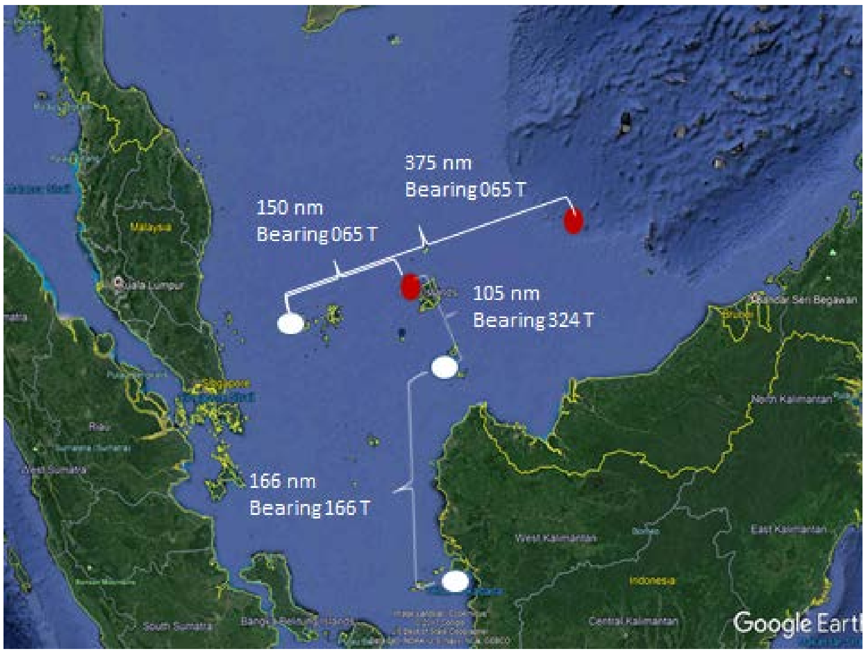
1. The required effectiveness level of the strike and isolation campaigns pre-D day

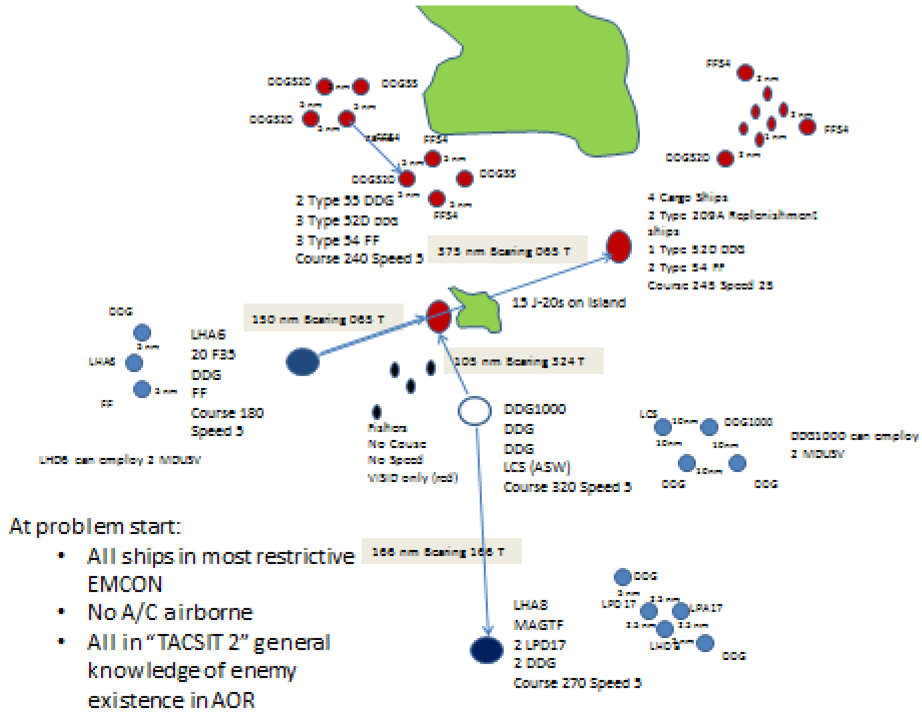
2. Role, timeliness and effectiveness of MAGTF (MAW) and SoF to locate, target, and destroy or neuter YJ-62, SA500, and J20 airfield.

3. Role and assessment of operational and tactical deception and decoys in the operation

4. The ground battle risk

Tactical force laydown situation for Natuna Besar in 2032




2 Type 55 DDG
3 Type 52D ddg
3 Type 54 FF
Course 240 Speed 5
375 nm Bearing 065 T
4 Cargo Ships
2 Type 209A Replenishment ships
1 Type 520 DDG
2 Type 54 FF
Course 245 Speed 25
15 J-20s on Island
150 nm Bearing 065 T
LHA6
20 F35
DDG
FF
Course 180
Speed 5
LHD8 can employ 2 MDUSV
105 nm Bearing 324 T
Fishers
No Course
No Speed
VISID only (red)
DDG1000
DDG
DDG
LCS (ASW)
Course 320 Speed 5
DDG1000 can employ 2 MDUSV
At problem start:
• All ships in most restrictive EMCON
• No A/C airborne
• All in "TACSIT 2" general knowledge of enemy existence in AOR
166 nm Bearing 166 T
LHA8
MAGTF
2 LPD17
2 DDG
Course 270 Speed 5

# APPENDIX E. CHAPTER V INPUTS

In this Appendix, we provide all the inputs used in Chapter V to form our base scenario. Each table has been drawn directly from the CSVs and represents the same process a user would follow to run the MACS-MC. The remainder of this Appendix provides a detailed explanation of the 10 separate CSVs a user must complete to create and run an amphibious scenario.

## A. COMMODITIES AND UNITS

The first CSV the user must complete enumerates the different commodities being modeled in the scenario (Table 53). Here we include the nine different commodities used in Chapter V and list their respective units of measure.

