



NAVAL POSTGRADUATE SCHOOL

MONTEREY, CALIFORNIA

THESIS

**A METHOD FOR QUANTIFYING MANPOWER, PERSONNEL,
TRAINING, AND EDUCATION (MPT&E) IMPACTS ON
COMMAND AND CONTROL (C2) RESILIENCE
IN MARITIME OPERATIONS CENTERS**

by

Shane Lattimore Beavers

March 2019

Thesis Advisor:
Second Reader:

David L. Alderson Jr.
Connor McLemore,
OPNAV N810

Approved for public release. Distribution is unlimited.

Reissued 17 APR 2020 to correct spelling of the term "pairing."

THIS PAGE INTENTIONALLY LEFT BLANK

REPORT DOCUMENTATION PAGE			<i>Form Approved OMB No. 0704-0188</i>	
Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instruction, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188) Washington, DC 20503.				
1. AGENCY USE ONLY (Leave blank)	2. REPORT DATE March 2019	3. REPORT TYPE AND DATES COVERED Master's thesis		
4. TITLE AND SUBTITLE A METHOD FOR QUANTIFYING MANPOWER, PERSONNEL, TRAINING, AND EDUCATION (MPT&E) IMPACTS ON COMMAND AND CONTROL (C2) RESILIENCE IN MARITIME OPERATIONS CENTERS			5. FUNDING NUMBERS	
6. AUTHOR(S) Shane Lattimore Beavers				
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) Naval Postgraduate School Monterey, CA 93943-5000			8. PERFORMING ORGANIZATION REPORT NUMBER	
9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES) N/A			10. SPONSORING / MONITORING AGENCY REPORT NUMBER	
11. SUPPLEMENTARY NOTES The views expressed in this thesis are those of the author and do not reflect the official policy or position of the Department of Defense or the U.S. Government.				
12a. DISTRIBUTION / AVAILABILITY STATEMENT Approved for public release. Distribution is unlimited.			12b. DISTRIBUTION CODE A	
13. ABSTRACT (maximum 200 words) This thesis develops a logic for objectively measuring the required effort that the staff of a maritime headquarters and its embedded maritime operations center must exert to accomplish tasks within a time period. It also provides a way to gain insights into the relative quality of work and risk assumed by them to do so, given the current manning and experience levels of its staff members. We focus on a method for considering staff activities—conducted under the constraints of limited resources by a versatile workforce—as a project scheduling problem, and we expand previous project scheduling methods to include the simultaneous consideration of multiple projects with limited resources. We develop a simulation-based tool, called the Dynamic Resource Allocation Analysis Simulation Tool Kit (DRAASTK), that incorporates manager-defined preferences and priority rules to provide task-resource pairings, a list of delays and the resources responsible, and resource utilization information. We use these outputs to quantify the risk to mission accomplishment that is accepted when staff levels, training, and equipment are underfunded, and articulate a more realistic picture for the level of effort a staff must exert to support the commander's command and control (C2) mission.				
14. SUBJECT TERMS integrated networks, maritime operations center, Resource Constrained Project Scheduling Problem			15. NUMBER OF PAGES 101	
			16. PRICE CODE	
17. SECURITY CLASSIFICATION OF REPORT Unclassified	18. SECURITY CLASSIFICATION OF THIS PAGE Unclassified	19. SECURITY CLASSIFICATION OF ABSTRACT Unclassified	20. LIMITATION OF ABSTRACT UU	

THIS PAGE INTENTIONALLY LEFT BLANK

Approved for public release. Distribution is unlimited.

**A METHOD FOR QUANTIFYING MANPOWER, PERSONNEL, TRAINING,
AND EDUCATION (MPT&E) IMPACTS ON COMMAND AND CONTROL (C2)
RESILIENCE IN MARITIME OPERATIONS CENTERS**

Shane Lattimore Beavers
Lieutenant Commander, United States Navy
BS, United States Naval Academy, 2004

Submitted in partial fulfillment of the
requirements for the degree of

MASTER OF SCIENCE IN OPERATIONS RESEARCH

from the

**NAVAL POSTGRADUATE SCHOOL
March 2019**

Approved by: David L. Alderson Jr.
Advisor

Connor McLemore
Second Reader

W. Matthew Carlyle
Chair, Department of Operations Research

THIS PAGE INTENTIONALLY LEFT BLANK

ABSTRACT

This thesis develops a logic for objectively measuring the required effort that the staff of a maritime headquarters and its embedded maritime operations center must exert to accomplish tasks within a time period. It also provides a way to gain insights into the relative quality of work and risk assumed by them to do so, given the current manning and experience levels of its staff members.

We focus on a method for considering staff activities—conducted under the constraints of limited resources by a versatile workforce—as a project scheduling problem, and we expand previous project scheduling methods to include the simultaneous consideration of multiple projects with limited resources.

We develop a simulation-based tool, called the Dynamic Resource Allocation Analysis Simulation Tool Kit (DRAASTK), that incorporates manager-defined preferences and priority rules to provide task-resource pairings, a list of delays and the resources responsible, and resource utilization information.

We use these outputs to quantify the risk to mission accomplishment that is accepted when staff levels, training, and equipment are underfunded, and articulate a more realistic picture for the level of effort a staff must exert to support the commander's command and control (C2) mission.

THIS PAGE INTENTIONALLY LEFT BLANK

Table of Contents

1	Introduction	1
1.1	Background	1
1.2	Research Objectives	3
1.3	Thesis Structure	3
2	Background and Literature Review	5
2.1	Project Scheduling Methods	5
2.2	Previous NPS Work	18
2.3	Our Contribution	20
3	Model and Methodology	23
3.1	Problem Statement	23
3.2	Motivating Example	24
3.3	Mathematical Model	27
3.4	Solution Method	30
3.5	Proof of Concept	46
4	Modeling MOC Staff Operations as a CMRCMPSP	59
4.1	Converting MOC Staff Operations into Model Inputs	59
4.2	DRAASTK Analysis	62
5	Conclusions and Future Work	71
5.1	Summary and Conclusions	71
5.2	Future Work	72
	Appendix: Terms	75
	List of References	77

List of Figures

Figure 1.1	Maritime Operations Center (MOC) Organization Concept. Adapted from Department of the Navy (2013).	2
Figure 2.1	Critical Path Method. Adapted from Pinha (2015).	7
Figure 2.2	Project Evaluation and Review Technique. Adapted from Pinha (2015).	7
Figure 2.3	Resource Constrained Project Scheduling Problem (RCPSP). Adapted from Pinha (2015).	10
Figure 2.4	Classification of Resource Constrained Project Scheduling Problems (RCPSPs). Adapted from Gen et al. (2008).	11
Figure 2.5	Multi-mode Resource Constrained Project Scheduling Problem (MR-CPSP). Adapted from Pinha (2015).	13
Figure 2.6	Combinatorial Multi-mode Resource Constrained Multi-Project Scheduling Problem (CMRCMPSP). Adapted from Pinha (2015).	14
Figure 2.7	CMRCMPSP Queue Schema. Adapted from Pinha (2015).	17
Figure 2.8	Integrated Training Center (ITC) Overview Diagram. Adapted from Davidson (2011)	19
Figure 2.9	RCPSP Classification of the Problem Model. Adapted from Gen et al. (2008)	20
Figure 3.1	Example Program Model. Adapted from Pinha (2015).	24
Figure 3.2	Dynamic Resourcing Allocation Analysis Simulation Tool Kit (DRAASTK) Overview Diagram	31
Figure 3.3	Class Object Attributes	32
Figure 3.4	Dynamic Resourcing Allocation Analysis Simulation Tool Kit (DRAASTK) Model Design	33
Figure 3.5	Resource Selection	36

Figure 3.6	Conflict Resolution	37
Figure 3.7	Resource Selection Reevaluation	39
Figure 3.8	Example Resource Utilization Chart.	45
Figure 3.9	Example Skill Utilization Chart.	46
Figure 3.10	Project Baseline.	47
Figure 3.11	Task Finish Time Chart: Vignette 1	51
Figure 3.12	Resource Utilization Chart: Vignette 1	52
Figure 3.13	Task Finish Time Chart: Vignette 2	55
Figure 3.14	Resource Utilization Chart: Vignette 2	56
Figure 4.1	Supporting Task OP 5.3.1 Procedures. Adapted from Department of the Navy (2014).	60
Figure 4.2	Supporting Task OP 5.3.1 Required Skills. Adapted from Department of the Navy (2014).	62
Figure 4.3	OP 5.3.1 Network Model. Adapted from Jones et al. (2018).	63
Figure 4.4	Task Finish Time Chart: OP 5.3.1	66
Figure 4.5	Skill Utilization: OP 5.3.1	68

List of Tables

Table 2.1	Resource Pool: CMRCMPSP	16
Table 3.1	Example Program Model Modes	25
Table 3.2	Example Resource Pool	26
Table 3.3	Example Schedule Summary Log.	41
Table 3.4	Example Program Schedule Log	42
Table 3.5	Example Resource Conflict Log	43
Table 3.6	Example Skill Scarcity Log	43
Table 3.7	Example Resource Utilization Log	44
Table 3.8	Example Skill Utilization Log	46
Table 3.9	Project Modes: Baseline	48
Table 3.10	Resource Pool: Baseline	48
Table 3.11	Schedule Summary Log: Baseline.	49
Table 3.12	Resource Conflict Log: Baseline	50
Table 3.13	Modes: Vignette 2	53
Table 3.14	Resource Pool: Vignette 2	54
Table 3.15	Schedule Summary Log: Vignette 2.	56
Table 4.1	Project Modes: OP 5.3.1	64
Table 4.2	Resource Pool: OP 5.3.1	65
Table 4.3	Schedule Summary Log: OP 5.3.1	66
Table 4.4	Resource Conflict Log: OP 5.3.1	67

THIS PAGE INTENTIONALLY LEFT BLANK

List of Acronyms and Abbreviations

C2	Command and Control
CAR	Critical Activity Rating
CFMCC	Coalition Force Maritime Component Commander
CFT	Cross Functional Team
CMRCMPSP	Combinational Multi-mode Resource Constrained Multi-Project Scheduling Problem
CPM	Critical Path Method
DoD	Department of Defense
DRAASTK	Dynamic Resource Allocation Analysis Simulation Tool Kit
GCC	Geographic Combatant Commander
ITC	Integrated Training Center
JFMCC	Joint Force Maritime Component Commander
JTF	Joint Task Force
JWICS	Joint Worldwide Intelligence Communication System
MET	Mission Essential Task
MHQ	Maritime Headquarters
MOC	Maritime Operations Center
MPT&E	Manpower, Personnel, Training, and Education
MRCMPSP	Multi-mode Resource Constrained Multi-Project Scheduling Problem
MRCPSP	Multi-mode Resource Constrained Project Scheduling Problem

NAVFOR	Naval Force
NPS	Naval Postgraduate School
PERT	Project Evaluation and Review Technique
PMSGs	Parallel Mode Schedule Generation Scheme
RCPSp	Resource Constrained Project Scheduling Problem
SME	Subject Matter Expert
STREAM	Short-Term Resource Allocation and Management
USN	U.S. Navy
USG	United States Government

Executive Summary

In professions where loss of life or serious harm is a tangible and ever-present consequence, failure is simply not an option, regardless of the circumstances or resources at hand. This is especially true among warfighters and the organizations that support them. Bolstered by such resolve, administrative staffs may experience periods of success where the strains of limited manpower, personnel, training, and educational (MPT&E) resources are overcome by a “never say die” ethos. However, their chances for success or quality of their work is put at risk the longer they must operate under the stresses of limited resources, particularly in instances of rapid changes to intensity or operational tempo. Unfortunately, the near and long-term operational impacts of these MPT&E factors, and risks to staff performance as they continue to “do more with less,” are difficult to articulate to higher echelons when those impacts are effectively mitigated by the efforts of the very same people experiencing them.

One prime example of this resilient type of staff are the dedicated men and women serving within any of the Navy’s maritime headquarters.

As the theater combatant commander’s maritime warfighter, the Navy fleet commander manages theater maritime assets to prepare and provide forces, and employs those naval forces to support and perform operational missions. To execute these duties, each of their maritime headquarters (MHQ)s is designed to be appropriately manned, trained, and equipped to support both functions; even as the commander is assigned various responsibilities to conduct enduring and contingency missions as a Naval Force (NAVFOR) Commander, Joint Force Maritime Commander (JFMCC), Coalition Force Maritime Component Commander (CFMCC), or even support for joint task force (JTF) operations.

Within the MHQ staff, a separate maritime operations center (MOC) bridges the command and control (C2) gap between strategic level guidance and tactical employment of forces. These MOC staffs perform tasks that provide the commander with the organizational structure, procedures, and expertise support needed to make and communicate informed decisions, while setting the conditions for subordinate success; separate from the MHQ’s fleet management mission. However, even with well-defined requirements in place, MPT&E

impacts still pose potential risks to a MOC's ability to effectively support the commander's C2 role. On a practical level, it may not be possible for all staffs to be sufficiently equipped or manned to designed levels at all times, and en route training timelines could either cause gaps in billet fills, or delays in training for newly reporting personnel. The potential cumulative effects of these impacts could result in a reduced, but highly dedicated staff, capable of accomplishing its day-to-day missions, or even contingency operations for a limited time. However, this level of performance would require the consolidation of multiple MOC position responsibilities, and the leveraging of in-house experience as a replacement for proper training and education. Ultimately, this model could prove unsustainable, as the increased operations tempo of contingency operations stress the MOC's current manpower beyond the limits of what their efforts are able to mitigate, and leads to either delays in task completion and/or impacts to the quality of support they are able to provide.

This thesis develops a logic for objectively measuring the required effort a staff member must exert to accomplish tasks within a time period, and provides a way to gain insights into the relative quality of work and risk assumed by them to do so, given the current manning and experience levels of their staff members. We focus on a method for considering staff activities, conducted under the constraints of limited resources by a versatile workforce, as project scheduling problems for task-on-node networks. To create those networks that represent the functions of a MOC, we describe a procedure for converting the supporting tasks that make up the mission essential tasks (METs) a MOC is assigned, into a collection of network nodes with precedence relationships, expected durations, and resource requirements in the form of the functional team-leads needed to work tasks.

We explore current techniques for representing and solving resource-constrained project scheduling problems, in search of a model that best describes the "real world" aspects of a staff, their work, and the MPT&E factors that impact them. In reviewing the merits of the Critical Path Method (CPM) and the Project Evaluation and Review Technique (PERT) for identifying the tasks who most affect the completion time of a network, we ultimately dismiss them as necessary, but not sufficient because they do not account for resource requirements or constraints. To better suit the level of complexity a model must account for to represent staff activities, the family of Resource Constrained Project Scheduling Problems (RCPSPs) is explored. Many variations to the RCPSP exist, each adding a level of variability with the aim of modeling a more general scenario; including multiple combinations of resource types

to complete the same task, and the simultaneous consideration of multiple projects. Each variation has its own set of solution methods as well, with each technique leaning farther from finding optimal answers and more towards “good enough” as complexity increases.

We finally settle on an adaptation of the Combinatorial Multi-mode Resource Constrained Multiple Project Scheduling Problem (CMRCMPSP) model, which allows us to capture the flexibility of a MOC staff in many ways. It accounts for a supervisor’s ability to consider staff members as sets of combinations according to the skills they can perform, rather than by job titles. This mirrors the situations where a staff must tap personnel for the skills they possess without regard for the positions to which they are assigned; in the name of “getting the job done.” The CMRCMPSP model also incorporates the various options in the way resources can be applied to complete a task, the need to conduct multiple parallel projects that must share a common pool of resources, the abilities of talented staff members whom are able to wear many positional “hats” while contributing to multiple concurrent efforts with varying degrees of proficiency (based on level of training, education, and/or experience), and a supervisor’s preferences or organizational policies for the employment of resources which serve to resolve conflicts between competing tasks for those limited resources.

Using the Python programming language, we create a custom discrete event simulation algorithm to solve our adaptation of the CMRCMPSP model, which develops a schedule for the given projects by using the available resource pool. Designated the Dynamic Resource Allocation Analysis Simulation Tool Kit, or DRAASTK, simulation outputs include information for both the task-resource pairings and resource usage necessary to achieve the provided schedule. Along with a recommended schedule for task completion, a record for each delayed task, and the lacking required skills that caused its delay at the time of its evaluation, is provided to give insight into the organization’s skill deficiencies and potential points for where additional resources should be focused. As for the current resource utilization and risk, a record for the percentage of time each staff member is involved with a task is taken; broken down by how much of that time is spent engaged with either a single or multiple simultaneous tasks as a measure of that staff member’s exerted effort. The level of proficiency for which a resource holds for the skill applied to a task is also noted and provided. A subjective value based on the training, education, and experience level a staff member holds for a particular skill, the collective average for the

level of proficiency of all staff members whose skills were utilized to complete a task is analogous to the relative level of risk in quality and performance a staff assumes to achieve the provided timeline.

As a resourcing decision aide, this thesis provides many opportunities for additional work, in both data and simulation refinement. In particular, in order to consider MOC staff operations as networks to analyze, significant subject matter expert (SME) input is required to apply precedence relationships and expected durations to all supporting task and procedures. A MOC organizational review and development of these networks is an important first step in evaluating MOC staff operations. Developing granularity in the resource pool is also a viable area for expansion. Currently DRAASTK considers skills as the positions of team leaders common to all MOCs, but this can be evolved to include the duties and skills of each team member and their individual availability schedules, allowing for a more robust examination of a specific MOC staff. Finally, future work can focus on the flexibility of the simulation itself. Incorporating stochastic elements for initial and/or resource-based task durations, as well as consideration for individual resource availability, all works towards developing a more “real world” model and method for gaining viable insights to inform and aide decision makers with complex resource allocation resolutions.

Our method for pairing a limited pool of skilled resources with the tasks that require them and recording the points of resource-induced delays and relative quality of skill in task completion are the main contributions of this thesis. Compared to a staffing profile inline with a MHQ’s intended manning, the schedules and outputs generated by DRAASTK give insights into the amount of effort and risk to quality and timeliness an actual staffing profile exhibits. This information provides a quantifiable basis to articulate the existing MPT&E impacts and risks to mission that a staff experiences, and may inform the conversation needed to justify the resourcing requirements of a staff that continues to succeed, despite “doing more with less.”

Acknowledgments

It is no exaggeration when I say that this work, and everything that I have achieved since high school leading up to it, couldn't have been possible without the love, support, understanding, and patience of our family's 5-star admiral-general, my incredible wife, Samantha.

To my extended family and fellow members of cohort "tuplé", I couldn't have done this for the last two-plus years without your support. With Mom, Dad, my sisters, Aunt Rochelle and Uncle Richie, Sam and Gabriel were never alone when I couldn't be there, and that is a debt I couldn't begin to repay. As for my second family, only you know what we've achieved, and what it truly cost us to get here. Stronger bonds may only be forged on literal battlefields. Fair winds and following seas, to all of my fellow shipmates and nerds from other herds.

To the faculty and staff, I want to extend my appreciation for your expertise and guidance. Even when it didn't seem like it, I was always paying attention, and I'm sure that the knowledge you imparted to my cohort is reflected throughout this work. Thank you.