Table 53. Commodities and their Units of Measure

| type | units |
|---|---|
| FUEL | ThousandGallons |
| WATER | ThousandGallons |
| AMMO1 | Pallets |
| PAX | People |
| HMMWV | Vehicles |
| AMMO2 | Pallets |
| MRE | Pallets |
| LAV | LAV |
| TANK | Tank |

## B. CONFIGURATIONS

Each of the amphibious and land connectors must have all their configurations enumerated and the user must define the quantity of each commodity that is capable of being transported simultaneously. Additionally, the loading and offloading time requirements associated with each configuration must be included. Table 54 presents a complete listing of all the configurations used in Chapter V, with no adjustments being made during the analysis.

Table 54. Chapter V Configurations

| type | configurati | FUEL | WATER | AMMO1 | PAX | HMMWV | AMMO2 | LAV | TANK | MRE | loadTime | unloadTime |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| MV22 | A | 0.5 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 15 | 5 |
| MV22 | B | 0 | 0.5 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 15 | 5 |
| MV22 | C | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 20 | 10 |
| MV22 | D | 0 | 0.2 | 2 | 0 | 0 | 0 | 0 | 0 | 0 | 20 | 15 |
| MV22 | E | 0 | 0.2 | 0 | 0 | 0 | 2 | 0 | 0 | 0 | 20 | 15 |
| MV22 | F | 0 | 0.2 | 0 | 12 | 0 | 0 | 0 | 0 | 0 | 20 | 10 |
| MV22 | G | 0 | 0.2 | 0 | 0 | 0 | 0 | 0 | 0 | 2 | 20 | 15 |
| MV22 | H | 0 | 0.3 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 20 | 15 |
| MV22 | I | 0 | 0.1 | 0 | 0 | 0 | 0 | 0 | 0 | 3 | 20 | 15 |
| MV22 | J | 0 | 0.1 | 0 | 18 | 0 | 0 | 0 | 0 | 0 | 10 | 5 |
| MV22 | K | 0 | 0.3 | 0 | 6 | 0 | 0 | 0 | 0 | 0 | 20 | 15 |
| MV22 | L | 0 | 0 | 0 | 18 | 0 | 0 | 0 | 0 | 1 | 10 | 5 |
| MV22 | M | 0 | 0 | 0 | 6 | 0 | 0 | 0 | 0 | 3 | 15 | 10 |
| MV22 | N | 0 | 0 | 0 | 24 | 0 | 0 | 0 | 0 | 0 | 5 | 5 |
| MV22 | O | 0 | 0 | 1 | 18 | 0 | 0 | 0 | 0 | 0 | 10 | 5 |
| MV22 | P | 0 | 0 | 2 | 12 | 0 | 0 | 0 | 0 | 0 | 15 | 10 |
| MV22 | Q | 0 | 0 | 3 | 6 | 0 | 0 | 0 | 0 | 0 | 20 | 15 |
| MV22 | R | 0 | 0 | 0 | 18 | 0 | 1 | 0 | 0 | 0 | 10 | 5 |
| MV22 | S | 0 | 0 | 0 | 12 | 0 | 2 | 0 | 0 | 0 | 15 | 10 |
| MV22 | T | 0 | 0 | 0 | 6 | 0 | 3 | 0 | 0 | 0 | 20 | 15 |
| MV22 | U | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 4 | 25 | 20 |
| CH53 | A | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 15 | 5 |
| CH53 | B | 0 | 0 | 0 | 36 | 0 | 0 | 0 | 0 | 0 | 10 | 5 |
| CH53 | C | 0 | 0 | 0 | 27 | 0 | 0 | 0 | 0 | 2 | 15 | 10 |
| CH53 | D | 0 | 0 | 0 | 18 | 0 | 0 | 0 | 0 | 4 | 20 | 15 |
| CH53 | E | 0 | 0 | 0 | 9 | 0 | 0 | 0 | 0 | 6 | 25 | 20 |
| CH53 | F | 0 | 0 | 2 | 27 | 0 | 0 | 0 | 0 | 0 | 20 | 15 |
| CH53 | G | 0 | 0 | 4 | 18 | 0 | 0 | 0 | 0 | 0 | 25 | 25 |
| CH53 | H | 0 | 0 | 6 | 9 | 0 | 0 | 0 | 0 | 0 | 35 | 30 |
| CH53 | I | 0 | 0.2 | 0 | 27 | 0 | 0 | 0 | 0 | 0 | 25 | 20 |
| CH53 | J | 0 | 0.4 | 0 | 18 | 0 | 0 | 0 | 0 | 0 | 30 | 25 |
| CH53 | K | 0 | 0.6 | 0 | 9 | 0 | 0 | 0 | 0 | 0 | 40 | 30 |
| CH53 | L | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 8 | 35 | 30 |
| CH53 | M | 0 | 0.2 | 0 | 0 | 0 | 0 | 0 | 0 | 6 | 35 | 30 |
| CH53 | N | 0 | 0.4 | 0 | 0 | 0 | 0 | 0 | 0 | 4 | 35 | 30 |
| CH53 | O | 0 | 0.6 | 0 | 0 | 0 | 0 | 0 | 0 | 2 | 35 | 30 |
| CH53 | P | 0 | 0 | 2 | 0 | 0 | 0 | 0 | 0 | 6 | 35 | 30 |
| CH53 | Q | 0 | 0 | 4 | 0 | 0 | 0 | 0 | 0 | 4 | 35 | 30 |
| CH53 | R | 0 | 0 | 6 | 0 | 0 | 0 | 0 | 0 | 2 | 35 | 30 |
| CH53 | S | 0 | 0 | 8 | 0 | 0 | 0 | 0 | 0 | 0 | 35 | 15 |
| CH53 | T | 0 | 0 | 0 | 0 | 0 | 8 | 0 | 0 | 0 | 30 | 15 |
| CH53 | U | 0.5 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 15 | 10 |
| LCAC | A | 0 | 0 | 0 | 125 | 0 | 0 | 0 | 0 | 0 | 40 | 20 |
| LCAC | B | 0 | 0 | 0 | 25 | 0 | 0 | 0 | 1 | 0 | 35 | 5 |
| LCAC | C | 0 | 0 | 0 | 25 | 0 | 0 | 6 | 0 | 0 | 45 | 10 |
| LCAC | D | 0 | 0 | 0 | 25 | 3 | 0 | 3 | 0 | 0 | 45 | 10 |
| LCAC | E | 0 | 0 | 0 | 25 | 0 | 0 | 0 | 0 | 40 | 35 | 15 |
| LCAC | F | 0 | 0 | 0 | 25 | 0 | 10 | 0 | 0 | 20 | 35 | 15 |
| LCAC | G | 0 | 0 | 10 | 25 | 0 | 0 | 0 | 0 | 20 | 35 | 15 |
| LCAC | H | 0 | 2 | 0 | 25 | 0 | 0 | 0 | 0 | 20 | 35 | 15 |
| LCAC | I | 2 | 0 | 0 | 25 | 0 | 0 | 0 | 0 | 20 | 35 | 15 |
| LCAC | J | 0 | 0 | 0 | 25 | 2 | 0 | 0 | 0 | 20 | 40 | 20 |
| LCAC | K | 0 | 4 | 0 | 25 | 0 | 0 | 0 | 0 | 0 | 35 | 20 |
| LCAC | L | 0 | 2 | 20 | 25 | 0 | 0 | 0 | 0 | 0 | 40 | 25 |
| LCAC | M | 0 | 2 | 0 | 25 | 2 | 0 | 0 | 0 | 0 | 35 | 20 |
| LCAC | N | 0 | 2 | 0 | 25 | 0 | 20 | 0 | 0 | 0 | 40 | 25 |
| LCAC | O | 2 | 2 | 0 | 25 | 0 | 0 | 0 | 0 | 0 | 35 | 30 |
| LCAC | P | 4 | 0 | 0 | 25 | 0 | 0 | 0 | 0 | 0 | 30 | 15 |
| LCAC | Q | 2 | 0 | 0 | 25 | 2 | 0 | 0 | 0 | 0 | 35 | 25 |
| LCAC | R | 0 | 0 | 0 | 25 | 0 | 4 | 0 | 0 | 0 | 30 | 15 |
| LCAC | S | 0 | 0 | 4 | 25 | 0 | 0 | 0 | 0 | 0 | 30 | 15 |
| LCAC | T | 0 | 0 | 0 | 25 | 11 | 0 | 0 | 0 | 0 | 45 | 10 |
| LCU | A | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 2 | 0 | 45 | 10 |
| LCU | B | 0 | 0 | 0 | 0 | 0 | 0 | 3 | 1 | 0 | 45 | 15 |
| LCU | C | 0 | 0 | 0 | 0 | 0 | 10 | 0 | 1 | 0 | 45 | 15 |

| | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| LCU | D | 0 | 0 | 0 | 0 | 4 | 0 | 0 | 1 | 0 | 45 | 15 |
| LCU | E | 0 | 0 | 0 | 50 | 0 | 0 | 0 | 1 | 0 | 45 | 15 |
| LCU | F | 0 | 0 | 10 | 0 | 0 | 0 | 0 | 1 | 0 | 45 | 15 |
| LCU | G | 0 | 2 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 45 | 15 |
| LCU | H | 2 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 45 | 20 |
| LCU | I | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 20 | 45 | 20 |
| LCU | J | 0 | 0 | 0 | 0 | 0 | 0 | 6 | 0 | 0 | 30 | 5 |
| LCU | K | 0 | 0 | 0 | 0 | 0 | 0 | 3 | 0 | 20 | 30 | 10 |
| LCU | L | 0 | 0 | 0 | 0 | 0 | 10 | 3 | 0 | 0 | 30 | 15 |
| LCU | M | 0 | 0 | 0 | 0 | 6 | 0 | 3 | 0 | 0 | 30 | 15 |
| LCU | N | 0 | 0 | 0 | 50 | 0 | 0 | 3 | 0 | 0 | 30 | 15 |
| LCU | O | 0 | 0 | 10 | 0 | 0 | 0 | 3 | 0 | 0 | 30 | 15 |
| LCU | P | 0 | 2 | 0 | 0 | 0 | 0 | 3 | 0 | 0 | 30 | 15 |
| LCU | Q | 0 | 0 | 0 | 0 | 12 | 0 | 0 | 0 | 0 | 45 | 15 |
| LCU | R | 0 | 0 | 0 | 0 | 6 | 0 | 0 | 0 | 20 | 35 | 10 |
| LCU | S | 0 | 0 | 0 | 50 | 6 | 0 | 0 | 0 | 0 | 35 | 10 |
| LCU | T | 0 | 0 | 10 | 0 | 6 | 0 | 0 | 0 | 0 | 35 | 10 |
| LCU | U | 0 | 0 | 0 | 0 | 6 | 10 | 0 | 0 | 0 | 35 | 10 |
| LCU | V | 2 | 0 | 0 | 0 | 6 | 0 | 0 | 0 | 0 | 35 | 10 |
| LCU | W | 0 | 0 | 0 | 300 | 0 | 0 | 0 | 0 | 0 | 40 | 15 |
| LCU | X | 2 | 0 | 0 | 150 | 0 | 0 | 0 | 0 | 0 | 35 | 15 |
| LCU | Y | 0 | 2 | 0 | 150 | 0 | 0 | 0 | 0 | 0 | 35 | 15 |
| LCU | Z | 0 | 0 | 10 | 150 | 0 | 0 | 0 | 0 | 0 | 35 | 15 |
| LCU | AA | 0 | 0 | 0 | 150 | 0 | 10 | 0 | 0 | 0 | 35 | 15 |
| LCU | BB | 2 | 1 | 0 | 75 | 0 | 0 | 0 | 0 | 0 | 35 | 15 |
| LCU | CC | 1 | 2 | 0 | 75 | 0 | 0 | 0 | 0 | 0 | 35 | 15 |
| LCU | DD | 1 | 1 | 0 | 75 | 0 | 0 | 0 | 0 | 20 | 35 | 15 |
| LCU | EE | 0 | 2 | 0 | 75 | 0 | 0 | 0 | 0 | 20 | 35 | 15 |
| LCU | FF | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 40 | 25 | 15 |
| LCU | GG | 2 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 40 | 25 | 15 |
| LCU | HH | 0 | 2 | 0 | 0 | 0 | 0 | 0 | 0 | 40 | 25 | 15 |
| LCU | II | 3 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 20 | 25 | 15 |
| LCU | JJ | 0 | 3 | 0 | 0 | 0 | 0 | 0 | 0 | 20 | 25 | 15 |
| LCU | KK | 2 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 20 | 25 | 15 |
| LCU | LL | 1 | 2 | 0 | 0 | 0 | 0 | 0 | 0 | 20 | 25 | 15 |
| LCU | MM | 0 | 1 | 10 | 0 | 0 | 0 | 0 | 0 | 20 | 25 | 15 |
| LCU | NN | 0 | 0 | 20 | 0 | 0 | 0 | 0 | 0 | 20 | 25 | 15 |
| LCU | OO | 0 | 3 | 10 | 0 | 0 | 0 | 0 | 0 | 0 | 25 | 15 |
| LCU | PP | 0 | 2 | 20 | 0 | 0 | 0 | 0 | 0 | 0 | 25 | 15 |
| LCU | QQ | 0 | 1 | 0 | 0 | 0 | 10 | 0 | 0 | 20 | 25 | 15 |
| LCU | RR | 0 | 0 | 0 | 0 | 0 | 20 | 0 | 0 | 20 | 25 | 15 |
| LCU | SS | 0 | 3 | 0 | 0 | 0 | 10 | 0 | 0 | 0 | 25 | 15 |
| LCU | TT | 0 | 2 | 0 | 0 | 0 | 20 | 0 | 0 | 0 | 25 | 15 |
| CLP | A | 3 | 2 | 10 | 32 | 4 | 10 | 6 | 4 | 20 | 120 | 40 |