To my official and adopted advisors, I cannot thank you enough for your mentorship and direction. The framework, insights, and practical feedback of Dr. Alderson, Dr. Eisenberg, Dr. Norton, Dr. Carlyle, LCDR McLemore, COL Appleget, and CAPT Kline often provided the nuggets of inspiration I needed to get over the many, many humps and instances of writer's block I had to tackle in this effort. I can't imagine ever pulling off something like this without your counsel, gentlemen. Each of you influenced significant contributions to the way I approached and presented my thesis, and for that, I thank you.

Finally, I reserve a special expression of gratitude toward Dr. Denis da Cruz Pinha. His work in extending project scheduling techniques, specifically within the naval maintenance and construction industries, was a unique inspiration and reference for my research into a human resources management problem. Dr. Pinha's graciousness in sharing and conversing about his efforts proved instrumental to the development of my approach to the problems I address in my thesis, and I will forever be grateful to him and what I consider to be his revolutionary contributions to the field of operations research. Muito obrigado pela sua ajuda, senhor!

THIS PAGE INTENTIONALLY LEFT BLANK

CHAPTER 1:

Introduction

Maritime operations centers (MOCs) provide Navy fleet commanders with a command and control (C2) organization that enables them to operationally employ their forces. In order to provide the required support to the commander in both short-term and prolonged engagements, this organization must consist of a sufficiently resourced, capable, and proficient staff. In December of 2016, the U.S. Fleet Forces Command's MOC Training Team conducted certification inspections for both the U.S. Pacific (Williams et al. 2016) and U.S. Seventh Fleet (Meade et al. 2016) MOCs. Their findings indicated that manning and training levels impacted the MOC in several functional areas. However, the reports concede that this operational impact is difficult to subjectively measure, mainly due to the mitigation efforts of assigned personnel with on-the-job experience and the perseverance of staffs facing limited resources (Meade et al. 2016). This study proposes a methodology for analyzing the operations of a MOC and constructing quantifiable metrics for articulating impact and risk.

1.1 Background

The following sections provide relevant background for maritime operations centers. A brief discussion concerning their staff organization is also provided, along with a description of our research objectives and thesis structure

1.1.1 MOC Organization

As the maritime component commander, the Navy fleet commander is responsible for both preparing and providing forces to the geographic combatant commander (GCC) as well as employing them (Department of the Navy 2008). Within a maritime headquarters (MHQ), there exist two complementary arrangements for personnel and processes. The first is the traditional, vertical "N-code" structure, that assigns people by functional areas like intelligence, logistics, or administration (Department of the Navy 2013). This classic hierarchical staff structure has served the Navy well in the past, but because expertise is stovepiped in that structure, it does not facilitate the level of cross-talk needed to support a commander's

decision-making process and C2 functions during fast-paced military operations. In contrast to the traditional organizational style, the MOC organization can be thought of as a loosely-bound network, embedded within and spanning across the N-code structure (see Figure 1.1 for a simplified diagram of the MOC organizational structure concept). At the core of this network are 94 specific MOC staff officer positions known as *process drivers* (Department of the Navy 2014). Per the MOC standardization manual (Department of the Navy 2014), these process drivers are the commissioned officers critical to leading MOC work. In accordance with Department of the Navy (2014), they determine the products, timelines, support requirements, battle rhythm, and coordination requirements for the various boards, bureaus, centers, cells, and working groups, as well as elements, groups, offices, planning teams, and other coordinating bodies that form the cross-functional teams (CFTs) that plan, direct, monitor, and assess assigned missions in support of the commander's battle rhythm.

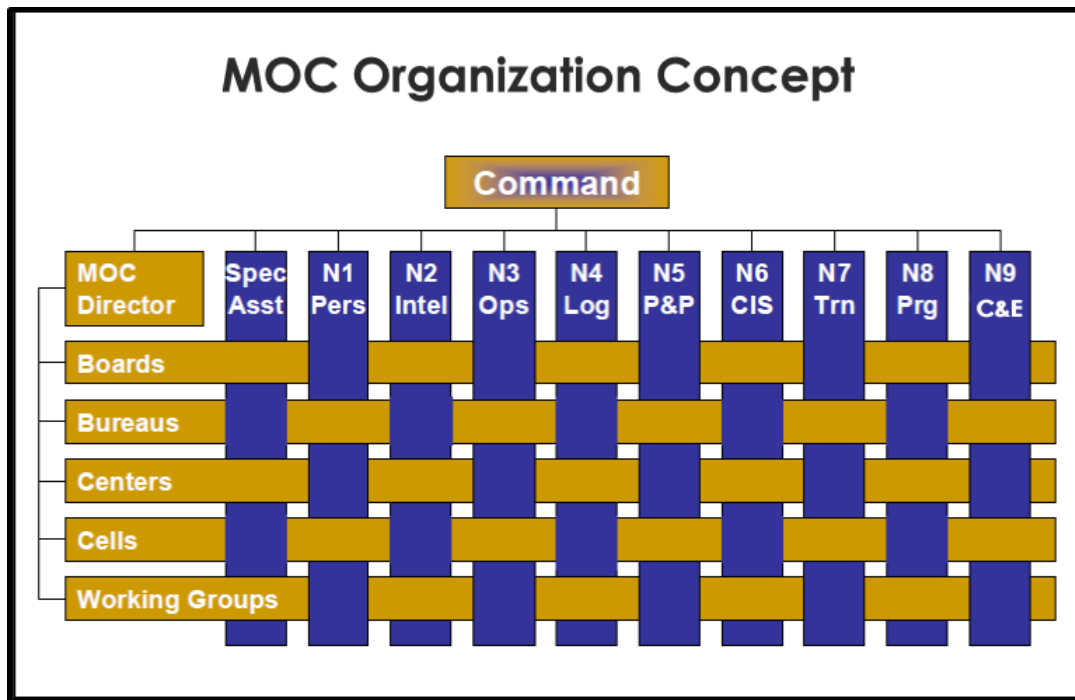


Figure 1.1. Maritime Operations Center (MOC) Organization Concept. Adapted from Department of the Navy (2013).

A representation of the MOC organizational concept, illustrating how MOC functions are embedded and interwoven across the vertical N-code structure of the headquarters organization.

1.1.2 MOC Tasks and Personnel

In its support of the commander’s role as the maritime component commander, the MOC must meet the requirements of two core functions that are divided among a set of 11 mission essential task (METs). These METs are themselves composed of various numbers of 177 total supporting tasks, each of which consisting of a set of procedures that are accomplished by specific combinations of the 94 process driver-led functional teams (see Department of the Navy 2014, for a description of mission essential tasks and supporting tasks). These METs are specific to the MOCs core functions and designed to be separate from those associated with fleet management. However, the staff members responsible for either function are not mutually exclusive. Many often overlap, performing the duties of several positions within the organization as “dual-hatted” staff members. While an understaffed or insufficiently trained MOC may be able to meet many of its mission requirement metrics during the relatively stable routine of “deterrence, maritime security and forward presence shaping operations” in Phase 0 of the operational levels of war (Joint Chiefs of Staff 2018), this inherent sharing of critical personnel resources becomes a potential point of failure as the operational tempo of both the fleet headquarters and MOC increase significantly when the commander is assigned a variety of contingency support missions in the latter operations of Phases 1 through 5.

1.2 Research Objectives

In this thesis, we apply network analysis techniques, specifically project scheduling methods, to the operations of a maritime operations center staff. We expand on previous work conducted to quantify the risk to mission accomplishment that is accepted when staff levels, training, and equipment are under-funded. Furthermore, we expand that methodology to include the simultaneous consideration of multiple projects and limited resources.

1.3 Thesis Structure

This thesis proceeds as follows. In Chapter 2, we review the history, terminology and current literature related to applicable project scheduling problems. In Chapter 3, we formulate the model we use for consideration. We also describe the components of the simulation we design to provide insight into the scheduling problem we have developed, and conclude the chapter with an analysis of a proof of concept example. In Chapter 4, we consider MOC

staff operations as a project scheduling problem, apply our simulation to an expansion of previous work, and present the results of our analysis and its implications. Finally, in Chapter 5, we present our conclusions and research areas for future work.

CHAPTER 2:

Background and Literature Review

This chapter provides basic background information on project scheduling methods and reviews their advances in complexity to better model more “real world” situations. At the end of this chapter, we review previous work directly relating to this thesis.

2.1 Project Scheduling Methods

The Project Management Institute’s Project Management Body of Knowledge defines a *project* as “a temporary endeavor of tasks undertaken to create a unique product, service, or result which may be subject to precedence and require resources” (Snyder 2013). Examples of projects exist in all aspects of the economy, including manufacturing, construction, repair and maintenance, design and development, and government programs. In general, all projects consist of *tasks* that must be accomplished according to a schedule constrained by precedence relations among those tasks, associated resource requirements for their execution, and an overall goal of minimizing total project cost; cost being in relation to either monetary, temporal or any other consideration deemed important to the project manager.

According to Vanhoucke (2012), project managers have utilized formalized methods for organizing, scheduling, and controlling projects to best achieve their stated goals, since at least the late 1950s. He further explains that the earliest of these approaches are powerful but simple techniques for analyzing, planning, and scheduling large, complex projects. Known as the *Critical Path Method (CPM)* and *Project Evaluation and Review Technique (PERT)*, Vanhoucke (2012) describes how these tools provide a means for determining which tasks, of the many that comprise a project, are "critical" because of their impact on total project time and/or cost.

Gen et al. (2008) illustrates that, while CPM and PERT are exceedingly popular in industry due to their simplicity in construction and relative ease in determining exact solutions, they do not directly take resource requirements into account. They explain that this is both unrealistic and potentially costly to the project manager who may have to consider variability

in resource availability and/or even executing multiple competing projects. Gen et al. (2008) goes on to define the problem of scheduling tasks under resource and precedence constraints, with the objective of minimizing the project duration, as a *Resource-Constrained Project Scheduling Problem (RCPSP)*. These resource constraints, as referred to by Gen et al. (2008), are limited renewable supplies, such as manpower, materials and machines, which are necessary to completing project tasks. They go on to state that many variations of the RCPSP model have been developed and examined, all with the aim of more accurately modeling “real world” situations. As explained by Gen et al. (2008), these variants incorporate concepts such as multiple task duration-resource requirement options for a single task (known as *modes*), resources that are multi-skilled and interchangeable, or multiple projects that compete for the same resources but are conducted simultaneously. Kolisch (1995) goes on to elaborate that each variant of the RCPSP serves to create a more accurate model than that of the CPM, PERT or basic RCPSP; however, their utilities are diminished given that with each added level of complexity, the computational effort to find an optimal or even feasible solution for the project schedule grows exponentially.

The following sections provide necessary details of CPM, PERT, and RCPSP models.

2.1.1 CPM and PERT

Both the CPM and its extension PERT were first developed in the 1950s and made popular in industry by their use in the U.S. Navy’s Polaris Submarine-Launched Fleet Ballistic Missile program (Kolisch 1995). These techniques assume a project possesses unconstrained resources and represents it as a directed graph network without loop cycles. Figures 2.1 and 2.2 show a network of six tasks for analysis by the CPM and PERT methods, respectively. They use a task-on-node diagram, with nodes describing project tasks and arrows displaying the precedence relationships among them. Additionally, each network has a start (ST) and an end (EN) node, with the values above each numbered node representing task duration.

Task duration is where PERT analysis diverges from the CPM. While task duration for the CPM is deterministic, PERT attempts to address time uncertainty in projects by assuming that each task duration is a random variable between two extreme values which follows a beta distribution (Vanhoucke 2012). The overall estimate is then calculated as a weighted average of the optimistic, most likely, and pessimistic estimates, with more weight given to

the most likely value.

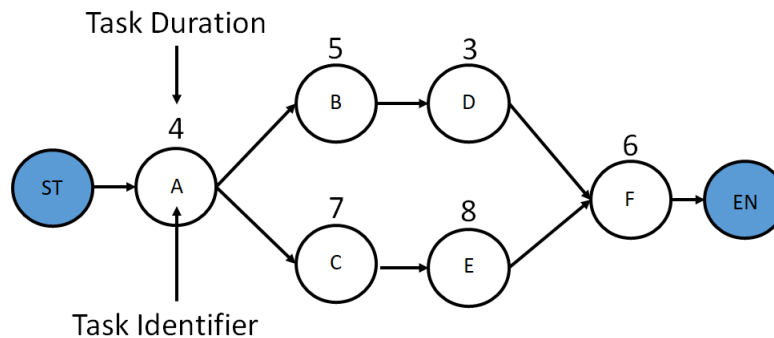


Figure 2.1. Critical Path Method. Adapted from Pinha (2015).

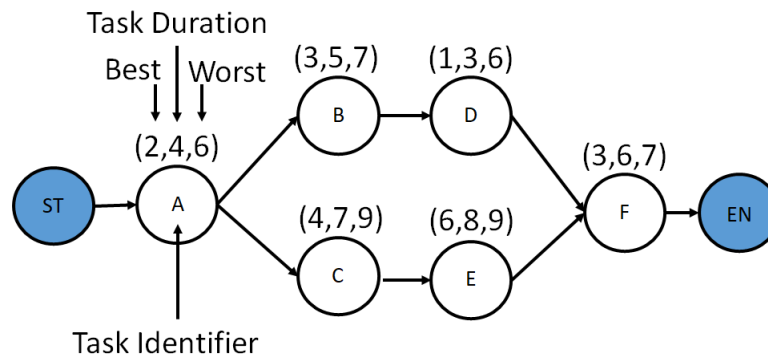


Figure 2.2. Project Evaluation and Review Technique. Adapted from Pinha (2015).

The goal of CPM/PERT, and for that matter all project scheduling efforts, is to determine the tasks within a network of precedence relationships that cannot be delayed without delaying the duration of the entire project. Each of those tasks are referred to as a *critical task* and their relations within the project's network as the *critical path*. Vanhoucke (2012) gives a detailed explanation and practical example for how CPM/PERT problems can be solved to identify these critical tasks and paths, which he summarizes into three generalized steps:

1. Calculate the earliest start schedule
2. Calculate the latest start schedule
3. Calculate the slack for each task

Based on the work of Vanhoucke (2012), we introduce some basic notation to facilitate the

calculation of each of these below. Let $j \in A$ denote the tasks in a given project (alias k). Let d_j denote the *duration* of task j . Let P_j and S_j denote the sets of *predecessor tasks* and *successor tasks*, respectively, for task j .

Earliest Start Schedule (ESS)

According to Vanhoucke (2012), forward calculations in the project network can be used to determine the *earliest start time* (es_j) of each task j . A task's earliest start time is either equal to or larger than the earliest finish time from among all of its predecessors. That earliest start time can be calculated using the following set of forward calculation equations, with the start node ST assumed to begin at time zero:

$$es_{ST} = 0 \quad (2.1)$$

$$es_k = \max(es_j + d_j | j \in P_k). \quad (2.2)$$

The *earliest finish time* (ef_j) of a task j is defined as the sum of its earliest start time and estimated duration, which is given by:

$$ef_j = es_j + d_j. \quad (2.3)$$

In a simple network like that in Figure 2.1, it is easy to calculate that the earliest start times are $es_{ST} = 0$ and $es_A = 0$, es_B and $es_C = 4$, $es_D = 9$, $es_E = 11$, $es_F = 19$, and $es_{EN} = 25$. This gives an overall minimum project duration of 25 units of time.

Latest Start Schedule (LSS)

Vanhoucke (2012) also provides a method for calculating the *latest finish time* (lf_j) of each task j , that is similar in manner to the earliest start time calculation. Using backward calculations that begin at a project end node's assumed finish time δ_{EN} , the latest finish time of a task is determined to be either equal to or less than the latest start time from among all of its successors. That latest finish time can be calculated using the following set of backward calculation equations, starting with the dummy end node EN :

$$lf_{EN} = \delta_{EN} \quad (2.4)$$

$$lf_j = \min(lf_k - d_k | k \in S_j) \quad (2.5)$$

The latest start time (ls_j) of a task j is defined as its latest finish time reduced by its estimated duration, which is given by:

$$ls_j = lf_j - d_j \quad (2.6)$$

Given the project end time of 25 units of time for Figure 2.1's dummy end node, the latest start times of each task are calculated as $ls_{ST} = 0$, $ls_A = 4$, $ls_B = 16$, $ls_C = 11$, $ls_D = 19$, $ls_E = 19$, $ls_F = 25$, and $ls_{EN} = 25$.

Task Slack

Finally, Vanhoucke (2012) provides a method for calculating the amount of slack associated with each task. This value represents the amount of time each task j can be delayed without affecting the duration of the entire project. Vanhoucke (2012) calculates it as follows:

$$ls_j - es_j = lf_j - ef_j \quad (2.7)$$

A task with a slack of zero cannot be delayed without affecting the entire project's duration. Vanhoucke (2012) refers to such a task as a *critical task*, where the *critical path* is the path along a network that consists of these critical tasks. For Figure 2.1, nodes ST, A, C, E, F, and EN have a slack value of 0 and therefore, constitute the critical path for that network.

While simple forms of analysis, the CPM and PERT tools help to draw the focus of a project manager to the most consequential activities, providing an effective basis for the scheduling and monitoring of progress without the consideration of resources.

2.1.2 Resource Constrained Project Scheduling Problems (RCPSP)

The basic Resource Constrained Project Scheduling Problem (RCPSP) was first introduced by Pritsker, Watters, and Wolfe in the late 1960s (Pritsker et al. 1969). This simple version consist of interrelated tasks, each with a single associated time of duration and resource

requirements pair. Figure 2.3 depicts the task-on-node network representation of a single six-task RCPSP project. Each task, nodes A through F, shows their single duration-resource requirements pair, which is predetermined and fixed in a RCPSP.

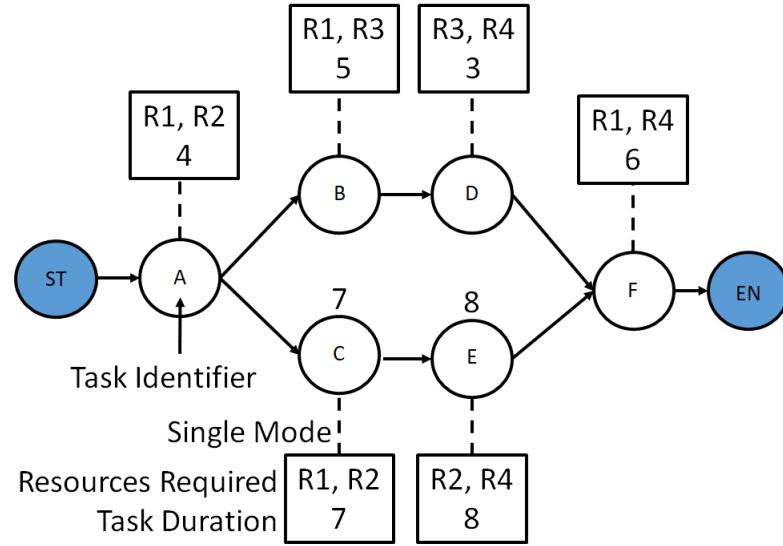


Figure 2.3. Resource Constrained Project Scheduling Problem (RCPSP). Adapted from Pinha (2015).

Several variants of the RCPSP have been proposed since the initial work of Pritsker and his colleagues. As explained by Gen et al. (2008), these variants can be divided into categories based on their problem characteristics, i.e., the number of projects, the number of task-duration/resource requirement options available, resource consumption, and the ability to stop a task's execution before its completion (interruption) (see Figure 2.4). Gen et al. (2008) also states that RCPSP models can be labeled as either *single-project*, where there is only one project to be scheduled, or *multi-project*, where there is more than one project to be scheduled and executed.