## C. CONNECTOR INFORMATION

The connector CSV allows the user to define the velocity and commodity priority per connector. While the information in Table 55 is specific to the scenario used in Chapter V, there is nothing preventing the user from employing different connector types so long as their required information is included in the configurations, seabase information, and land node information CSVs. The user must indicate the category the connector belongs to (surface, air, or land) so that the MACS-MC can make the appropriate edges within the network and define the velocity that each connector travels. Finally, the user must designate what commodity is a priority for each connector. MACS-MC uses these priorities to select those configurations that maximize the delivery of the priority item per commodity until its demand has been satisfied ashore or no connector runs remain. This information (Table 55) remains unchanged throughout Chapter V.

Table 55. Connector Information

| type | category | velocity | priority |
|---|---|---|---|
| LCAC | surface | 30 | LAV |
| LCU | surface | 8 | TANK |
| CH53 | air | 140 | WATER |
| MV22 | air | 200 | MRE |
| CLP | land | 15 | FUEL |

## D. GENERAL PARAMETERS

The general parameters tab provides the user with the basic bounds used to run the MACS-MC. These parameters and their function are described in detail in Chapter III and their values used in Chapter V are presented in Table 56. dayLength corresponds to the length of time the well-deck and flight-deck can operate, which directly impacts how many connector runs can be sent ashore. The missionWindow dictates the time set T. In this case we want to satisfy as much demand for commodities in a 24-hour period. mipGap and mipTimeMin are parameters associated with the mixed integer linear program formulation of Quickest Flow model in Chapter III.A.4 that ensure the algorithm terminates in a reasonable amount of time (although possibly with a poor solution). When we perform analysis on increasing the operational window by five hours we adjust dayLength to equal 15.

Table 56. Chapter V General Parameters

| | |
|---|---|
| dayLength | 10 |
| timeStepsPerHour | 4 |
| alpha | 0.99999 |
| beta | 0.00001 |
| missionWindow | 24 |
| mipTimeMin | 10 |
| mipGap | 0.001 |

## E. LAND CAPACITY

This CSV provides the upper bound of supplies that can be stored at any land node per commodity at any given time period *t*. Values must be in the same units enumerated in Table 53. For more information regarding this parameter see Chapter III.A. The information presented in Table 57 remains the same throughout Chapter V.

Table 57. Maximum Capacity per Commodity per Node

| id | FUEL | WATER | AMMO1 | PAX | HMMWV | AMMO2 | LAV | TANK | MRE |
|---|---|---|---|---|---|---|---|---|---|
| B1 | 10 | 3 | 20 | 100 | 20 | 15 | 14 | 12 | 20 |
| B2 | 5 | 2 | 15 | 100 | 10 | 8 | 8 | 6 | 10 |
| B3 | 1 | 0.5 | 5 | 100 | 10 | 4 | 8 | 6 | 5 |
| LZ1 | 0.5 | 2 | 5 | 100 | 2 | 2 | 8 | 6 | 3 |
| LZ2 | 0.5 | 1 | 5 | 100 | 2 | 4 | 8 | 6 | 3 |
| LZ3 | 0.5 | 1 | 5 | 100 | 2 | 0 | 8 | 6 | 1 |

## F. LAND DEMAND

All shore base demand must be entered by commodity and node (Table 58) and the unit values must match those enumerated in Table 53. MACS-MC schedules connectors by configuration based on the most efficient means available to satisfy this demand. This information remains the same throughout the Chapter V analysis.

Table 58. Chapter V Demand

| id | FUEL | WATER | AMMO1 | PAX | HMMWV | AMMO2 | LAV | TANK | MRE |
|---|---|---|---|---|---|---|---|---|---|
| B1 | 8 | 1.5 | 15 | 75 | 8 | 10 | 12 | 10 | 10 |
| B2 | 3 | 1 | 10 | 30 | 2 | 4 | 6 | 4 | 6 |
| B3 | 0.5 | 0.5 | 4 | 30 | 8 | 1 | 6 | 4 | 4 |
| LZ1 | 0 | 1 | 4 | 100 | 2 | 0 | 0 | 0 | 3 |
| LZ2 | 0 | 0.6 | 2 | 50 | 2 | 2 | 0 | 0 | 3 |
| LZ3 | 0 | 0.6 | 2 | 24 | 4 | 0 | 0 | 0 | 1 |

## G. LAND EDGE INFORMATION

The land edge information data is used by the MACS-MC to build the shore-based portion of the amphibious network. The user inputs the distances between the different land nodes to create the edges that CLPs use to push demand between nodes. To represent two-way traffic along a land-based edge, the user would input the start and ending location as well as their inverse. For example, in Table 59 the B1 to LZ1 edge is a one-way route. If it were a two-way road a second row of information would be entered underneath as LZ1 to B1 with the prescribed distance listed. This information was updated in Chapter V.D.3 to create our CLP adjustment scenario. In that scenario we made routes from LZ1 to B1, and LZ to B2, B3, and LZ3.

Table 59. Chapter V Land Edge Information

| start | end | distance |
|---|---|---|
| B1 | LZ1 | 1.16 |
| B2 | LZ2 | 2.4 |
| B2 | B3 | 5.77 |
| B2 | LZ3 | 6.95 |

## H. LAND NODE INFORMATION

The land node information CSV details the total number of connector spots located by node where a platform can land to offload. This information is used during the Scheduler Linear Program to help determine if a landing spot is available at a land node before dispatching a run. Additionally, this CSV also provides the user the means of detailing where each, or any, of the networks CLPs are located. For example, if we observe B1 in

Table 60, we see that it has one CLP capable of pushing supplies to additional nodes, three LCAC positions and no space for any air connectors. In Chapter V we modify this table by updating the two CLPs to be at LZ1 and LZ2 when analyzing the effects of different CLP placement (Chapter V.D.3).

Table 60. Chapter V Land Node Information

| id | CLP | LCACSpots | LCUSpots | CH53Spots | MV22Spots |
|---|---|---|---|---|---|
| B1 | 1 | 3 | 0 | 0 | 0 |
| B2 | 1 | 0 | 1 | 0 | 0 |
| B3 | 0 | 1 | 1 | 0 | 0 |
| LZ1 | 0 | 0 | 0 | 3 | 4 |
| LZ2 | 0 | 0 | 0 | 2 | 3 |
| LZ3 | 0 | 0 | 0 | 1 | 1 |

## I. SEABASE INFORMATION

The seabase information CSV provides a very similar purpose as Table 60. It identifies the location for each of the amphibious connectors aboard our ARG shipping, as well as the number of loading locations each one has. If we take for example the LSD in Table 61 we can see that it has two LCACs, two CH-53s and two MV-22s. Furthermore, it has the landing spots to load one LCAC, one LCU, one CH-53 and two MV-22s. This information is used during the Scheduler Linear Program to decide which aircraft depart from which ship and when and where they will load depending on loading spot availability. This information remains unchanged throughout Chapter V.

Table 61. Chapter V Seabase Information

| id | LCAC | LCU | CH53 | MV22 | LCACSpots | LCUSpots | CH53Spots | MV22Spots |
|---|---|---|---|---|---|---|---|---|
| LHD | 3 | 0 | 2 | 8 | 2 | 1 | 2 | 4 |
| LPD | 0 | 1 | 1 | 2 | 1 | 1 | 1 | 2 |
| LSD | 2 | 0 | 2 | 2 | 1 | 1 | 1 | 2 |

## J. SEABASE TO LAND INFORMATION

The seabase to land information serves the same purpose as the land edge information CSV in that it builds the sea to land portion of the amphibious network. As enumerated in Table 62, distances from each of the amphibious ships to each of the land nodes must be inputted. This information forms the seabase to land edges that our surface and air connectors use to conduct their deliveries. This information is modified in Chapter V.D.1 by subtracting approximately 10 NM from each value.

Table 62. Chapter V Seabase to Land Information

| | B1 | B2 | B3 | LZ1 | LZ2 | LZ3 |
|---|---|---|---|---|---|---|
| LHD | 25 | 25 | 26 | 26 | 27 | 26 |
| LPD | 29 | 28 | 26 | 30 | 27 | 25 |
| LSD | 24 | 25 | 27 | 25 | 26 | 28 |

# APPENDIX F. CASE STUDY: OPERATION STARLIGHT

Our objective for this appendix is to provide a comparison between a historical example of an amphibious landing and the output produced by our MACS-MC tool. Up to this point we demonstrate how MACS-MC performs when using notional scenarios but have yet to demonstrate how these results compare to actual landings. Using historical data that details the planned landing schedule and the actual completion time we demonstrate that the MACS-MC tool accurately projects the landing timetable, while presumably reducing planning time requirements. We use the 1965 amphibious landing in Vietnam, known as OPERATION Starlight, as our historical example for comparison.

## A. BACKGROUND ON OPERATION STARLIGHT

OPERATION Starlight is considered the first major offensive operation undertaken during the Vietnam War by American forces against the North Vietnamese Army [17]. The operation was executed by Regimental Combat Team 7 (RCT-7) and involved several Marine Corps infantry battalions in what is called a classic hammer and anvil operation. These operations involve a dedicated blocking force to prevent an enemy's escape, while several other friendly forces maneuver to trap and destroy the enemy between their advance and the blocking force. One of these maneuver forces dedicated to OPERATION Starlight was 3rd Battalion 3rd Marines (V33), which landed in the objective area from amphibious shipping [17].