According to Gen et al. (2008), RCPSP models can also be divided into two more groups based on the number of task duration-resource requirement pairs available to perform a task. These various alternative combinations are referred to by Gen et al. (2008) as *modes*. Per Kolisch (1995), in the single-mode version, there exists only one task to resource pairing capable of executing a task, whereas in the multiple-mode version, each task can be performed by selecting one out of many different combinations of resources that can complete a given task. Kolisch (1995) goes on to state that the resources in

RCPSP models may also be considered as “limited-but-renewable” from period to period, “non-renewable,” “of-limited-amount” for the duration of the project, or doubly constrained by both restrictions. Finally, Gen et al. (2008) explains that RCPSP models can again be divided into those that are *preemptive*, where a task can be interrupted after it has started, and those that are not (i.e., *non-preemptive*). However, the main body of work in the development of RCPSPs center on the non-preemptive case of its two main variants and their combination; the *Multi-mode Resource Constrained Project Scheduling Problem (MRCPSP)*, the *Resource Constrained Multi-Project Scheduling Problem (RCMPSP)* and their combination, the *Multi-mode Resource Constrained Multi-Project Scheduling Problem (MRCMPSP)* (Wauters et al. 2016).

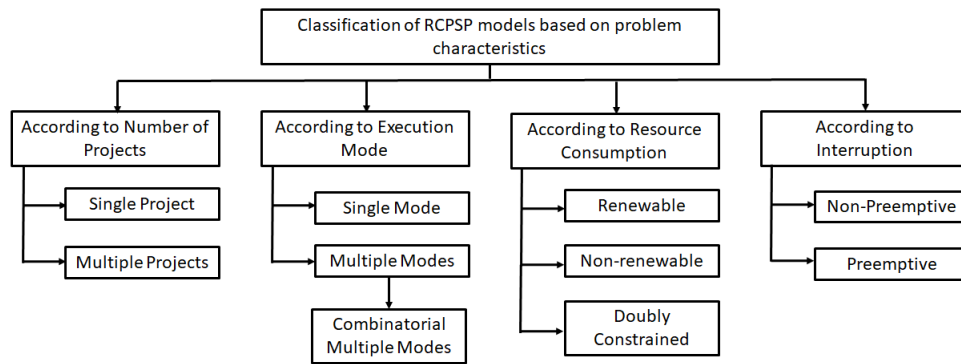


Figure 2.4. Classification of Resource Constrained Project Scheduling Problems (RCPSPs). Adapted from Gen et al. (2008).

The RCPSP, and its more complex extensions, are considered to be the standard approaches to project scheduling, but belong to the class of strongly NP-hard problems (Kolisch 1995). The majority of the solution methods to these types of problems tend to be either exact, or some form heuristic. Exact procedures strive to find the optimal solution, often through the enumeration and evaluation of all possible solutions. These procedures, however, are only useful for solving small projects that have undergone several simplifying assumptions. Heuristic methods, on the other hand, search for a good enough but not necessarily optimal scheduling solution, using more realistic restrictions and assumptions (Vanhoucke 2012). Still, both exact and heuristic methods are limited in their efficiency for solving the complex models that more accurately represent “real world” problems. Therefore, many researches attempt to capture more realistic model restrictions through discrete event simulations that

make use of priority rules (Pinha 2015). These priority-rule-based simulations are very popular among researchers for solving the more complex RCPSP extensions due to several reasons: they achieve a satisfying quality of solutions, they are computationally fast, and they proceed similar to the way decisions are made by project managers (Kolisch 1995).

2.1.3 Combinatorial Multi-Mode Resource Constrained Multi-Project Scheduling Problem (CMRCMPSP)

As best stated by Pinha (2015), the chosen representation for a problem determines its complexity and the size of the search space in which a usable solution exist. For instance, a highly specific model may significantly reduce the breadth of the solution search, and could even be relatively simple to solve, but may only apply to a single or limited range of problem instances.

To better represent the complex nature of “real world” situations and provide meaningful insight to problem solutions, models must possess the flexibility to consider multiple projects, with several tasks competing for limited resources that are themselves dynamic. Until recently, the MRCMPSP was the most general of resource-constrained scheduling model extensions cited in literature (Pinha et al. 2016). It allows for the simultaneous consideration of tasks from several projects, all of which are under precedence and resource constraints, with multiple modes available for each. For illustrative purposes, Figure 2.5 shows the six-task project discussed in previous examples, now with multiple modes available for each task and the inclusion of four resources (denoted R1 through R4). For example, task C has three modes; mode 1 calls for resources R1 and R2 for a duration of 7 units of time, mode 2 requires R2 and R3 for a duration of 8 units of time, and mode 3 needs R1 and R4 for a duration of 10 units of time.

As illustrated by Pinha (2015), multiple modes provide added flexibility to develop project schedules. In the example given in Figure 2.5, the duration of task C increases with the order of the mode option. However, this is not a strict limitation. Depending upon the requirement, the mode order may be analogous to preferences for some value other than duration—such as quality, materials, and cost—that the project manager uses to determine the number and order of modes for each task. As Pinha (2015) goes on to explain, these preferences are also independent of resource requirements by other modes. For instance,

if mode 1 requires more resources than any of the other modes, that does not necessarily equate to a shorter duration for mode 1. On the contrary, its duration will vary depending on its process, not its resource requirements. Conversely, less equipment with fewer laborers may be more efficient than the resources mode 1 calls for, but they may also be more costly, produce products of a lesser quality, or otherwise result in an outcome that is less desirable than the application of the resource requirements for mode 1.

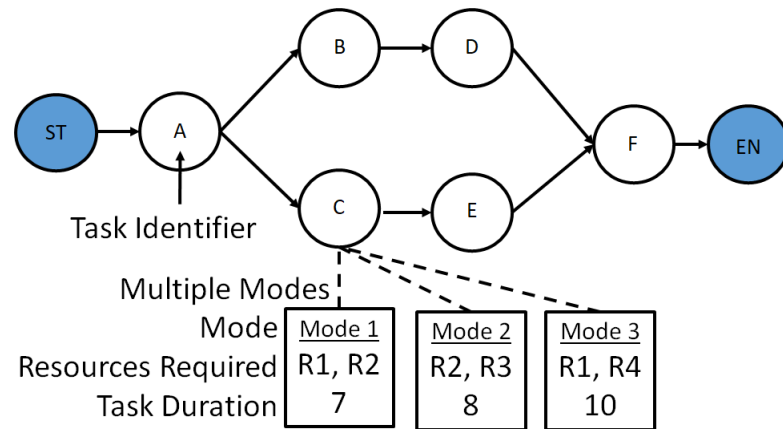


Figure 2.5. Multi-mode Resource Constrained Project Scheduling Problem (MRCPSP). Adapted from Pinha (2015).

At its heart, the consideration of a MOC's operations and the risk associated with any under-resourcing of staff levels, training, etc., is a human resources problem. In the context of a project, human resources have multiple skills, with a certain level of proficiency in that skill, or skill level, which may or may not affect the overall performance of a task. While the MRCMPSP model can accommodate flexibility in the application of multiple resources, very few approaches are capable of handling multi-skilled workers. The Combinatorial Multi-mode Resource Constrained Multi-Project Scheduling Problem (CMRCMPSP) developed by Pinha (2015) is an extension of the RCPSP that expands its execution mode category (see Figure 2.4). It allows for several project tasks to be handled simultaneously, under precedence and resource constraints, with multiple modes available for each task. It also enables the consideration for the skill level of the resources at hand, which may be made analogous to the level of risk assumed with assigning a resource to perform a given skill for a task.

While the CMRCMPSP is similar to the MRCMPSP, according to Pinha (2015), it differs

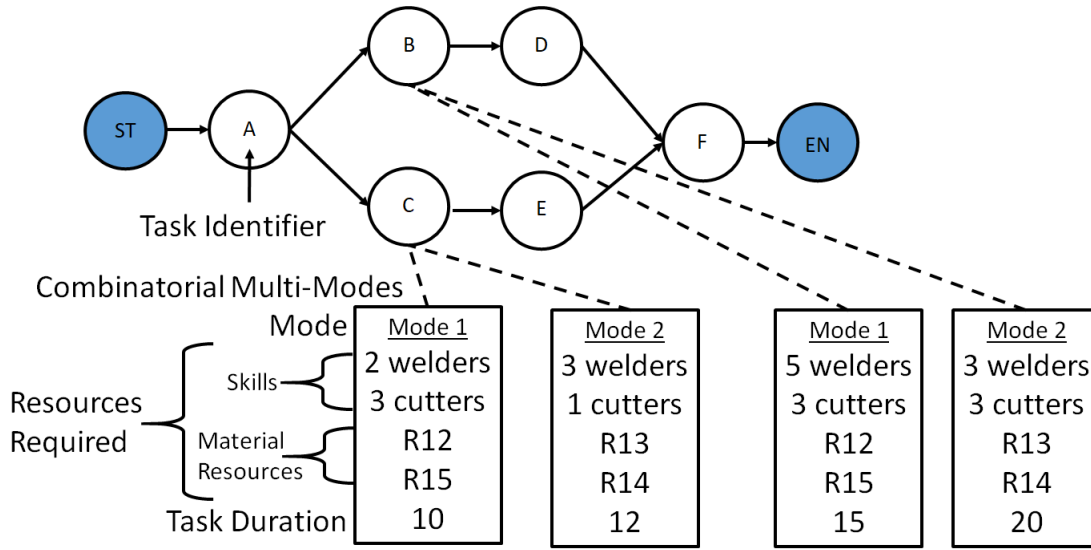


Figure 2.6. Combinatorial Multi-mode Resource Constrained Multi-Project Scheduling Problem (CMRCMPSP). Adapted from Pinha (2015).

widely from a standard multi-mode approach in the way it considers resources. In a CMRCMPSP, the quantity of modes does not solely define the multitude of options available to perform a task, and task performance may or may not depend on the resources selected. Per Pinha (2015), a mode in a CMRCMPSP is no longer a set of resources, but a set of combinational subsets of unknown resources who possess the skills needed to execute a given task but who also possess a certain level of proficiency in those skills. Figure 2.6 shows the previously discussed six-task project, now with both skill and resource requirements for each task and multiple modes to illustrate the CMRCMPSP's differences from the MRCMPSP. In task C of the CMRCMPSP example, there are two resource-defined modes provided, but actually 896 different ways to perform the task. For example, given the collection of resources, skills and associated skill levels listed in table 2.1, it can be seen that mode 1 of task C requires two of eight total welders, three of six cutters, and resources R12 and R15. Also, mode 2 of task C requires three of eight welders, one of six cutters, and resources R13 and R14. The resources with the skills of welders and cutters are the combinational subsets, with each resource bringing a certain level of proficiency in the skill it is chosen to conduct. The total number of options for each of these subsets can

be described by the binomial distribution factorial formula (Equation 2.8),

$$\binom{n}{p} = \frac{n!}{(n-p)!p!} \quad (2.8)$$

where p is the required quantity of resources with a given skill and n is the available quantity of resources in the resource pool possessing that skill. Therefore, the total possible number of choices for mode 1 of task C is given by $\binom{8}{2}welders * \binom{6}{3}cutters * 1R12 * 1R15 = 560$, and the total possible number of choices for mode 2 is given by $\binom{8}{3}welders * \binom{6}{1}cutters * 1R13 * 1R14 = 336$, for a total number of 896 different combinations of resources to perform task C, not just two. The overall effect on the performance of the task due to the various skill levels of the resources chosen, however, is a parameter analogous to risk that is decided by the project manager or other participant deemed a subject matter expert.

Until the work of Pinha et al. (2016), no exact or heuristic methods had developed an approach to solve the CMRCMPSP. Mathematical modeling of the CMRCMPSP is especially cumbersome and cannot incorporate the resource allocation or mode priority preferences of a knowledgeable project manager, so Pinha and colleagues developed a discrete event based simulation called STREAM (Short-Term Resource Allocation and Management) that generates a series of feasible solutions along with their impacts on schedule and cost.

STREAM was designed to address scheduling and resource allocation issues within the industrial naval shipyard maintenance and construction industry; therefore, it is based on the consideration of both material resources (renewable and nonrenewable) and multi-skilled human resources, all of which have their own availability schedules, associated cost for their use, and empirical priority rules set by a knowledgeable project manager. These *priority rules*, based on the work of Vanhoucke (2012) who specified four types—Task based, Network based, Critical Path based, and Resource based—are specific to each resource and are considered in priority order without any weight. This flexibility in rule application, e.g. the rules for welders do not apply to the rules for cutters and vice-versa, more accurately models the "real world", as it is common for each unit to run its own operations (Pinha 2015).

According to Pinha (2015), STREAM focuses on short-term scheduling (in terms of hours, days and weeks), which assumes the subject is a dynamic system where resources, priorities,

Resource	Skill Level	
	Welding	Cutting
R1 (worker 1)	1	2
R2 (worker 2)	1	
R3 (worker 3)	1	
R4 (worker 4)		1
R5 (worker 5)		1
R6 (worker 6)		1
R7 (worker 7)	1	2
R8 (worker 8)	1	2
R9 (worker 9)	1	
R10 (worker 10)	1	
R11 (worker 11)	1	
R12 (Machine 1)		
R13 (Machine 2)		
R14 (Machine 3)		
R15 (Machine 4)		
Quantity	8	6

Table 2.1. Resource Pool: CMRCMPSP

Each resource listed is either a human worker with an associated skill level (1-Excellent, 2-Good, etc.) for one or both skills, or a machine that is considered as its own skill. The last row, Quantity, shows that there are 8 resources that possess the skill of welding and 6 that can perform cutting in the resource pool. Adapted from Pinha (2015).

and other constraints change over time; an assumption that creates a simulation that is better suited to dealing with “real world” scheduling. As such, STREAM was created with the goal of starting tasks as early as they become eligible for execution. Therefore, if a given task were eligible to start because all of the tasks that proceed it were completed, and all required resources to perform it were available for at least one of its modes, the task would be executed immediately under the highest priority mode that has resources available at that point in time, i.e., even if the least desirable mode of a task is the only one capable of being executed at a given point in time, it will be executed. The system will not delay its execution in favor of waiting until a more desirable mode, with an affect on the schedule that is ultimately more favorable, becomes capable.

Pinha et al. (2016) describes STREAM as a discrete event simulation that utilizes a global variable (*simulation clock*) to control how the clock changes over the simulated period, and generates outputs by assigning resources to tasks through a unique queue management technique named parallel mode schedule generation scheme (PMSGs). Traditional queue management schemes cannot be applied to queue problems in a CMRCMPSP, since it is difficult to efficiently arrange queues when a single task can have nearly 900 different combinations among resources, each of which with their own priority rules.

Figure 2.7 illustrates the complexity of the queue management system for the CMRCMPSP, highlighting the multiple links that exist between modes and corresponding resources.

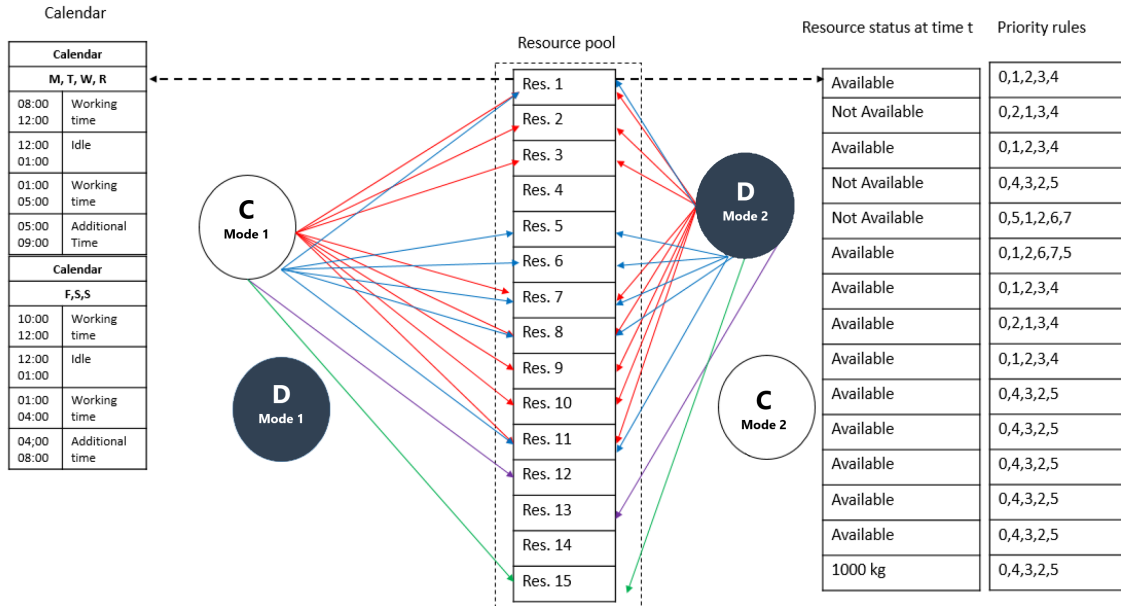


Figure 2.7. CMRCMPSP Queue Schema. Adapted from Pinha (2015).

An illustration of the parallel mode schedule generation scheme (PMSGs), displaying the connections between a mode's resource requirements and the resource pool's availability, calendar, and priority rules.

These links are the resource requirements for each mode, along with the availability calendar, usage status, and priority rules for each resource. Pinha (2015) describes the approach to the management of their assignment as follows:

1. Break individual task into their respective modes.
2. There exist no predefined or organized queue.

- (a) There is no benefit to organizing a queue ahead of time because each resource has its own priority rules. It is difficult to organize a queue if the same mode requires different resources, and their rules are in conflict to each other.
 - (b) Eligible modes from eligible tasks are listed as possible candidate for execution.
 - (c) The decision of which mode wins among eligible modes is performed at each decision point in time.
3. Verify if all the required resources for the first mode of a given task are available.
- (a) If so, verify if all the required resources for the first mode of given different task are available.
 - (b) If no common resources exist between these two modes, mode 1 of the first task is a tentative winner and further comparison is performed with another mode.
 - (c) If a common resource exists, a priority rule defines which mode wins for the first pair comparison
 - (d) If the modes are still tied up, a second priority rule is used, and so forth. The comparison is performed while modes exist. In the end, only one mode wins. (Pinha 2015)

The CMRCMPSP solution model developed by Pinha et al. (2016) does not set out to find the single scheduling solution that is “optimal” with respect to time or cost. Their methodology instead aims to provide a series of feasible solutions that incorporate a feature of resource allocation flexibility that is more akin to the realities of “real world” situations and the nature of the resources under consideration; thereby, enhancing the decision making process for the project manager.

2.2 Previous NPS Work

Other NPS works have also focused on the analysis of resource allocation in the pursuit of developing a basis for insights that articulate the justifications for the procurement of resources, or as an aide for resourcing decisions.

In 2011, Davidson (2011) published *Robust Analysis of the Joint Strike Fighter Integrated Training Center Pilot Scheduling*, to develop and analyze resourcing and scheduling courses of action that affect student pilot throughput. His work utilized a discrete event simulation model first developed by Lockheed Martin to simulate the flow of students through the

training pipeline and provide insight into the appropriate initial procurement levels for resources (simulators, classroom spaces, instructors, etc.). The integrated training center model (ITC) model process (Figure 2.8) considers student syllabi (classes, training events, required meetings, etc.) as the tasks of a network, while parameters for quantity and availability schedule for items like flight simulators, training aircraft, and instructor pilots are inputs for the resources that service the tasks. System outputs, such as resource utilization, are analyzed and used as justification for procurement or to highlight potential bottlenecks in the system.

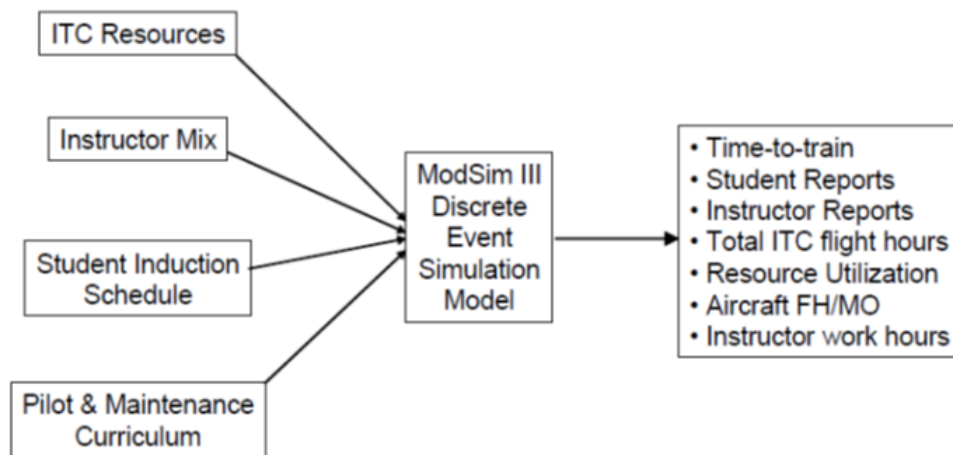


Figure 2.8. Integrated Training Center (ITC) Overview Diagram. Adapted from Davidson (2011)

A diagram of the integrated training center model simulation developed by Lockheed Martin to develop initial resourcing requirements for the joint strike fighter training center schoolhouse.

Recently, Lieutenant Chris Jones, Lieutenant Commander Ryan Hughes, and Commander Maxwell Oliver considered project scheduling of a MOC for their Naval Postgraduate School Master of System Analysis Capstone Project (Jones et al. 2018). Specifically, they created a method to quantify the risk accepted when MOC manning, training, and equipment are underfunded. Their research provides a measure of performance, the critical activity rating (CAR), that identifies the frequency a task appears on the critical path, and the tasks that can improve a project's overall completion time by adding resources that serve to reduce their individual duration. However, the model developed is based on the CPM and elements of PERT. This limited their analysis to a single project that is under precedence, but not resource constraints. Further research was recommended that could consider multiple

simultaneous projects and limited resources to better model the impacts of staff resources on a MOC’s performance over sustained operations.

2.3 Our Contribution

We seek a representation of MOC staff operations that is specific enough to generate realistic solutions and reduce that solution search space to a manageable size. At the same time, that representation must also be sufficiently general enough in nature to allow for the flexibility needed to model the “real world” characteristics of human resources performing administrative tasks. Essentially, we require a RCPSP that is multi-project, has combinatorial multiple modes with renewable resources, and is also non-preemptive; see Figure 2.9. This allows for the consideration of situations where the MOC staff must conduct multiple concurrent operations under a fluid set of priorities, with a staff of renewable human resources that may be multi-skilled and/or multi-“hatted” (assigned with more than one responsibility within the organization). This staff also possesses varying levels of proficiency in their given skills (attributable to degrees of training and experience) and are performing tasks that generally will not be interrupted or restarted before their initially scheduled time of completion.

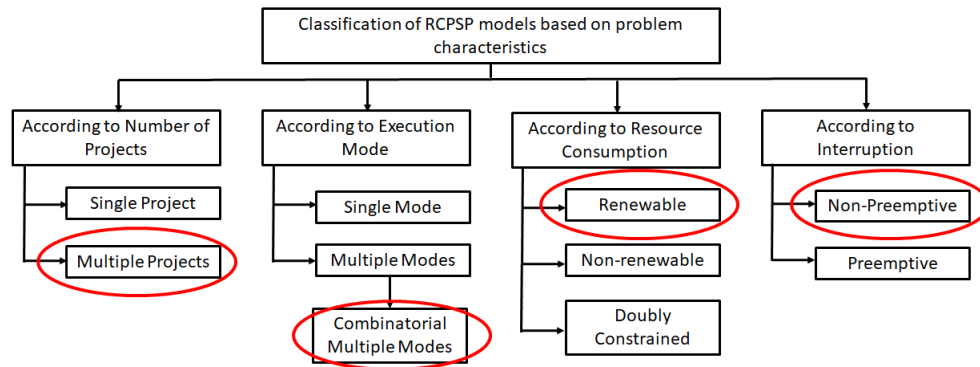


Figure 2.9. RCPSP Classification of the Problem Model. Adapted from Gen et al. (2008)

The classification of the RCPSP model considered for the problem model of MOC staff operations. The model is a RCPSP that is multi-project and non-preemptive, with combinatorial multiple modes and renewable resources.

Ultimately, we present an adaptation of the CMRCMPSP model solution developed by

Pinha (2015) that is designed to mimic the decision making process and preferences of a project manager.

THIS PAGE INTENTIONALLY LEFT BLANK

CHAPTER 3:

Model and Methodology

In this chapter, we introduce the terms and notation that make up our lexicon. We then provide an example of our problem model along with a mathematical formulation that explains assumptions about tasks, resources, and their associated constraints. This is followed by a description of our solution method, and concluded with a proof of concept analysis of an example model.

3.1 Problem Statement

We build on the standard terms introduced in the last chapter.

The basic building block of our model is a *task*, which represents a unit of work that cannot be further divided. A *project* is a collection of tasks that possess precedence relationships, duration and resource requirements to accomplish work. At the highest level, a *program* is a collection of projects to be considered.

There is a variety of constraints imposed on a task. Most simply, a *predecessor* is a task that must be completed before the given task, and a *successor* is a task that cannot be started prior to the completion of the given task.

In general, a task requires one or more *resources* to perform its work. In turn, a resource can produce one or more different types of work, as dictated by a *skill*. A resource can have multiple skills, each with its own *skill level* representing the competence of the resource for that skill.

The way in which resources are used to perform the skills required to complete a task is called its *mode*. A task may have more than one possible mode for completion.

Thus, given (1) a program of projects, each consisting of a collection of tasks having their own precedence relationships as well as resource and skill requirements, and (2) a pool of resources, each with their own skill levels, our challenge is to assign resources and schedule tasks in a manner that minimizes the overall finish time.

The appendix provides a glossary of terms used throughout this chapter.

3.2 Motivating Example

Figure 3.1 is an example program consisting of three simultaneous projects that share a single Start (ST) and End (EN) node, but are separated by dotted lines in the figure. Each project is presented as a task-on-node network where each task contains at least one mode with its associated required skills and duration. For example, task C of project 2 has two modes: mode 1, which requires two resources with skill 1 and four resources with skill 2 for a duration of 8 units of time, and mode 2, which requires two resources with skill 1 and two resources with skill 3, for a duration of 10 units of time. A summary of the example modes is provided in Table 3.1.

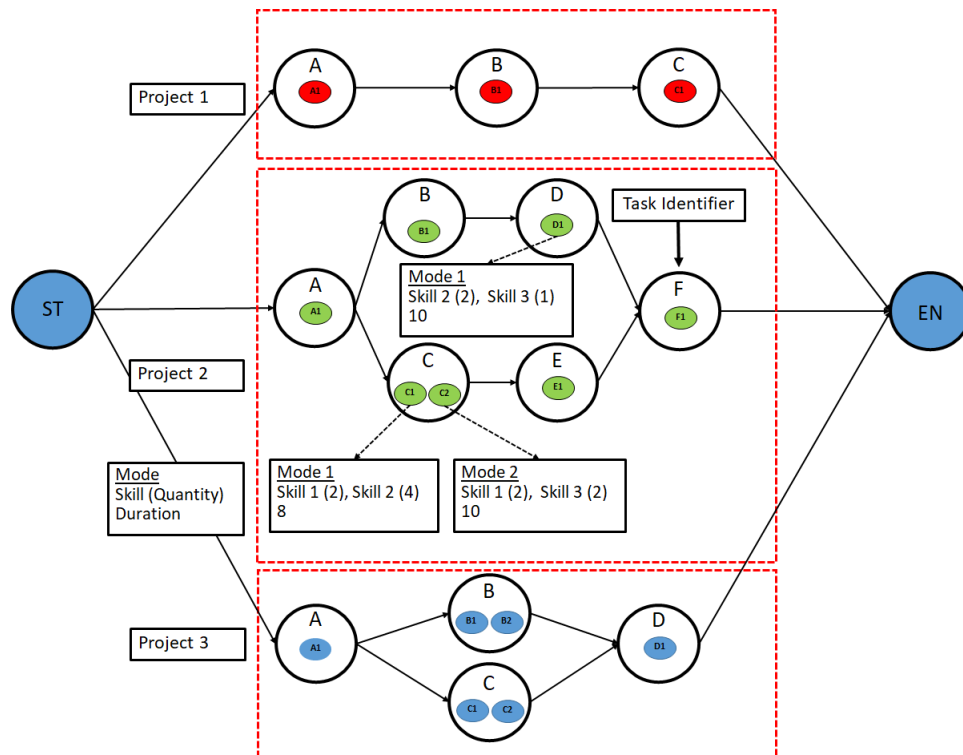


Figure 3.1. Example Program Model. Adapted from Pinha (2015).

Example task-on-node representation for the multiple projects that make up a program. Each project is inside a box of dotted lines and consists of tasks containing modes with associated skill requirements and duration.

Project	Task	Mode	Duration	Required Skills		
				Skill 1	Skill 2	Skill 3
2	C	1	8	2	4	0
2	C	2	10	2	0	2
2	D	1	10	0	1	2

Table 3.1. Example Program Model Modes

Sample of modes from project 2 of the example program model in Figure 3.1. The modes for tasks C and D are shown for illustrative purposes. Each mode is associated with a project and task, has a duration, and a quantity required for each skill.

As an aide for the management of projects in the naval shipyard maintenance and construction industry, the simulation solution to the CMRCMPSP model developed by Pinha (2015) is primarily concerned with the effects that the application of limited resources have on a program’s schedule and cost. On a practical level, these resources are either solid, construction-like materials or skilled labor performing intensive, construction-related work in an industrial environment. That is to say that the types of skills this labor force perform are generally physical in nature and occupy the full capacity of that laborer’s time and attention for the duration of their assignment. In contrast, our adaptation of the problem model must consider the realities of a labor force that primarily perform administrative-related tasks in an office environment. This type of work (conduct of research, attendance of coordination meetings and briefings, drafting of correspondence and reports, etc.) is more cognitive than physically demanding, allowing for the opportunity for a human resource to contribute to more than one concurrent effort.

We refer to a human resource’s potential ability to participate in multiple efforts simultaneously as their level of *bandwidth*. For the purposes of this work, that bandwidth is considered to be independent of the skills associated with the resource, e.g., a resource with a bandwidth level of 2 can perform exactly two skills simultaneously, regardless of which skill of its skill set it is performing. The only restriction our work places on this application of bandwidth is within a mode. A resource is not allowed to be assigned to a mode to perform the same skill more than once. However, a resource is able to be assigned to a

mode more than once to perform different skills, not to exceed the level of its bandwidth. The assumption being that if a mode requires a certain number of a type of skill, that number represents the amount of individual personnel, working in cooperation, needed with that skill; a number that cannot be physically reduced by increasing the contributions of singular resources with multiple levels of bandwidth.

Table 3.2 is the collection, or pool, of resources available to the program of Figure 3.1 for use in its execution. Each resource possesses a skill level for the set of skills (a value of 0 or none denoting that the resource does not possess that skill), a skill count for the number of skills it possesses (its multi-skill capacity), a value for bandwidth that is at least 1, and a set of user-defined priority rules listed in order of priority.

Resource	Bandwidth	Skill Count	Priority Rules	Skill Level		
				Skill 1	Skill 2	Skill 3
R1	2	3	1,2,3	1	3	2
R2	1	2	1,2,3	2	1	
R3	1	2	1,3,2		1	
R4	1	2	1,2,3		4	1
R5	2	3	1,2,3	3	4	1
R6	2	2	1,2,3	1	4	
Priority Rules						
1 = Shortest Duration, 2 = Fewest Resource Requirements, 3 = Best Skill Level						

Table 3.2. Example Resource Pool

Example collection of resources. Each resource possesses a value for bandwidth denoting the number of skills a resource can conduct simultaneously, a skill count for the number of skills it possesses, a set of user-defined priority rules listed in order of priority, and a skill level for the set of skills (1-Excellent, 2-Good, 3-Reasonable, 4-Poor, blank-none)

3.3 Mathematical Model

A mathematical formulation is useful for defining the problem model and any assumptions concerning its associated tasks, resources, and constraints. Pinha (2015) presents a formulation for his CMRCMPSP model which we have adapted as a basis for this work.

Index Sets

$i \in I$	projects; where $1 \leq i \leq NP = I $
$j \in J_i$	tasks in project i (alias k); where $1 \leq j \leq NA_i = J_i $
$(k, j) \in P$	precedence relationships, denoting that task k must be completed before the start of task j
$m \in M_{i,j}$	possible modes for task j of project i ; where $1 \leq m \leq NM_{i,j} = M_{i,j} $
$s \in S$	skills, where $1 \leq s \leq NS = S $
$s \in S_{i,j,m} \subseteq S$	subset of skills required for executing task j and project i using mode m ; where $1 \leq s \leq NS_{i,j,m} = S_{i,j,m} $
$r \in R$	resources; where $1 \leq r \leq NR = R $
$r \in R_{i,j,m,s} \subseteq R$	subset of resources with skill s required if operating on mode m for task j of project i
t	time interval, where $0 \leq t \leq T$

Parameters [Units]

$d_{i,j,m}$	duration of task j for project i , if operated using mode m [time]
$c_{i,j}$	tardiness cost of task j of project i per unit of time [cost/time]
$r_{i,j,m,s}$	quantity of resources with skill s required to operate on mode m for task j of project i [resources]
$BW_{r,t}$	bandwidth level of resource r at time t [resources]

Variables

$x_{i,j,m,t}$	assignment of task start time [binary] = 1, if task j of project i , operating on mode m , is started at time t = 0, otherwise
$TC_{i,j}$	Total tardiness cost for task j of project i
$st_{i,j}$	Start time of task j , project i
$ft_{i,j}$	Finish time of task j , project i
$AR_{s,t}$	Quantity of renewable resources available with skill s at time t
$E_{i,j}$	Eligible time to start for project i , task j
$u_{i,j,m,s,r,t}$	indicator of resource availability [binary] = 1, if resource r with skill s required is available at time t to operate on mode m for task j of project i because it is not being used for any other mode among all the projects = 0, otherwise

Formulation

$$f = \min_x \left(\sum_{i \in I} \sum_{j \in J_i} TC_{i,j} \right) \quad (3.1)$$

Subject to:

$$\sum_{m \in M_{i,j}} x_{i,j,m,t} = 1 \quad \forall j \in J_i, \forall i \in I, \forall t \in [0, T] \quad (3.2)$$

$$x_{i,j,m,t} \cdot t \geq st_{i,j} \quad \forall j \in J_i, \forall i \in I \quad (3.3)$$

$$ft_{i,j} \geq st_{i,j} + d_{i,j,m} \quad \forall j \in J_i, \forall i \in I, \forall m \in M_{i,j} \quad (3.4)$$

$$r_{i,j,m,s} \cdot x_{i,j,m,t} \leq AR_{s,t} \quad \forall i \in I, \forall j \in J_i, \forall m \in M_{i,j}, \forall s \in S_{i,j,m}, \forall t \in [0, T] \quad (3.5)$$

$$AR_{s,t} \leq \sum_{r \in R_{i,j,m,s}} u_{i,j,m,s,r,t} \quad \forall i \in I, \forall j \in J_i, \forall m \in M_{i,j}, \forall t \in [0, T] \quad (3.6)$$

$$\sum_{i \in I} \sum_{j \in J_i} \sum_{s \in S_{i,j,m}} u_{i,j,m,s,r,t} \leq BW_{r,t} \quad \forall r \in R, \forall m \in M_{i,j}, \forall t \in [0, T] \quad (3.7)$$

$$\sum_{m \in M_{i,k}} t \cdot x_{i,k,m,t} \geq \sum_{m \in M_{i,j}} (t + d_{i,j,m}) \cdot x_{i,j,m,t} \quad \forall (j, k) \in P, \forall i \in I, \forall t \in [0, T] \quad (3.8)$$

$$E_{i,j} = \max_{(k,j) \in P} \{ft_{k,j}\}, \quad \forall i \in I, \forall j \in J_i \quad (3.9)$$

$$TC_{i,j} \geq c_{i,j} \cdot \left(\sum_{t \in T} \sum_{m \in M_{i,j}} (t - E_{i,j}) \cdot x_{i,j,m,t} \right) \quad \forall i \in I, \forall j \in J_i \quad (3.10)$$

Discussion

The objective function in Equation 3.1 aims to minimize the tardiness cost for each task across all projects, which is analogous to limiting the amount of time between when a task becomes available and when it is actually started. The constraint in Equation 3.2 ensures that all tasks are scheduled only once, by allowing for the selection of only one mode. Equation 3.3 calculates the start time for all tasks, while Equation 3.4 calculates their finish times. The constraint of Equation 3.5 ensures that the number of resources assigned at a given time t are less than the amount that are available at that time, while Equation 3.6 calculates the amount of resources with skill s are available. Equation 3.7 limits a resource to be assigned to no more than its bandwidth. Equation 3.8 reflects the precedence relationships among the tasks of all projects. Equation 3.9 computes the eligible start time for each task, and Equation 3.10 calculates the associated tardiness cost for each task.

Alone, this formulation describes a set of feasible solutions for the assignment of resources to modes of tasks that meet the general restraint requirements for our category of RCPSPs.

However, in addition to being computationally cumbersome for all but the simplest of network models, this strictly mathematical solution method is unable to incorporate the network-specific restrictions that reflect the preferences of the project manager that we wish to impose. While it does account for the physical limitation of not allowing a multiple bandwidth-level-possessing resource to apply the same skill to a mode more than once for a given time, it does not reflect the use of any resource allocation flexibility tools, such as user-defined mode preferences or resource priority rules. It is for these reasons and more that Pinha et al. (2016) developed STREAM as a solution method for their problem model and why we have also decided to base our solution on a discrete event simulation as well.

3.4 Solution Method

Pinha (2015) recognizes that simulations utilize a wide array of methods to mimic the behavior of complex systems. The focus of this work is on dynamic systems with deterministic task durations, where the states of the system change at discrete points in time.

Our solution method's primary goal is to start tasks as early as they are able to be executed, subject to the restrictions of user-defined preferences. The Dynamic Resourcing Allocation Analysis Simulation Tool Kit, or DRAASTK (see Figure 3.2), accomplishes this by converting problem model inputs for resources and tasks into model entities. These entities inherit attributes of the problem model which are then processed through the states of *pending*, *eligible*, and *capable*, then either *extracted* or *scheduled* and *processed*. These transitions are based on both intrinsic and user-defined rules, resulting in an output that provides a suggested schedule for assigning resources and completing tasks, while offering insight into the benefits of resource flexibility and the risks assumed to accomplish the presented schedule.