The night prior to their assault, V33 arrived via three LSDs to the amphibious operations area where they would conduct their assault on the Vietnamese coast on the morning of 18 August 1965 (D-Day) at 0630 (H-Hour) [17]. The general scheme of maneuver entailed the position of the three ship ARG one mile off the coast and moving the force ashore in seven sequential waves [18]. In preparation for this landing V33 created a landing plan consisting of seven waves of multiple connector runs. We provide the specific land plan of each of these seven waves in the next section. For the remainder of this section, we discuss more generally what was transported ashore in those waves.

The first wave consisted of two companies loaded aboard amphibious tracked vehicles (ATV) for the beach assault. Amphibious tracked vehicles are a fifth form of amphibious connector that our model does not currently incorporate. These vehicles are armored personnel carriers that travel from the ship to shore with varying numbers of infantry personnel in the cargo space (nine-16). They are supplemented with medium and heavy machine guns to provide both fires when coming ashore, and to support dismounted infantry once they reach land [14]. The current ATV in the Marine Corps arsenal is the Amphibious Assault Vehicle (AAV) [14].

The subsequent six waves were scheduled to come ashore via the three LCUs organic to the ARG; each LSD in the ARG contained one LCU. These waves consisted of:

- an additional rifle company (Co. L)
- six tanks
- three flame throwing tanks
- four ONTOS (M50A1 self-propelled multiple recoilless rifle system)
- an 81mm mortar platoon
- Headquarters and Support (H&S) Company (reduced)

The total planned time for the landing was 75 minutes after H-Hour or culminating at 0745 local time. Based on information from V33's after action report of the operation we know that the first wave landed on time at H-Hour (0630) with the final forces in Wave seven coming ashore at H+60 (0730 local) [18].

## B. MODELING OPERATION STARLIGHT

For our analysis we ignore the first wave of forces coming ashore in the ATVs. The MACS-MC tool was built primarily as a sustainment planner and does not account for forces departing amphibious shipping via ATVs, an item that could be a point for future research. We then model the amphibious network of seabase and land nodes as depicted in Figure 26. This was a relatively simple network to create as it consisted of three ships of the same class, all operating one mile from the beach, and there was only one land node, Green Beach, with three LCU landing spots (one LCU per ship).

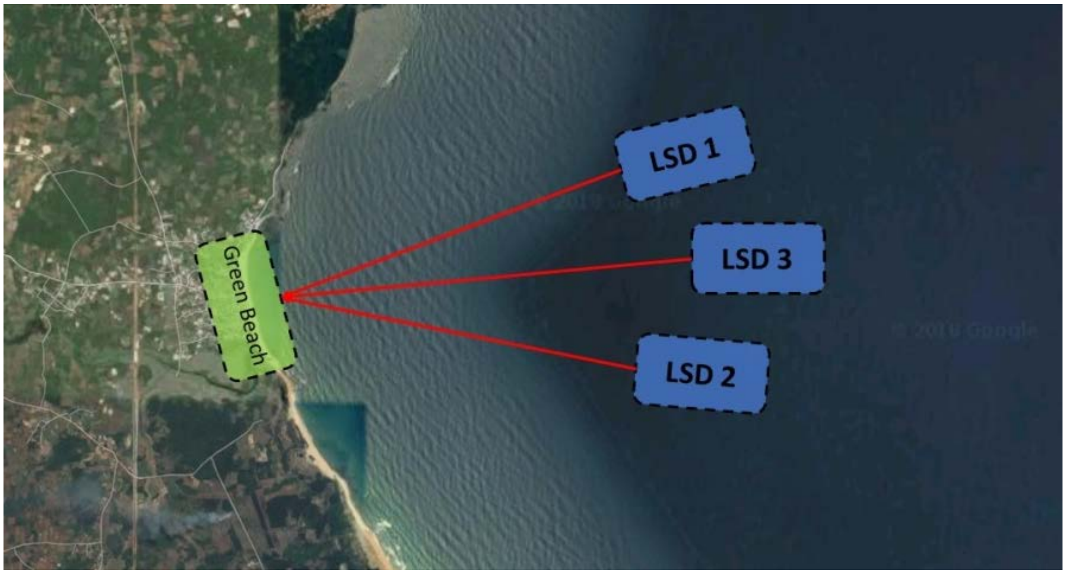



Figure 26. OPERATION Starlight Amphibious Network

As this scenario dealt with the buildup of combat power ashore at a single node we did not need to concern ourselves with Green Beach's capacity to receive forces, or any interconnected land network using CLPs. In a similar light, as no rotary wing assets were used to move V33 ashore we ignore any potential LZs and landing positions within our seabase.

We next model the commodities and demand to accurately reflect the needs of V33's scheme of maneuver. Based on V33's landing diagram we determine that there are six total commodities, which are as follows:

1. ONTOS (in individual vehicle)

2. Tank (in individual tank)

3. Flame Tanks (in individual tank)

4. Company L (in personnel)

5. H&S Company (in personnel)

6. 81mm Platoon (in personnel)

Of note there are three different commodities that consist of personnel that have been classified as separate groups (Co. L, H&S CO, and 81mm Plt.) so that we can track the movement of the different components of V33 as they make their way to shore. While we track these personnel by unit designator, it is important to note that we are really tracking their full combat capability. That is to say, presented with a gap in historical data we assume these units are traveling with mission essential equipment and supplies (MREs, ammunition, jeeps, etc.).