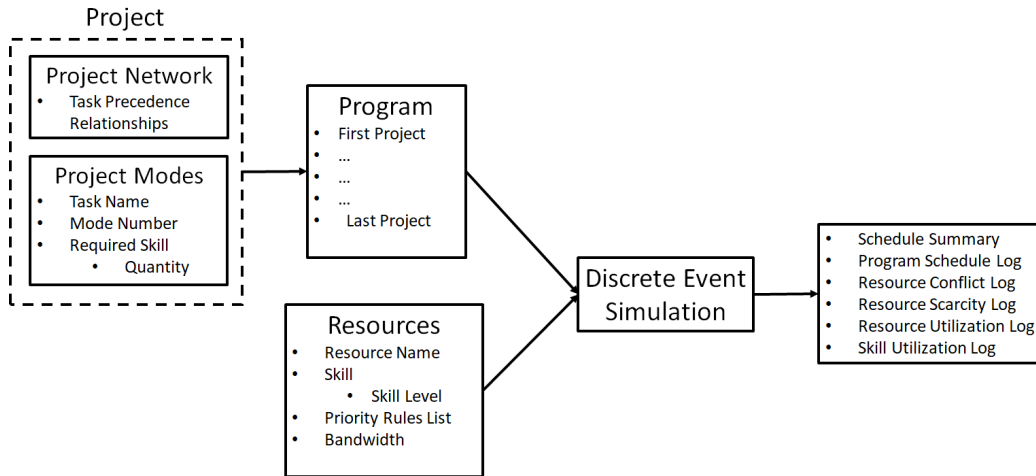


Figure 3.2. Dynamic Resourcing Allocation Analysis Simulation Tool Kit (DRAASTK) Overview Diagram

The Dynamic Resourcing Allocation Analysis Simulation Tool Kit (DRAASTK) accepts inputs for the project networks and modes that compose a program, as well as the pool of resources available to service the program. It then uses those inputs in a discrete event simulation to provide outputs that give insights into project schedules and resource allocation and utilization.

3.4.1 Model Primitives

The model entities which inherit the attributes of the problem model are known as “class objects.” An illustration of the attributes inherited by these class objects is provided in Figure 3.3. The resource class objects possess information concerning their skill sets and associated skill levels, as well as their ordered list of applicable priority rules, level of bandwidth, and skill count. The task objects, however, are further subset into task-mode (just referred to as “mode”) objects which have information regarding correlated tasks, projects, modes, skill requirements, and duration.

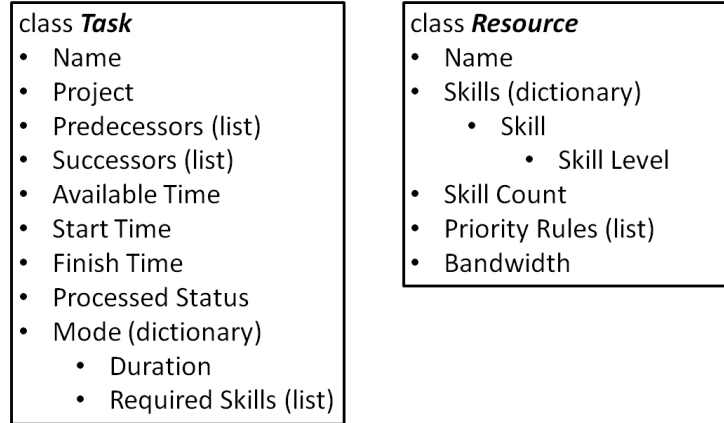


Figure 3.3. Class Object Attributes

The task and resource class objects both provide parameters of the problem model to and accept values of variables from the simulation. They act as storage mechanisms for the system's calculations and are the sources of data for the system output. The task class provides parameters for identification, modes and resource requirements while accepting concerning scheduling. The resource class provides parameters for identification and skills, while accepting changes to bandwidth.

3.4.2 Model Design

Utilizing the program of Figure 3.1 and the resource pool of Table 3.2 collectively as an example model, we provide Figure 3.4 to illustrate the design of our simulation solution method.

Simulation Process

DRAASTK's simulation process is controlled by the system clock, which dictates the values for the points in time where the system evaluates a state change. The system clock, in turn, is controlled by the global variable *event time* which belongs to a set of discrete values referred to as the *event time log*.

At the beginning of the simulation, event time is initialized to 0 and added to the event time log set. To start the process, the lowest value in the event time log set is removed and designated the current value for event time. At the time point of event time, and starting with the state of *pending*, the mode objects are evaluated to determine if all predecessor tasks are completed, making them *eligible* to be executed. Eligible modes are then processed with

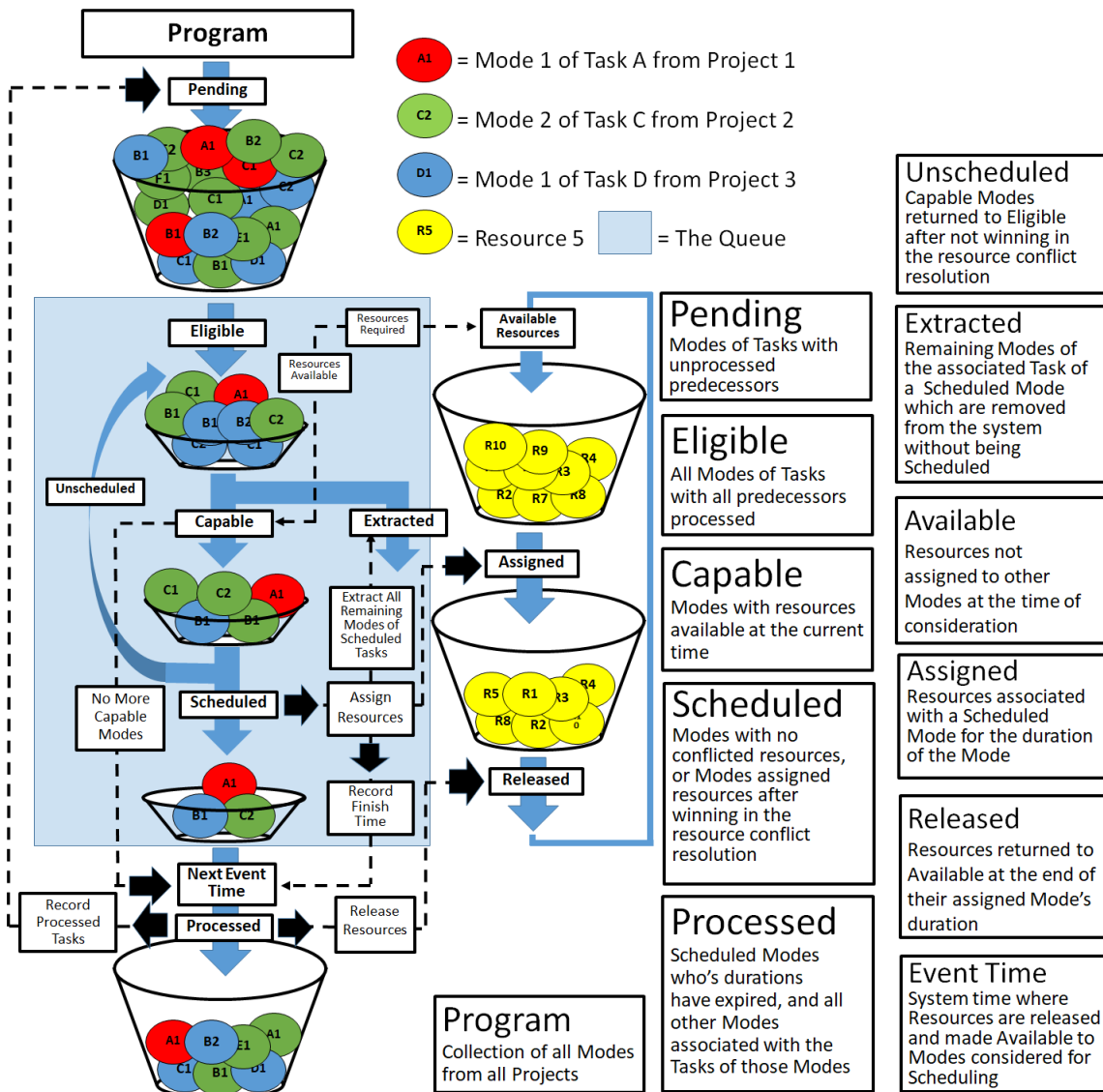


Figure 3.4. Dynamic Resourcing Allocation Analysis Simulation Tool Kit (DRAASTK) Model Design

A conceptual model of DRAASTK's discrete event simulation process. The process is controlled by a time variable, which dictates the values for the points in time where the system evaluates a state change for task class objects. These task objects are processed through the states of *pending*, *eligible*, and *capable*, and then either *extracted* or *scheduled* and *processed* based on a set of intrinsic and user-defined rules until no task objects remain in the system.

the resource objects to assign resources to each of the mode's required skills according to the

system's intrinsic and user-defined priority rules. Eligible modes whom successfully assign resources to every one of their required skills change states to *capable*. Capable modes are then evaluated to resolve any conflicts in resource assignments among them, resulting in a single winner. The modes that did not win the conflict resolution are considered *unscheduled* and shifted back to eligible. The winner mode, along with any capable modes without resources in conflict, are transitioned to *scheduled*.

The scheduled modes are evaluated with respect to the tasks to which they belong. For any scheduled modes that belong to the same task, the mode with the highest priority is kept, while the other scheduled modes of that task are designated as *extracted* and removed from the system. Additionally, for any task that has a scheduled mode, all eligible modes of that task are designated as extracted and removed as well. For those mode objects that are *scheduled*, their finish times are noted and added to the event time log set. The resource objects assigned to them are also made unavailable for the duration of their associated mode.

Using the remaining set of available resources, eligible modes are reevaluated to determine if any are able to become capable and ultimately added to the set of scheduled modes. This reevaluation cycle continues until no eligible mode is able to become capable with the resources available. Once this cycle completes, the unique values for the finish times of scheduled tasks are added to the event time log set, and information about the tasks, modes, and their assigned resources are noted for the system output. Finally, those tasks whom have a finish time equal to the current event time are transitioned to *processed* and their associated resources are made available for reassignment.

The process resets by removing the lowest value in the event time log set and designating it as the current value for event time. The process continues to cycle, adding information to the system output along with new values to the event time log, until the point where no values remain in the event time log because all tasks are completed.

The following sections provide detailed descriptions for the components of the simulation's process.

Pending Modes

Once the simulation translates the components of the problem model into the manipulable formats of class objects, it only considers the modes of tasks, rather than the tasks themselves from then on. These modes inherit the precedence relationships of their respective tasks and are considered *pending* until ready for further consideration.

Eligible Modes and Resource Selection

Once all predecessors of a given task are completed (in the state of “processed”), that task is categorized as *eligible*, as well as all of its associated modes. Eligible modes would then join a process akin to a queuing system for resource selection (see Figure 3.5), with the current event time noted as the *Available Time* for the associated task objects of each mode entering the queue for the first time. During resource selection, each eligible mode creates their own preliminary list of assignments from among the collection of those resources having a level of bandwidth of at least 1, known as available resources. To maximize resource allocation flexibility, this selection process is subject to a set of ordered priority criteria; 1) number of positions to fill for a skill in descending order, 2) the resource with the best skill level for the given skill, and 3) the resource with the lowest skill count. These criteria are designed to limit the risk to the metrics of the system from resource selection by choosing the resources with the best possible skill level, while at the same time maximizing selection flexibility by assigning the resources with the least amount of flexibility to the positions of the skills within the mode with the highest resource demand.

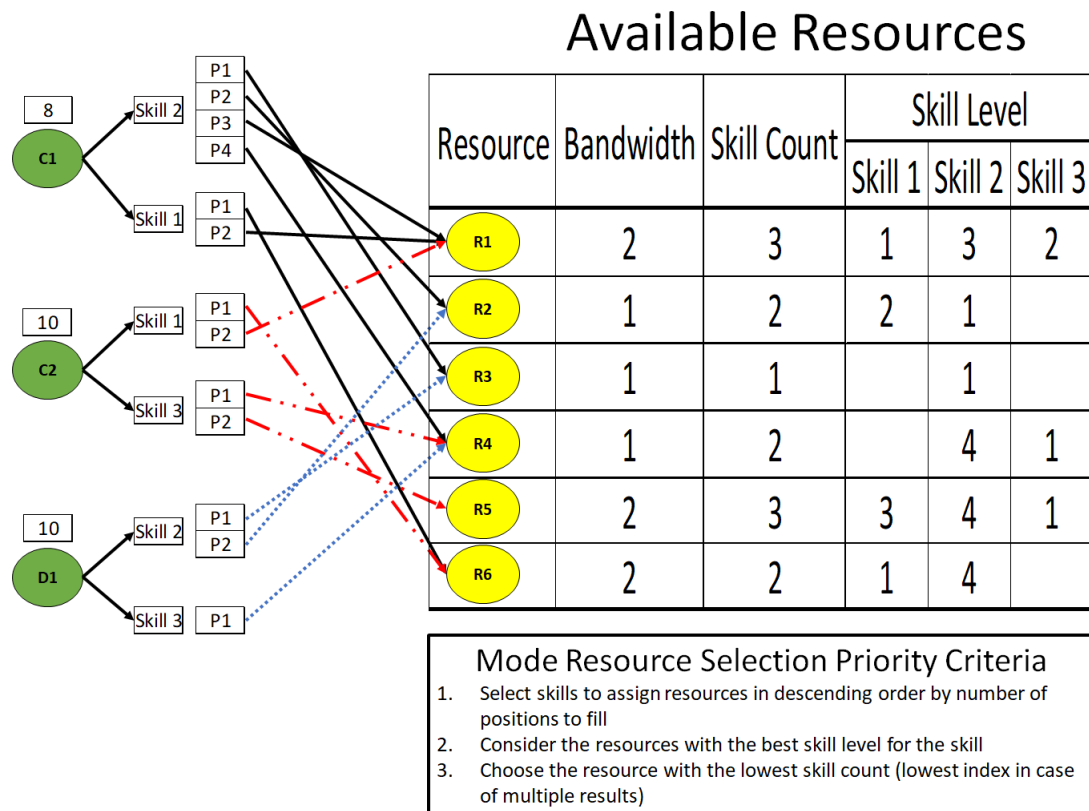


Figure 3.5. Resource Selection

Illustration of the resources selection process using the modes shown in project 2 of Figure 3.1. Each eligible mode selects resources to perform the skills they require for their duration, under a set of priority criteria. For example, mode 1 of task C requires two positions (P1 and P2) to be filled with resources possessing skill 1 and four positions (P1 to P4) to be filled with skill 2. In accordance with the mode resource selection priority criteria, the four positions of skill 2 are addressed first since it has the larger requirement for resources. As the only resources with a skill level of 1 (best rating) for skill 2, resources R3 and R2 are selected to fill P1 and P2, respectively, where R3 is chosen over R2 to fill P1 due to its lower skill count. Resource R1 is selected to fill P3 with a skill level of 3, and P4 chooses R4 over R6 due to R4 having a lower index and both skill level and skill count are equal to R6. The positions for skill 1 are then addressed with R6 selected for P1 with a skill level of 1, and P2 selecting R1 where R1 still possesses 1 level of bandwidth after being chosen to fill P3 for skill 2.

Capable Modes and Conflict Resolution

The eligible modes that are able to construct a complete preliminary list of assigned resources from the pool of available resources are categorized as *capable*. Those modes that are unable to construct a list of assigned resources from the available resource pool are added to a *schedule delay log*, along with the skills that they are unable to resource and the current event time. Since each mode conducts an independent selection of resources, this process could result in conflicts between modes who select the same resources. A round of conflict resolution that is based on an ordered list of priorities unique to the resource in conflict, as illustrated in Figure 3.6, is conducted to determine the single “winning” mode among all conflicted modes.

Conflicts			
Resource	Priority Rules	Modes in Conflict	Preliminary Winner
R1	1,2,3	C1, C1, C2	C1
R2	1,2,3	C1, D1	C1
R3	2,1,3	C1,D1	D1
R4	1,2,3	C1,C2,D1	C1
Winning Mode			C1

Priority Rules	
Intrinsic	Lowest Mode of common Task
User-Defined	1 = Shortest Duration
	2 = Fewest Required Resources
	3 = Best Skill Level

Figure 3.6. Conflict Resolution

An example of the results for a conflict resolution among common resources. During conflict resolution, the modes associated with each resource in conflict are sequentially compared with respect to the intrinsic rule for mode order (the better mode of modes of the same task have priority) and the parameters of the resource's ordered list of priority rules to determine a preliminary winner mode for each resource. The ultimate winning mode is designated as the mode among the preliminary set of winner modes that occurs the most frequently. For example, the only modes in conflict for R1 belong to the same task. Therefore, the intrinsic rule for mode order dictates C1 as the preliminary winner for R1. For R3, the second rule concerning fewest resources required has top priority and results in D1's determination as the preliminary winner (three total number of positions to fill as opposed to the six that C1 requires). The ultimate winning mode is determined after resolution reveals C1 to be a preliminary winner three times compared to once for D1.

The priority rules for each resource in conflict, and the associated set of modes which have chosen that resource to fill a position for one of their required skills, is considered during conflict resolution. As discussed in Section 2.1.3, priority rules are an ordered list of user-defined preferences for the assignment of resources to tasks, based on any metric

the user deems relevant. These rules are in addition to the inherent priority of mode order, however, since mode order is the primary priority rule for all resources regardless of any user-defined inputs.

Starting with the first rule in the priority list, each mode in the set of modes for the given resource in conflict is compared. If a single winner cannot be determined by use of the first rule, all of the modes for which the first rule resulted in a tie for are considered with regard to the next rule in the priority list. Ties among the modes are resolved in this manner until either a winner is determined, or the list of priority rules is exhausted. In the event that there still exist a tie after the entire priority list is evaluated, the mode with the lowest index in the list of remaining modes is designated the winner. The winning modes for each of the resources in conflict comprise a set of preliminary winners, among which the mode that occurs the most is designated as the “winning” mode. In the event of a tie for the number of occurrences of a mode in the list of preliminary winners, the mode with the lowest index in the list of tied preliminary winners is designated the “winning” mode. The remaining capable modes, from the group of competing modes that were not designated as the “winning” mode, have their preliminary list of assigned resources discarded and are reclassified as eligible once again; a status we refer to as *unscheduled*.

Scheduled Modes

A group of capable modes that are either not in competition for resources or is the designated “winning” mode, are considered in respect to the tasks to which they belong. Since only one mode of every task is allowed to be executed, this consideration is necessary to avoid scheduling resources for more than one mode of a single task. It is conducted in two parts: First, any modes in the group of currently eligible modes that belong to the same task as any modes in the remaining group of modes with assigned resources are extracted from the system. Second, for any modes in the group whom belong to the same task, only the lowest mode number (highest priority mode) is retained. The remaining modes of that task in the group have their preliminary list of assigned resources discarded and are excluded from any further consideration by the simulation. This is a status we refer to as *extracted*.

The resulting group of capable modes are now considered *scheduled*. The attributes of their associated tasks are updated with a start time equal to the current event time; a duration of that associated with the mode; and a finish time calculated from the start time and duration.

The bandwidth for each assigned resource is then deducted by the number of positions that resource is serving among the group of scheduled modes.

Once a group of modes is scheduled, the resource selection process is repeated for the remaining eligible modes (see Figure 3.7). The cycle of adding capable modes to the group of scheduled modes and updating the bandwidth levels of the newly assigned resources, repeats until no modes are left able to create a schedule from the available resource pool.

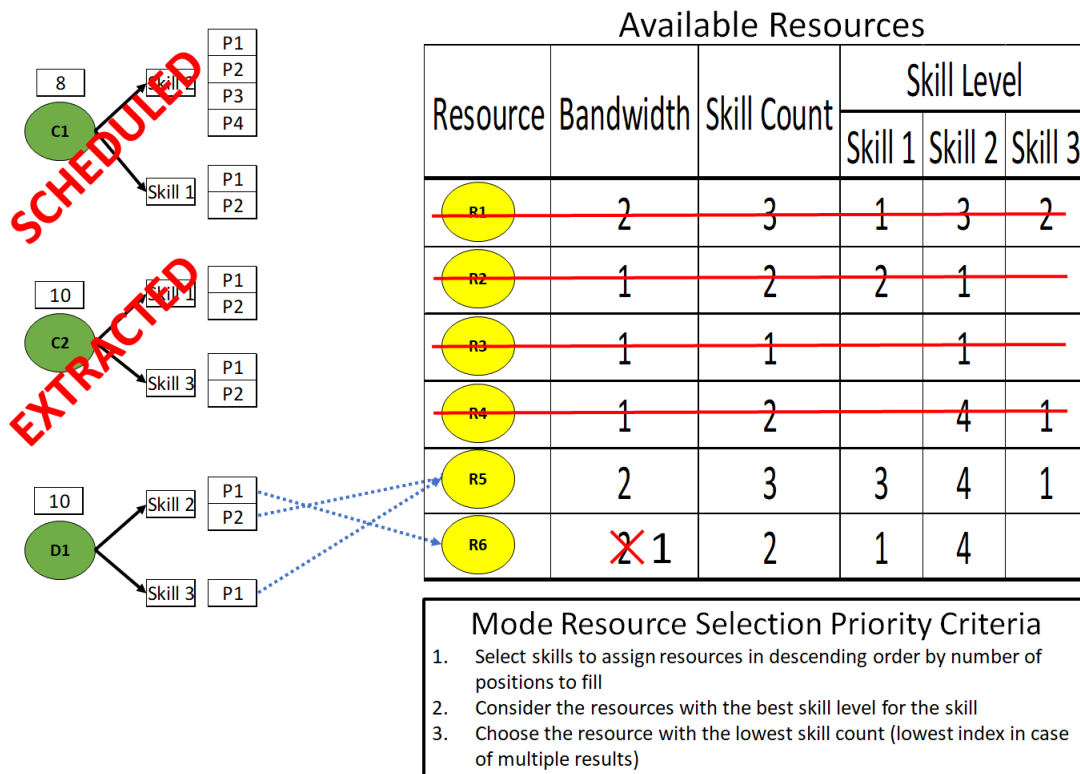


Figure 3.7. Resource Selection Reevaluation

An illustration of the reevaluation of eligible modes for scheduling at a single point in time. Once a group of eligible modes is scheduled and the bandwidth levels of the applicable resources are updated, the system continues to reevaluate the remaining eligible modes that have not been either extracted or scheduled until no mode is able to fill all of its required skills with the available resources. In this example of a reevaluation, the bandwidth levels of resources R1 through R4 were previously exhausted, while mode 1 of task C was scheduled. Mode 2 of task C was extracted, and mode 1 of task D is able to create a schedule because R6 still retains one level of bandwidth available for assignment.

Processed Modes

As the simulation progresses, the resource bandwidth assigned to scheduled modes remain unavailable for utilization until the point in time where the current event time equals the finish time of the mode's associated task. At this point in time, a task becomes *processed*. The mode of a processed task is considered to have completed its execution and all of the resource bandwidth occupied by it is returned to the applicable resource.

3.4.3 Model Outputs

The outputs of DRAASTK not only provide a suggested program schedule with accompanying resource assignments and associated skill levels, but information that gives insights into both resource and skill utilization, as well as potential resourcing shortfalls that cause project completion delays. These outputs include data sheets for logs of schedule summary, resource assignment, resource conflicts, resource scarcity, resource utilization, and skill utilization. A description of each output follows.

Schedule Summary. The schedule summary (Table 3.3) provides a summary of each task's available, start, finish, and delay time as well as the mode utilized and the average value for the skill level of the resources used to create the schedule. This summary not only gives the project manager a review of the overall program schedule, but displays the average skill level of the resources needed to achieve that schedule. This average skill level gives insight into the possible risk assumed by the project manager in order to complete the project as scheduled.

Project	Task	Mode	Average Skill Level	Available Time	Start Time	Finish Time	Delay Time	Slack Time	Impact
2	A	1	1	0	0	2	0	0	0
2	B	1	1	2	2	5	0	2	0
2	C	1	1.83	2	5	13	3	0	3
2	D	1	3	5	5	15	0	0	0
2	E	1	1	13	13	16	0	0	0
2	F	1	1	16	16	22	0	0	0

Table 3.3. Example Schedule Summary Log.

The schedule summary provides the overall schedule for each task, the average skill level for the skills used to execute a task, and the impact that task's delay has on the overall project's completion time. For example, task C in this schedule has 0 units of slack available and is delayed by 3 units of time, causing a 3-time unit delay in the overall project schedule. Additionally, while task D has no delay, it only achieves this schedule with an average skill level of 3.

Additionally, the schedule summary provides the slack time for each task as derived from a critical path method analysis based on the duration of the first mode. As described in Section 2.1.1, a critical path method evaluation of a project yields not only the best overall possible completion time for a project, but also the earliest and latest start and finish times, as well as available slack time for each task in the network based on precedence relationships and task durations. Without resource constraints, the tasks with no available slack time (critical task) become the most significant, as any delay in their execution impacts the overall completion time of the project. However, in the presence of limited resources, the identification of critical task is as dependent on resource availability as it is on precedence and duration. In order for a resource-constrained project to achieve the completion time of the CPM analysis, all tasks in that project must not be delayed any longer than their calculated slack time. Otherwise, those tasks become critical and impact the overall project's completion time. The listed slack time is compared to the corresponding delay time of the scheduled task to calculate an "impact" time. If the delay time is greater than the slack time, this impact time is the amount of delay to the project's completion time due to the scheduling of that task. This value can then be used to identify focus areas for resourcing efforts.

Program Log. An entry for the program log (see Table 3.4) is recorded for each processed task. Values for each field are derived from either the class object attributes at the time the task becomes processed, or calculated from system variables. Information for each task in the program includes: task name and project; available, start, and finish times, as well as duration and delay time; and resourcing information to include resource name, remaining bandwidth, skill performed, and associated skill level.

Project	Task	Mode	Resource	Bandwidth	Skill	Skill Level	Duration	Available Time	Start Time	Finish Time	Delay Time
2	C	1	R3	0	Skill 2	1	8	2	5	13	3
2	C	1	R2	0	Skill 2	1	8	2	5	13	3
2	C	1	R1	0	Skill 2	3	8	2	5	13	3
2	C	1	R4	0	Skill 2	4	8	2	5	13	3
2	C	1	R6	0	Skill 1	1	8	2	5	13	3
2	C	1	R1	0	Skill 1	1	8	2	5	13	3
2	D	1	R6	0	Skill 2	4	10	5	5	15	0
2	D	1	R5	0	Skill 2	4	10	5	5	15	0
2	D	1	R5	0	Skill 3	1	10	5	5	15	0

Table 3.4. Example Program Schedule Log

Example Program Schedule Log. It provides information concerning task available, start, and finish times, as well as any amount of delay time. It also includes the resources assigned to serve each skill and their associated skill levels.

Resource Conflict Log. The resource conflict log provides information for each capable mode that is unable to be scheduled at its time of evaluation due to resource conflicts. For each resource displayed, both the skill the mode attempted to assign it to perform and its associated skill level are listed, along with the level of bandwidth it possesses after that period of evaluation. This level of bandwidth gives insight into the specific resources, and by association skill, whose unavailability prevented the scheduling of the mode at that time.

Project	Task	Mode	Evaluation Time	Resource	Bandwidth	Skill	Skill Level
2	C	1	2	R3	0	Skill 2	1
2	C	1	2	R2	0	Skill 2	1
2	C	1	2	R1	1	Skill 2	3
2	C	1	2	R4	0	Skill 2	4
2	C	1	2	R6	1	Skill 1	1
2	C	1	2	R1	1	Skill 1	1

Table 3.5. Example Resource Conflict Log

The resource conflict log provides information for each capable mode that is unable to be scheduled at its time of evaluation due to resource conflicts.

Skill Scarcity Log. The skill scarcity log (Table 3.6) provides the project manager with specific insights into the program's skill deficiencies. It displays information for each mode that is unable to be scheduled, and the required skills not available at the time of its consideration. The unavailability of these skills is not only due to the resource conflicts where another mode takes priority at the time of evaluation like those listed in the resource conflict log, but also to the scarcity of the skill. In conjunction with the schedule summary and resource conflict logs, the resource scarcity log provides inputs into possible changes to resource availability schedules and training that would add to the flexibility needed to address any skill shortfalls, and improve the program's overall performance.

Project	Task	Mode	Skill	Evaluation Time
2	C	1	Skill 2	2
2	C	1	Skill 1	2
2	C	1	Skill 1	2
2	C	2	Skill 3	2

Table 3.6. Example Skill Scarcity Log

The skill scarcity log displays the time of evaluation and unavailable required skills for each eligible mode that is unable to be scheduled at the listed time of evaluation.

Resource Utilization. The resource utilization log provides usage information for each resource that is scheduled for the program. For each of those resources, the tabular data output (Table 3.7), along with a graphic that can be derived from that output (Figure 3.8), displays the percentage of overall time the schedule requires the maximum, minimum, or some other level of the resource's bandwidth to achieve the presented schedule. This output gives the project manager insights into the flexibility of the current resource pool by showing which resources have available idle time for expansion of their skill sets, but more importantly, it shows which resources must be fully exhausted in order to achieve the provided schedule.

Resource	Bandwidth	% Maxed	% Idle	% Other
R1	2	34.78	39.13	26.09
R2	1	86.96	13.04	0
R3	1	95.65	4.35	0
R4	1	73.91	26.09	0
R5	2	43.48	43.48	13.04
R6	2	43.48	4.35	52.17

Table 3.7. Example Resource Utilization Log

The resource utilization log provides usage information for each resource needed to schedule the program. It shows the percentage of the schedule's overall time that a resource's maximum level of bandwidth is needed, as well as the percentage of time the resource sits idle or is operating at some other level.

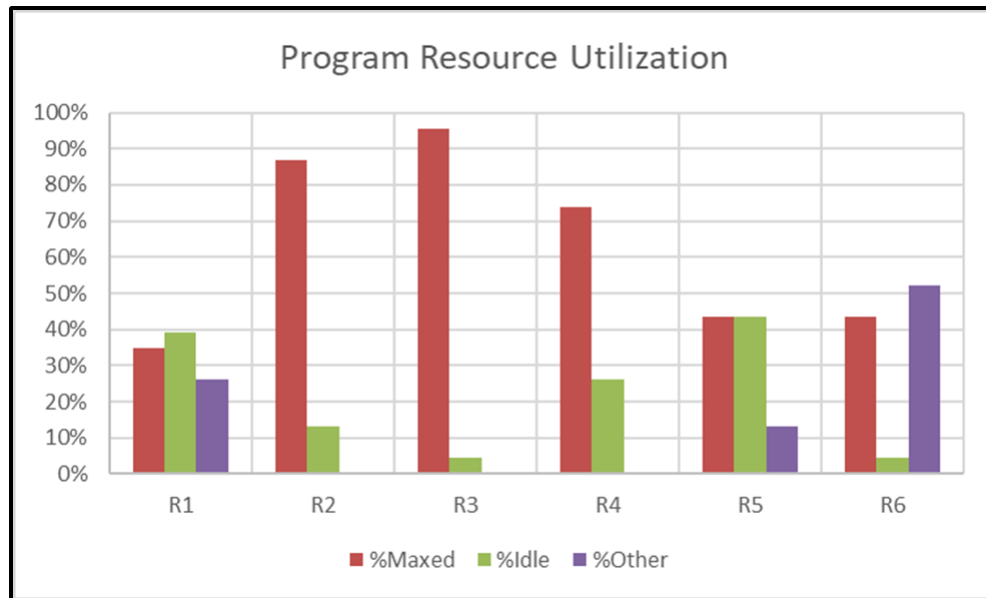


Figure 3.8. Example Resource Utilization Chart.

A graphical representation of resource utilization log data. For each resource used to schedule the program, the chart depicts the percentage of overall time that the resource is operating at its maximum bandwidth, sitting idle, or at some other level.

Skill Utilization Log. The skill utilization log provides information concerning the program's demand for each skill, as well as the percentage of the schedule's time the skill is needed and the average skill level at which the skill is performed. This data on the relative use for each skill not only allows for insights into alternative mode development, but gives a metric for the amount of risk assumed to achieve the schedule and focus points for future skill development.

Skill	Average Skill Level	% Demand	% Utilization
Skill 1	1	32	26.47
Skill 2	1.83	52	57.35
Skill 3	1	16	16.18

Table 3.8. Example Skill Utilization Log

The skill utilization log provides usage data for each skill utilized in a program. It displays the overall percentage of both the program's demand and the program's utilization of the skill, as well as the average skill level at which the skill was performed.

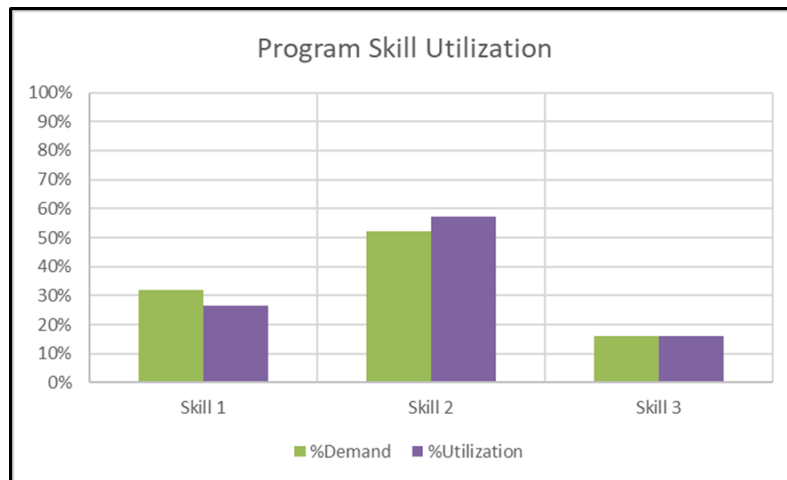


Figure 3.9. Example Skill Utilization Chart.

A graphical representation of skill utilization log data. For each skill performed in the program, the chart shows both the percentage of overall demand and the percentage of overall time a skill is needed to achieve the program's schedule..

3.5 Proof of Concept

To illustrate the insights gained by modeling and analyzing a project with DRAASTK, we return to and expand upon project 2 of our example program model from Figure 3.1. We first consider the results of a non-resource constrained CPM analysis of the project. We then compare it to a DRAASTK analysis of the project's single-mode and resource-constrained

tasks. Additionally, a pool of mono-bandwidth, single-skilled resources are associated to these tasks, with a priority for short durations and completing tasks as soon as they become available. Finally, using the results of that DRAASTK analysis, we perform adjustments to both the program modes and resource pool to demonstrate possible DRAASTK analysis-inspired changes that could improve overall program execution and give insights into the risks assumed to accomplish the provided schedules.

3.5.1 Baseline Schedule

We first consider our adapted project (Figure 3.10) with tasks that have only a single mode, and a pool of mono-bandwidth, single-skilled resources (Tables 3.9 and 3.10) that prioritize tasks with the shortest duration.

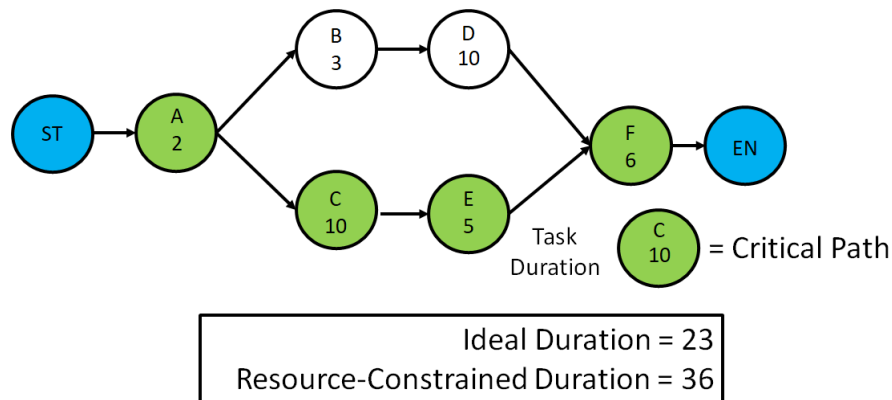


Figure 3.10. Project Baseline.

Project Baseline. By inspection, the project's critical path is A-C-E-F, with a completion time of 23 units of time. However, when available resource constraints are taken into account with DRAASTK, the calculated duration is 36 units of time.

Project	Task	Mode	Duration	Required Skills		
				Skill 1	Skill 2	Skill 3
2	A	1	2	1	1	0
2	B	1	3	2	2	2
2	C	1	10	2	0	2
2	D	1	10	0	2	1
2	E	1	5	2	2	0
2	F	1	6	1	2	1

Table 3.9. Project Modes: Baseline

Modes of the baseline case. Each task only has one mode.

Resource	Bandwidth	Skill Count	Priority Rules	Skill Level		
				Skill 1	Skill 2	Skill 3
R1	1	1	1	1		
R2	1	1	1		1	
R3	1	1	1		1	
R4	1	1	1			1
R5	1	1	1			1
R6	1	1	1	1		
Priority Rules: 1 = Shortest Duration						

Table 3.10. Resource Pool: Baseline

Resource pool for the baseline case. Each resource possesses only one skill, with a skill level and bandwidth of 1.

By inspection, the critical path of this project is through Tasks A-C-E-F, with an ideal completion time of 21 units of time. However, DRAASTK analysis of this project's resource requirements and the pool of available resources reveals a schedule (Table 3.11) that requires 36 units of time to complete. This schedule is based on the current project's organization and resource allocation, and is considered as the base line resource-constrained model to improve upon.

Project	Task	Mode	Average Skill Level	Available Time	Start Time	Finish Time	Delay Time	Slack Time	Impact
2	A	1	1	0	0	2	0	0	0
2	B	1	1	2	2	5	0	2	0
2	C	1	1	2	15	25	13	0	13
2	D	1	1	5	5	15	0	2	0
2	E	1	1	25	25	30	0	0	0
2	F	1	1	30	30	36	0	0	0

Table 3.11. Schedule Summary Log: Baseline.

The resulting schedule for the baseline requires 36 units of time to complete, where task C is delayed by 13 units of time due to resource availability.

3.5.2 Option Exploration

A review of the schedule summary log reveals that the 13 units of time difference between the project's ideal schedule and its resource constrained completion time is solely due to the 13 units of time delay impact of task C. This information allows for at least four courses of action, two of which we shall explore:

1. Provide additional resources to address the specific skill shortfalls.
2. Improve the flexibility of existing resources by adding multi-skills and/or increasing bandwidth.
3. Improve the flexibility of the system by developing additional modes.
4. Conduct any combination of the above.