In a similar fashion there are two types of tanks, the M48 Patton and the M67 “Zippo” Tank. The M48 is a standard tank with a 90mm main gun and the M67 is the flamethrower variant of the Patton. These commodities and the V33’s scheme of maneuver form the demand we use in Table 63 and the landing diagram presented in Table 64. Both tables were reconstructed from historical data taken from V33’s after action report [18].

Table 63. V33 Demand at Green Beach

| | |
|---|---|
| ONTOS | **4** |
| TANK | **6** |
| FLAMETANK | **3** |
| Co.L | **167** |
| H&S Co. | **80** |
| 81mm Plt. | **30** |

Table 64. Replication of V33's Landing Diagram. Source: [18].

| | Time | | | |
|---|---|---|---|---|
| Wave | H-Hour | Local | Connector | Load |
| 1 | H-Hour | 6:30 | (11) ATV | Co. I & Co. K |
| 2 | H+2 | 6:32 | (3) LCU | (3) Zippo Tank, (2) ONTOS |
| 3 | H+25 | 6:55 | (2) LCU | Co.L, 81mm Mortar Plt |
| 4 | H+30 | 7:00 | (1) LCU | (2) Patton Tank |
| 5 | H+50 | 7:20 | (2) LCU | (2) ONTOS |
| 6 | H+50 | 7:20 | (1) LCU | (2) Patton Tanks |
| 7 | H+75 | 7:45 | (3) LCU | (2) Patton Tank, H&S Co. |

We create our configurations for the LCUs to mimic the V33 landing diagram (Table 64). As we model our configurations to match that used by V33 it is important to note that the configurations presented in Table 65 reflect loading plans with potentially a lot of unused space. These partial loads reflect a gap in the historical data with respect to additional equipment that would have accompanied the personnel ashore such as jeeps, weapons, sustainment, etc. For example, we know that an LCU when fully loaded with personnel can carry over 300 passengers simultaneously. However, if we consider configuration C that transports only 40 Marines, we would assume this is a lightly loaded LCU and would want to combine it with additional personnel requirements. We refrain from doing so in this situation for two reasons. Firstly, V33 constructed their landing diagram so that combat forces would arrive to the beach first to establish security and only in later waves would support personnel come ashore to establish the battalion's support infrastructure. Secondly, while the configuration only indicates 40 personnel coming ashore it must be assumed that there are large amounts of additional equipment that would accompany this force. To establish a C2 network ashore and basic logistics infrastructure would require numerous containers, vehicles, and varying classes of supply. These assumptions would hold true of the other personnel configurations that would require different types of equipment and supplies to accompany them ashore. Had better historical data been available we would have been able to develop a more detailed plan.

Regardless of this potential gap in information, we construct our configurations to accurately model V33's landing diagram so we can produce a realistic comparison between the MACS-MC output and V33s intended plan.

Table 65. Replicated LCU Configurations

| Config | TANK | TANKFLAME | ONTOS | CoL | HSCo | 81mmPlt |
|---|---|---|---|---|---|---|
| **A** | 0 | 0 | 1 | 0 | 0 | 0 |
| **B** | 0 | 0 | 0 | 167 | 0 | 0 |
| **C** | 0 | 0 | 0 | 0 | 40 | 0 |
| **D** | 0 | 0 | 0 | 0 | 0 | 30 |
| **E** | 2 | 0 | 0 | 0 | 0 | 0 |
| **F** | 0 | 3 | 0 | 0 | 0 | 0 |

## C. RESULTS AND COMPARISON

Having modeled all our input parameters to reflect the same planning considerations, we ran our model in MACS-MC using the mixed integer linear program Quickest Flow Model. We used the mixed integer linear program because the amphibious network was uncomplicated, and we only had one connector using six configurations.

MACS-MC generates a schedule for 12 LCU runs that satisfies all shore-based demand. In Figure 27 we plot the planned landings of each LCU wave V33 estimated against the MACS-MC output. The blue triangles on the left represent the planned landings for the three LCUs created prior to the actual operation, while the green trials represent each LCUs landing times from the MACS-MC. Each column corresponds to one specific LCU. V33 planned for more times between runs, probably as a buffer in their planning process, with the final waves arriving at the beach at H+75. The buffer V33 included in their planning process could have served the purposes of providing a more conservative approach to how the landings would take place while also hoping to space the arrivals to limit congestion at the beach. The MACS-MC tool estimated a much tighter timeline between runs, with the final wave landing on the beach at H+62, which was only two minutes past the actual landing of the final wave.