Vignette 1: Provide Additional Resources

The resource conflict log for this schedule (Table 3.12) shows that task C is evaluated and delayed at two points in time: 2 and 5. At point 2, the log reveals that two units of skill 1 and two units of skill 3 are lacking, with the selected resources assigned to another task. This would require at least four additional resources with the requisite skills to address. Conversely, at time 5, only skill 3 is lacking with no available resource to provide the skill.

Project	Task	Mode	Evaluation Time	Resource	Bandwidth	Skill	Skill Level
1	C	1	2	R1	0	Skill 1	1
1	C	1	2	R6	0	Skill 1	1
1	C	1	2	R4	0	Skill 3	1
1	C	1	2	R5	0	Skill 3	1
1	C	1	5	R1	1	Skill 1	1
1	C	1	5	R6	1	Skill 1	1
1	C	1	5	R4	0	Skill 3	1
1	C	1	5	R5	1	Skill 3	1

Table 3.12. Resource Conflict Log: Baseline

The resource conflict log for the baseline case of the project. The log shows that task C is evaluated and delayed at times 2 and 5 due to a lack of resources needed to provide the listed skill.

The addition of one resource (R7), with a bandwidth and skill level for skill 3 of 1, results in a reduction of the delay time for task C to 3 units of time, with a corresponding reduction to the project's overall completion time. A graphical representation (Figure 3.11) of the resulting proposed, baseline, and ideal earliest finish times for each task illustrates the impact this additional resource has on the project.

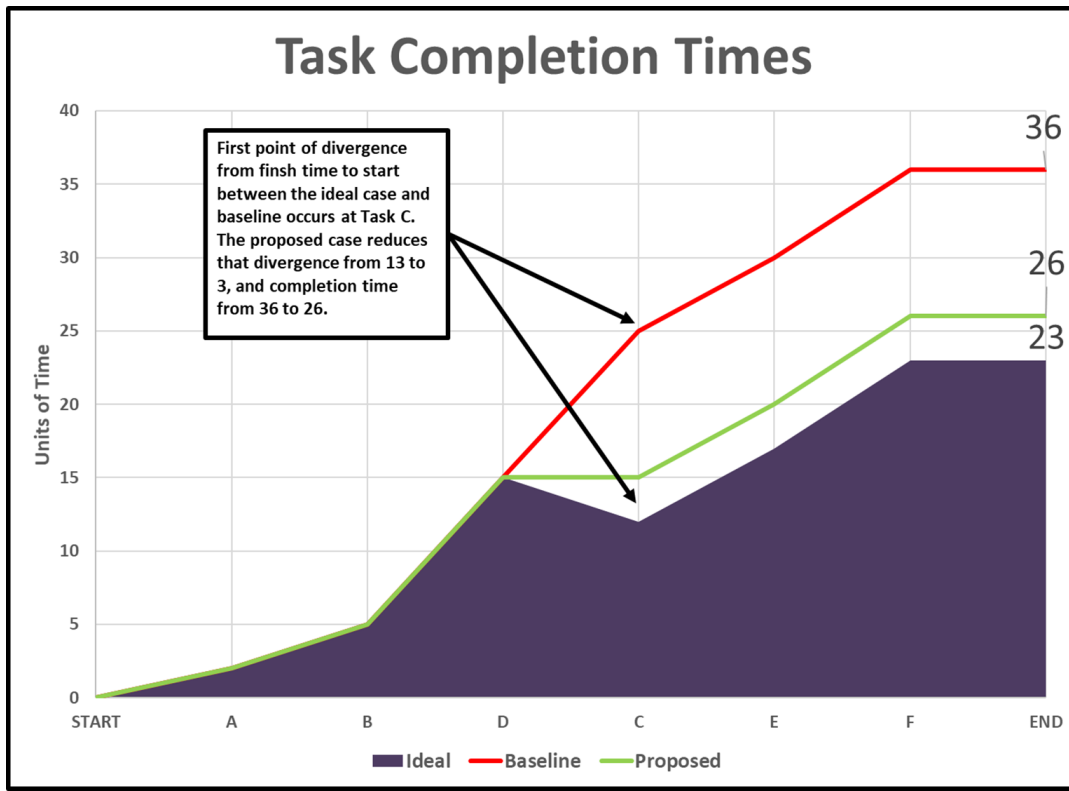


Figure 3.11. Task Finish Time Chart: Vignette 1

The earliest and calculated finish times for the ideal, baseline and proposed schedules for the project after the introduction of an additional resource.

While the addition of R7 does reduce the project's overall completion time by 10 units of time over the baseline case, a review of the graphical representation for both resource utilization logs reveals that, practically speaking, the cost of its addition may outweigh its benefits. As Figure 3.12 illustrates, R7's addition allows for this improvement in completion time, which causes a significant decrease in the idle times for most of the other resources. However, R7's own idle time is over 60 percent, as it is used only once to perform the least demanded skill. Whether or not this option is viable would depend on the project manager's assessment of completion time versus personnel cost, but this DRAASTK analysis does make the implications of this option clearer.

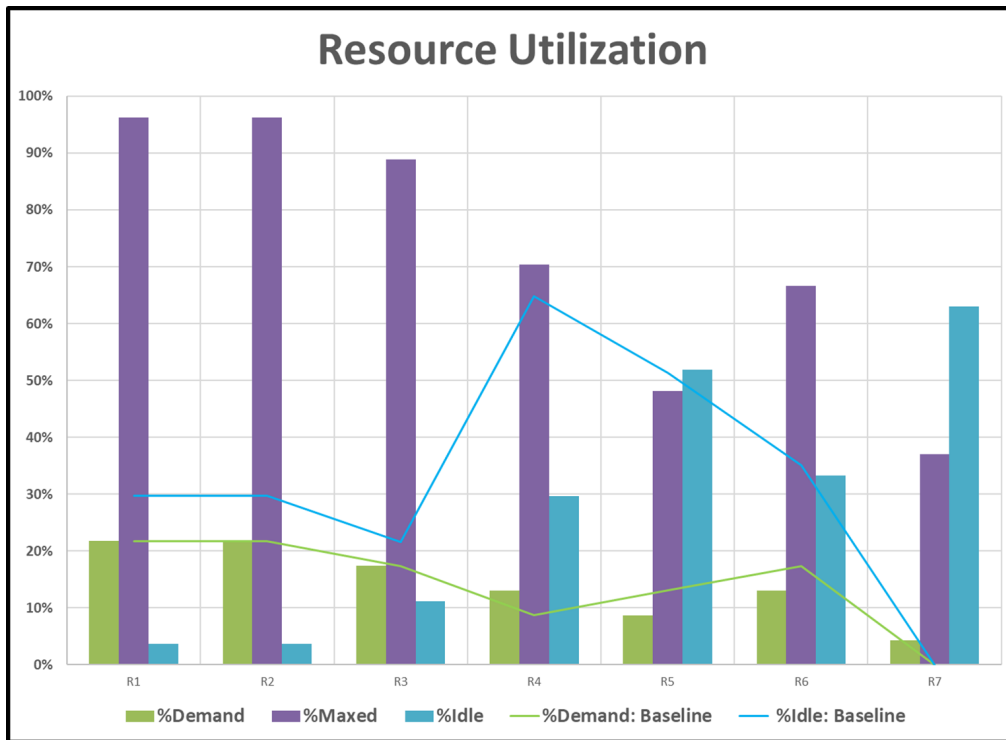


Figure 3.12. Resource Utilization Chart: Vignette 1

Chart comparing the resource utilization of the project after the addition of an additional resource, with the baseline case. The addition of R7 causes a significant decrease in the idle time of the other resources, however, the idle time of R7 itself is over 60 percent.

Vignette 2: Improve Resource and System Flexibility

In many “real world” cases, the simple addition of resources to an organization to perform limited tasks may not be a feasible solution. Resource allocations may not make it possible to hire this kind of contract work, so an organization is forced to achieve their objectives by finding ways to “do more with less.” In the context of a staff-like organization, this effort could translate into the training in additional skills for members of their existing staff, or the reconsideration of a task’s required skills to better fit the skill sets of the staff the organization has on hand.

For this vignette, the required resources to execute task C are reconsidered, and the skill sets and bandwidth of the existing resource pool are altered to expand the project’s flexibility. As

task C is the focus, it is determined that it is also able to be conducted using two resources with skill 1 and four resources with skill 2, for a duration of 8 units of time. As this mode is of a shorter duration than the original configuration for task C, it is now considered mode 1 with the previous retained as mode 2 (see Table 3.13).

Project	Task	Mode	Duration	Required Skills		
				Skill 1	Skill 2	Skill 3
2	A	1	2	1	1	0
2	B	1	3	2	2	2
2	C	1	8	2	4	0
2	C	2	10	2	0	2
2	D	1	10	0	2	1
2	E	1	5	2	2	0
2	F	1	6	1	2	1

Table 3.13. Modes: Vignette 2

The modes for vignette 2 add an additional option for task C that has a duration of 8 units of time but requires four resources with skill 2 instead of any resources with skill 3.

Changes to the flexibility of the project's resource pool is also explored in this vignette (Table 3.14). Practically speaking, an existing resource pool is limited in the amount of flexibility that can be achieved based on the type of work, policy restrictions, and inherent capabilities of the labor force. However, for illustrative purposes, we have expanded the number of multi-skilled resources and levels of bandwidth with the knowledge that skill 2 is the highest demanded skill.

Resource	Bandwidth	Skill Count	Priority Rules	Skill Level		
				Skill 1	Skill 2	Skill 3
R1	2	3	1	1	3	2
R2	1	2	1	2	1	
R3	1	1	1		1	
R4	1	2	1		4	1
R5	2	3	1	3	4	1
R6	2	2	1	1	4	
Priority Rules: 1 = Shortest Duration						

Table 3.14. Resource Pool: Vignette 2

A pool of resources with multiple skills and levels of bandwidth.

The addition of an additional mode for task C, and the changes to resource flexibility, results in a reduction in project completion time from 36 for the baseline, to 24 units of time (Figure 3.13).

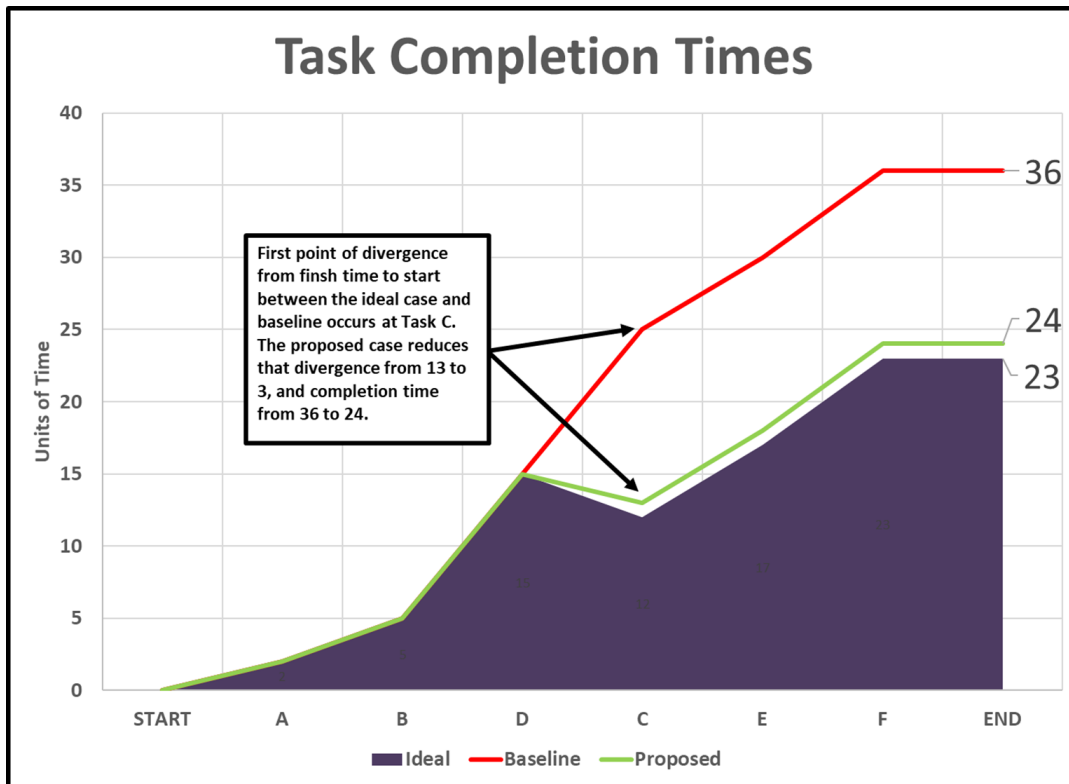


Figure 3.13. Task Finish Time Chart: Vignette 2

The earliest and calculated finish times for the ideal, baseline and proposed schedules for the project after the introduction of resource flexibility and an additional mode for task C.

While these changes in system and resource flexibility do provide an improvement to project performance, this improvement comes with the assumption of a certain amount of cost and risk. As can be seen in the resulting schedule summary (Table 3.15), in order to accomplish this improved schedule, both tasks C and D must be performed with average skill levels other than excellent.

Additionally, the change in resource utilization (Figure 3.14) from the baseline case shows little change in relative individual resource demand, but a marked decrease in resource idle time and corresponding increases to maxed and other utilization percentages. This all indicates a significant increase in individual resource utilization, where those resources with multi-level bandwidth capability must be tapped to perform simultaneous duties.

Project	Task	Mode	Average Skill Level	Available Time	Start Time	Finish Time	Delay Time	Slack Time	Impact
2	A	1	1	0	0	2	0	0	0
2	B	1	1	2	2	5	0	2	0
2	C	1	1.833	2	5	13	3	0	3
2	D	1	3	5	5	15	0	2	0
2	E	1	1	13	13	18	0	0	0
2	F	1	1	18	18	24	0	0	0

Table 3.15. Schedule Summary Log: Vignette 2.

To produce the provided schedule, both task C and task D must be performed at skill levels other than excellent: 1.83 and 3, respectively. While the utility of these values are subjective, they do help to articulate the risk an organization must assume to produce this outcome.

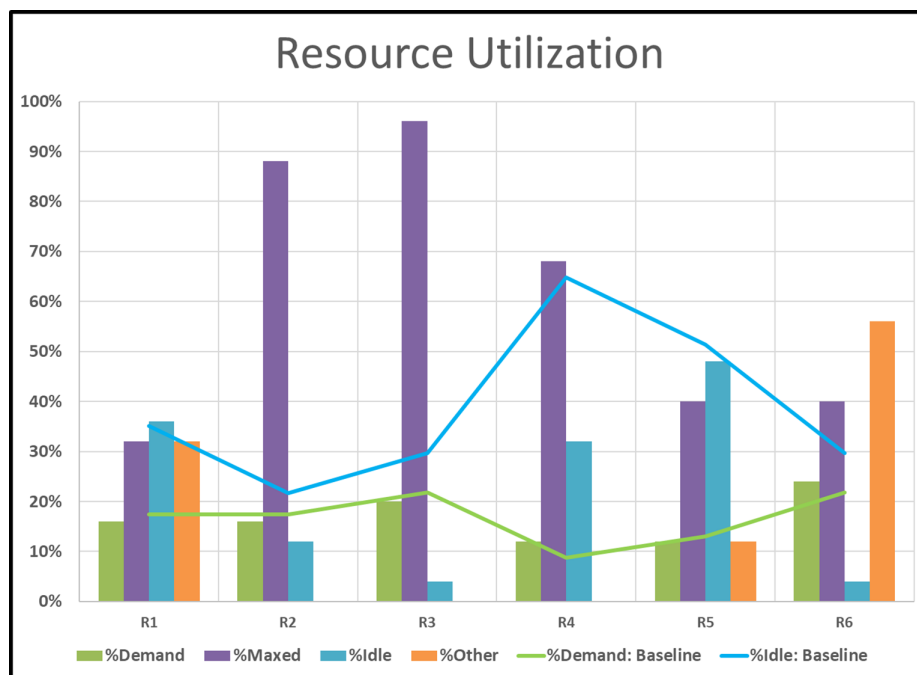


Figure 3.14. Resource Utilization Chart: Vignette 2

Chart comparing the resource utilization of the baseline case and the project after the addition of an additional mode for task C and changes to resource flexibility.

3.5.3 Analysis Conclusions

Even though the values for skill level and the effects of operating on multiple levels of bandwidth are subjective, they do help to articulate the risk an organization must accept in order to achieve the resulting schedule. Coupled to objective parameters such as level of training or time of experience, average skill level provides insights into potential quality of work performed while resource bandwidth utilization speaks to the intensity of work resources must perform in order for the system to achieve the resulting schedule. Together, the DRAASTK analysis outputs provide not only insights into system changes that could possibly improve performance, but metrics to articulate the performance of the current resource pool and the risk to work quality and sustainment assumed in achieving an outcome.

THIS PAGE INTENTIONALLY LEFT BLANK

CHAPTER 4:

Modeling MOC Staff Operations as a CMRCMPSP

In this chapter, we explain how to analyze the operations of a MOC staff as a CMRCMPSP model by converting supporting task and MOC staff positions into project and resource inputs for DRAASTK analysis. An example supporting task is used to demonstrate the procedure, comparing the results to that of previous work to highlight the advantages of DRAASTK's outputs with regard to insights into resourcing and risk.

4.1 Converting MOC Staff Operations into Model Inputs

As explained in Section 1.1.2, the MOC must meet the requirements of two core functions that are divided among a set of 11 mission essential tasks (METs). These METs are themselves composed of various subsets from among the 177 total supporting tasks, each of which consisting of a set of procedures that are accomplished by specific combinations of the 94 process driver-led functional teams (see Department of the Navy 2014, for a description of MOC mission essential and supporting tasks).

As an analogy to a CMRCMPSP, the METs that a MOC must meet can be considered as separate programs which are composed of a set of projects in the form of supporting task. These supporting tasks, as projects of a program, are composed of individual procedures, which are analogous to the tasks of a project. These procedures possess precedence relationships and expected duration, as well as key personnel requirements to lead the teams conducting work to complete the procedures. To finish the analogy, the roles of the process drivers leading the teams conducting the work to complete the procedures can be considered as the required skills for each procedure, with the MOC staff serving as the pool of resources possessing these skills and associated skill levels based on their relative levels of training and experience.

Appendix A of the MOC standardization manual (Department of the Navy 2014), provides a two-page “dashboard” graphic for each supporting task. Each graphic displays a supporting task's designation, procedures, and key manpower support required to complete each task, by procedure. In the following sections, the supporting task OP 5.3.1: *Conduct Mission*

Analysis, is used as an example to demonstrate how to convert a MOC supporting task into DRAASTK inputs.

4.1.1 Project, Task, and Mode Development

Figure 4.1 is the first page of the OP 5.3.1 dashboard from the MOC standardization manual (Department of the Navy 2014). It displays the procedures for completing the supporting task, henceforth referred to as the project. Although numbered, this numbering does not define an order or establish a precedence relationship for the process. Subject matter expert (SME) input is required to determine the precedence relationships among the procedures (tasks) and to derive values for expected durations.