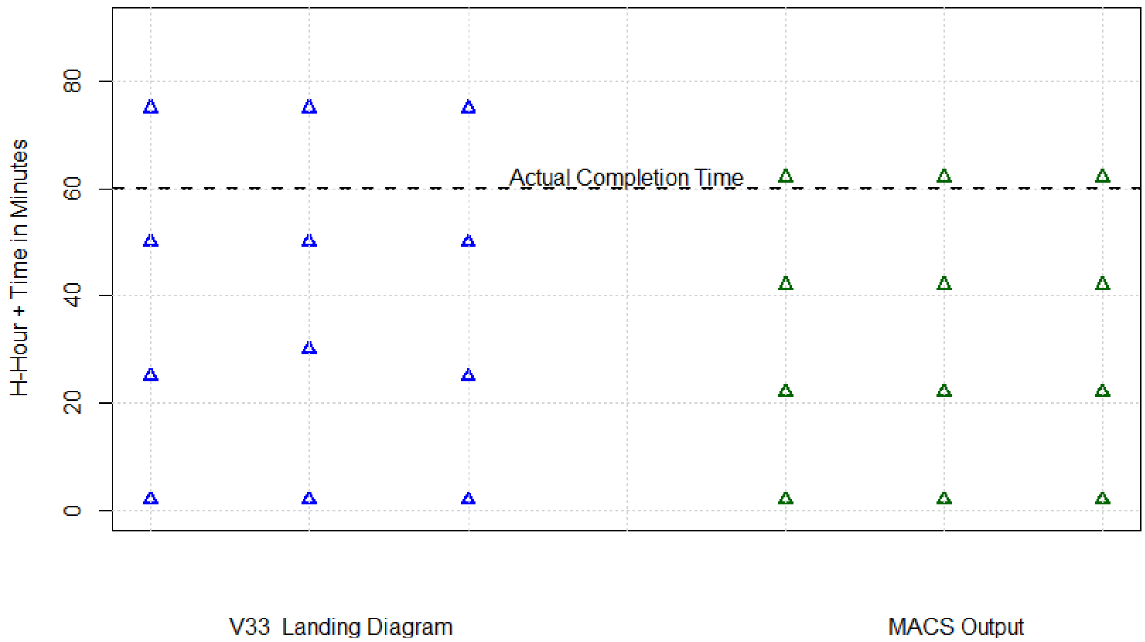



Figure 27. V33 and MACS-MC Landing Diagram Comparison

What we demonstrate with this comparative example is that when using the same planning factors as V33, MACS-MC can rapidly produce a landing diagram that accurately predicts the landing of friendly forces. It is important to remember that we had to compensate for gaps in the historical data by developing partially filled configurations. Had accurate historical information been available for the additional equipment and supplies that would have accompanied the force ashore we could have produced a more accurate schedule. While this historical example is relatively simple in terms of the amphibious network and connectors used, it still demonstrates the speed and accuracy the MACS-MC tool can provide to amphibious planners.

THIS PAGE INTENTIONALLY LEFT BLANK

# LIST OF REFERENCES

[1] *The Marine Corps Operating Concept: How an Expeditionary Force Operates in the 21st Century*, 1st ed., Headquarters United States Marine Corps, Washington, D.C., USA, 2016, pp. 8.

[2] *Professional Military Education*, MCO 1553.4B, Headquarters United States Marine Corps, Washington, D.C., USA, 2008, pp. 1–2.

[3] R. M. Christafore, "Generating ship-to-shore bulk fuel delivery schedules for the Marine Expeditionary Unit." M.S. thesis, Dept. of OR, NPS, Monterey, California USA, 2017. [Online]. Available: https://calhoun.nps.edu/handle/10945/55582

[4] H. W. Hamacher and S. A. Tjandra, "Mathematical modelling of evacuation problems: Astateofart," Institut Techno- und Wirtschaftsmathematik, Los Angeles, CA, Tech. Rep. TR-2000 (4230-46)-3, Nov. 1988.

[5] E. P. Kearns, "Air Cushioned Landing Craft (LCAC) based ship to shore movement simulation: A decision aid for the amphibious commander, a (SMMAT) application." M.S. thesis, Dept. of OR, NPS, Monterey, California USA, 1994. [Online]. Available: https://calhoun.nps.edu/handle/10945/30569

[6] M. J. Powell, "Optimal allocation of assault support aircraft in the sustainment of Marine Corps expeditionary maneuver warfare." M.S. thesis, Dept. of OR, NPS, Monterey, California USA, 2002. [Online]. Available: https://calhoun.nps.edu/handle/10945/9807

[7] N. L. Reitter. "A decision support system for sea-based sustainment operations." M.S. thesis, Dept. of OR, NPS, Monterey, California USA, 1999. [Online}. Available: https://calhoun.nps.edu/handle/10945/8195

[8] D. S. Powell. "An optimization model for sea-based logistics supply system for the Navy and Marine Corps." M.S. thesis, Dept. of OR, NPS, Monterey, California USA, 2004. [Online}. Available: https://calhoun.nps.edu/handle/10945/1360

[9] P. W. Ward. "Optimizing ship-to-shore movement for hospital ship humanitarian assistance operations." M.S. thesis, Dept. of OR, NPS, Monterey, California USA, 2008. [Online]. Available: https://calhoun.nps.edu/handle/10945/4153

[10] *Amphibious Operations: Amphibious Foundations Sub-Course*, Marine Corps CDET, Quantico, VA, US, 2017, pp. 141, 148.

[11] *Amphibious Operations: Landing Plan Sub-Course*, Marine Corps CDET, Quantico, VA, USA, 2017, pp. 121.

[12] *Principle Technical Characteristics of U.S. Marine Corps Engineer Equipment*, TM 11275-15/3D, Marine Corps Systems Command, Quantico, VA, USA, 2009, pp. 2–12, 2–65, 2–128, 2–129.

[13] *Principle Technical Characteristics of U.S. Marine Corps Motor Transport Equipment*, TM 11240-ODA, Marine Corps Systems Command, Quantico, VA, USA, 2010, pp. 2–3, 3–17, 6–17.

[14] *Commander's Quick Reference Amphibious Warfare Handbook*, EWTG-PAC, Coronado, CA, USA, 2016, pp. 8–13, 11.

[15] C.W. Strickland, "Generating efficient and robust schedules to deliver bulk fuel via amphibious connectors." M.S. thesis, Dept. of OR, NPS, Monterey, California USA, 2018. [Online]. Available: https://calhoun.nps.edu/handle/10945

[16] "Analytical Tasking for the Maritime War of 2030," class handout for Joint Campaign Analysis, Dept. of OR, NPS, Monterey, CA, USA, winter 2018.

[17] J. SHULIMSON and C. M. JOHNSON, *U.S. Marines in Vietnam- the Landing and the Buildup*. Quantico, VA, USA: Marine Corps University Press, 1978, pp 69–73.

[18] United States Marine Corps, "3rd Marine Division Operation Starlight command diary for the month of August 1965," Chu Lai, Vietnam, Rep. 65–929, 1965. [Online]. Available: http://www.operation-starlite.com/ 3rdmardiv-command-diary .pdf

# INITIAL DISTRIBUTION LIST

1. Defense Technical Information Center
   Ft. Belvoir, Virginia

2. Dudley Knox Library
   Naval Postgraduate School
   Monterey, California