OPNAV M-3500.42
6 Nov 2014

Supporting Task:
Conduct Mission Analysis (OP 5.3.1, Staff-Supporting Task)

Procedures:

1. Establish a cross-functional tailored OPT to conduct mission analysis based on HHQ order, fleet commander guidance, recommendations from Assessment, supporting command or stakeholder request, or subordinate / supporting force RFS / RFS.
2. Frame the problem. If the operational crisis is complex, develop an operational approach to design maritime operations that supports HHQ intent (objectives, conditions / effects, supporting phases, and LOOs).
3. Identify source of mission: determine ~~force, support, assets, and~~ conditions.
4. Determine support relationships.
5. Analyze higher commander's mission activities, and strategic guidance.
6. Determine own specified, implied, and
7. State purpose; determine own military initial effects / conditions.
8. Determine and analyze operational and limitations.
9. Analyze available and allocated forces and assets.
10. Determine own and enemy center of gravity (COG), critical factors,
11. Develop
12. Conduct
13. Develop effects
14. Develop
15. Develop
16. Determine
17. Publish
18. Prepare
19. Send
20. Develop mission success criteria.
21. Utilize regional and cultural understanding to synthesize and evaluate intelligence with a systems view of second/third-order effects.

Project Name:
Project name can be derived from the unique Universal Joint Task List task number of the Supporting Task

Project Tasks:
The task for project OP 5.3.1 are procedures 1-21. However, this is not an ordered list and SME input is required to determine precedence relationships and expected durations.

Mission
Service Title 10 (Man, Train, Equip) Mission
Force Management Mission
TSC / Phase 6 Mission
Offensive/Defensive Ops Contingency Missions
Non-Combat Crisis Response Contingency Missions

Function
Prepare the Force
Provide the Force
Employ the Force
Support Commander's Decision-making
Set Conditions for Subordinates Success
Monitor
Deterrence
Forward Presence
Maritime Security
Sea Control
Power Projection
Non-combat Crisis Response

Phases of Employment
0 1 2 3 4 5
Contingency

For each Procedure (TPO) – the Level of Performance
Is an objective % based on a subjective measure:
(100%) Cross-functional procedure is in place at STARTEX / commencement of operations
(75%) Cross-functional procedure is developed by FINEX / within 96 hours of ops commencement
(50%) Limited cross-functional / functional procedure is in place by FINEX / within 96 hours of ops commencement
(25%) General understanding of procedure at the functional level, but not implemented
(0%) Procedure not understood or implemented

Training Standard
Supporting Process Standard is the Training Objective
= average % of all procedures (TPE)

Joint Doctrine Refs: JP 3-33, JP 5-0

Figure 4.1. Supporting Task OP 5.3.1 Procedures. Adapted from Department of the Navy (2014).

The first page of the OP 5.3.1 supporting task dashboard. This page provides a name for a project as well as the procedures that compose the project tasks. However, this list of procedures is not an ordered list, and SME input is required to determine precedence relationships and expected durations.

While deriving task precedence relationships and skill requirements for supporting task is possible from the information provided in the MOC standardization manual and inputs from SMEs, developing multiple modes for a supporting task may not be relevant in the context of analyzing the allocation of human resources. Alternative modes for a task require that a different combination of skill sets be capable of completing that task. In an industrial context, an example of an alternative mode would be for a task to require the skills of one backhoe and one human operator for four hours, as opposed to the skills of five manual laborers with shovels for twelve. In the context of a MOC staff, however, the skills of the positions that serve as team leads are specific to the position and not, as such, interchangeable. A more realistic application of resource flexibility would be to determine which positions are similar enough in function to cross serve, then designate those resources who possess the training for those skills as having some proficiency in both skill sets, if applicable. For instance, many future operations (FOPS) and future plans (FPLANS) positions may share many capability traits, allowing those who serve in the positions in FOPS to realistically be capable of performing the duties in the corresponding FPLANS position, and vice versa.

4.1.2 Resource Pool Development

Figure 4.2 is the second page of the OP 5.3.1 dashboard. It displays the core MOC positions (process drivers) required to complete project OP 5.3.1, by procedure (task) number. Supervisor and/or SME input is required to determine the specific skill level and bandwidth of individual human resources.

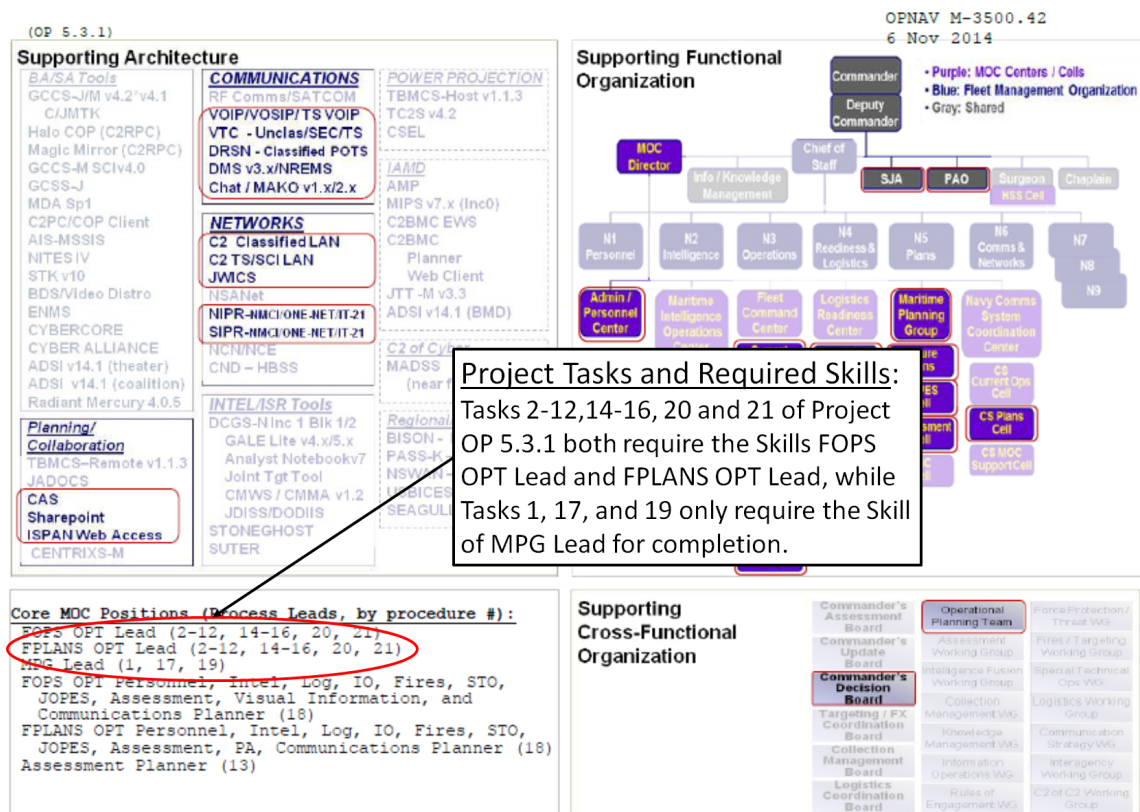


Figure 4.2. Supporting Task OP 5.3.1 Required Skills. Adapted from Department of the Navy (2014).

The second page of the OP 5.3.1 supporting task dashboard. This page provides the key personnel who lead the teams conducting work to complete procedures, by procedure number.

For a project, the positions of the process drivers are considered the skills required to complete each task. The MOC staff consists of personnel who possess the skills to serve in these positions, each with a certain level of proficiency in leading the procedure (skill level) and potentially possessing multiple skills and levels of bandwidth to lead other teams with an associated proficiency level based on training, experience, etc.

4.2 DRAASTK Analysis

Jones et al. (2018) worked with MOC staff SMEs to develop a Task-on-Node network representation of the procedures for OP 5.3.1 and values for each expected duration. For

this example, we have adopted that network and incorporated the skill requirements for each procedure as listed in the MOC standardization manual to develop a project and resource pool input for DRAASTK analysis. We compare the DRAASTK results to that of the ideal CPM results of Jones et al. (2018) to highlight the benefits that the methodology of DRASSTK provides to resource allocation analysis.

4.2.1 Inputs

Figure 4.3 is a task-on-node representation for the procedures of OP 5.3.1, with associated example task durations. The procedure numbers 1-21, directly correspond to the nodes A-Z in alphanumeric order with the exclusion of the letters; S, T and W. From this network, task precedence relationships are determined and a single mode is created for each task using the skill requirements illustrated in Figure 4.2 (see Table 4.1).

Project, Tasks, and Modes

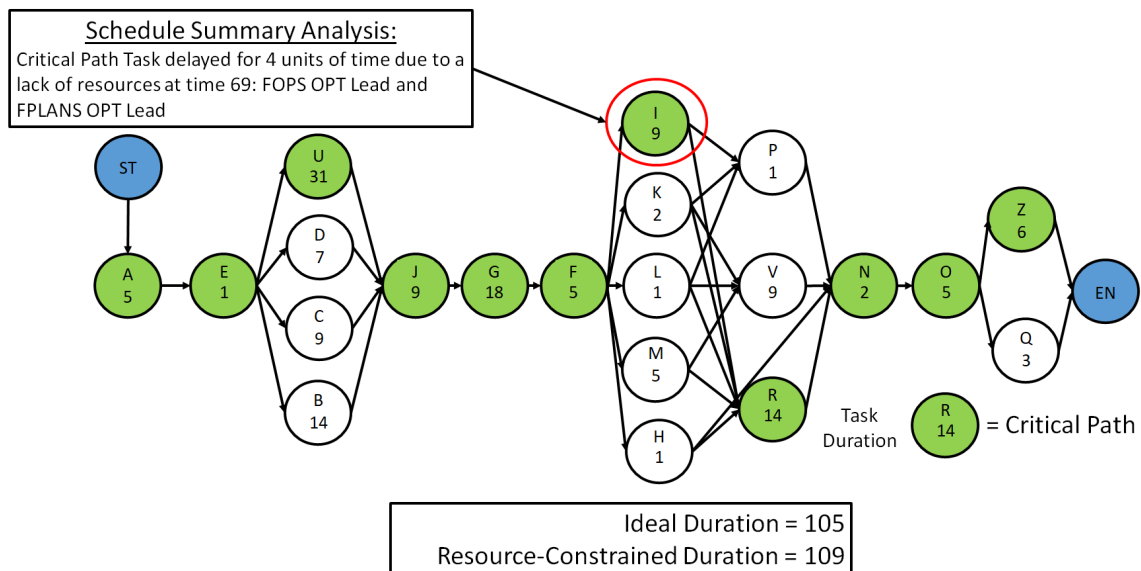


Figure 4.3. OP 5.3.1 Network Model. Adapted from Jones et al. (2018).

OP 5.3.1: *Conduct Mission Analysis* supporting task network baseline. The network has an ideal duration of 105 units of time based on its critical path, while resource constraints result in a duration of 109 after task I is delayed.

Project	Task	Mode	Duration	Required Skills							
				Assessment Planner (FOPS)	Assessment Planner (FPLANS)	Fires Planner (FOPS)	Fires Planner (FPLANS)	FOPS OPT Lead	FPLANS OPT Lead	...	MPG Lead
OP 5.1.3	A	1	5	0	0	0	0	0	0	...	1
OP 5.1.3	B	1	14	0	0	0	0	1	1	...	0
OP 5.1.3	C	1	9	0	0	0	0	1	1	...	0
OP 5.1.3	D	1	7	0	0	0	0	1	1	...	0
OP 5.1.3	E	1	1	0	0	0	0	1	1	...	0
OP 5.1.3	F	1	5	0	0	0	0	1	1	...	0
...	...	...	...	...	...	...	...	...	...	...	...
OP 5.1.3	R	1	14	1	1	1	1	0	0	...	0
OP 5.1.3	U	1	31	0	0	0	0	0	0	...	1
OP 5.1.3	V	1	9	0	0	0	0	1	1	...	0
OP 5.1.3	Z	1	6	0	0	0	0	1	1	...	0

Table 4.1. Project Modes: OP 5.3.1

An excerpt from the modes of the baseline case. Each task only has one mode, requiring only one resource, or team lead, for each required skill.

Resource Pool

The resource pool for this OP 5.3.1 example is based on the skill requirements derived from Figure 4.2 and the assumption that each process-driver position is filled by one individual who exclusively performs that skill with an excellent level of proficiency. While assuming no manning shortfalls and that each staff member is sufficiently trained and experienced is an ideal case, use of this resource pool still serves to highlight the insights gained in “real world” resource allocation and system performance when resource constraints are considered.

Resource	Bandwidth	Skill Count	Priority Rules	Skill Level							
				Assessment Planner (FOPS)	Assessment Planner (FPLANS)	Fires Planner (FOPS)	Fires Planner (FPLANS)	FOPS OPT Lead	FPLANS OPT Lead	...	MPG Lead
Assessment Planner (FOPS)	1	1	1	1						...	
Assessment Planner (FPLANS)	1	1	1		1					...	
Fires Planner (FOPS)	1	1	1			1				...	
Fires Planner (FPLANS)	1	1	1				1			...	
FOPS OPT Lead	1	1	1					1		...	
FPLANS OPT Lead	1	1	1						1	...	
...	...	...	...	...	...	...	...	...	...	...	...
MPG Lead	1	1	1							...	1
Priority Rules: 1 = Shortest Duration											

Table 4.2. Resource Pool: OP 5.3.1

An excerpt from the resource pool of the baseline case. Each resource represents one of the 94 process driver positions on the MOC staff, with a skill level of 1 for their corresponding skill, and a single level of bandwidth.

4.2.2 Results and Analysis

As illustrated in Figure 4.3, a CPM analysis of the OP 5.3.1 results in an ideal completion time of 105 units of time. However, the schedule summary of a DRAASTK analysis (Table 4.3) shows, and a finish time comparison graphic illustrates (Figure 4.4), that the resource-constrained completion time of the baseline case is delayed by 4 units of time due to the delay impact of task I.

Project	Task	Mode	Average Skill Level	Available Time	Start Time	Finish Time	Delay Time	Slack Time	Impact
OP 5.3.1	A	1	1	0	0	5	0	0	0
OP 5.3.1	B	1	1	6	22	36	16	17	0
OP 5.3.1	C	1	1	6	13	22	7	22	0
...	...	...	...	...	...	...	...	...	...
OP 5.3.1	I	1	1	69	73	82	4	0	4
...	...	...	...	...	...	...	...	...	...
OP 5.3.1	K	1	1	69	71	73	2	7	0
OP 5.3.1	L	1	1	69	70	71	1	8	0
...	...	...	...	...	...	...	...	...	...
OP 5.3.1	V	1	1	82	83	92	1	5	0
OP 5.3.1	Z	1	1	103	103	109	0	0	0

Table 4.3. Schedule Summary Log: OP 5.3.1

Excerpt from the baseline schedule for OP 5.3.1 shows a completion time of 109 units of time, with task I possessing a delay impact of 4.

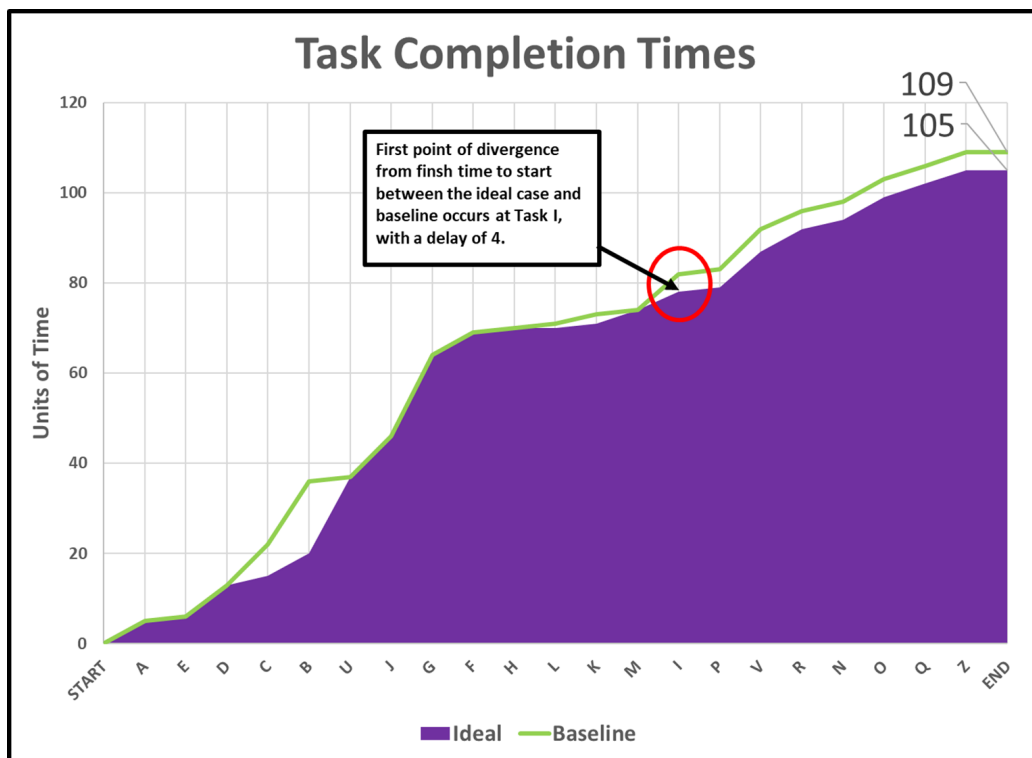


Figure 4.4. Task Finish Time Chart: OP 5.3.1

A comparison of the task completion times for the baseline case of OP 5.3.1 and the earliest finish times for the CPM analysis. With the task listed in order of completion time for the baseline case, the task closest to the project completion time where the baseline finish time first diverges from the ideal finish times of the CPM analysis, is a point of focus for addressing system delay.

A review of the resource conflict log (Table 4.4) for task I reveals that task I is evaluated at times 69, 70, and 71, before finally being executed at time 73. In each case of its delay, the lack of a FOPS OPT Lead and a FPLANS OPT Lead skill is its only hindrance, however, this is not the only insight that the DRAASTK analysis offers.

Project	Task	Mode	Evaluation Time	Resource	Bandwidth	Skill	Skill Level
...	...	...	...	...	...	...	...
OP 5.3.1	I	1	69	FOPS OPT Lead	0	FOPS OPT Lead	1
OP 5.3.1	I	1	69	FPLANS OPT Lead	0	FPLANS OPT Lead	1
OP 5.3.1	I	1	70	FOPS OPT Lead	0	FOPS OPT Lead	1
OP 5.3.1	I	1	70	FPLANS OPT Lead	0	FPLANS OPT Lead	1
OP 5.3.1	I	1	71	FOPS OPT Lead	0	FOPS OPT Lead	1
OP 5.3.1	I	1	71	FPLANS OPT Lead	0	FPLANS OPT Lead	1
...	...	...	...	...	...	...	...

Table 4.4. Resource Conflict Log: OP 5.3.1

The resource conflict log for the baseline case of OP 5.3.1. The log shows that task I is evaluated and delayed at times 69, 70 and 71 due to a lack of the FOPS and FPLANS OPT Lead skills.

While a review of the schedule summary and resource conflict log for OP 5.3.1 provide a truer measure of the impacts the employment of the current staff has on the project's performance, the resource and skill utilization logs also give insights into the areas of staff organization and training that could require focus. For instance, scarcity in the FOPS and FPLANS OPT Lead skills is responsible for all delays calculated in Table 4.3 and illustrated in Figure 4.4. The percentage of demand and utilization in the skill utilization log (Figure 4.5), confirms the relative importance of the FOPS OPT Lead and FPLANS OPT Lead that the schedule summary implies, but it also points to the potential impacts of the other skill sets and resources who possess them.

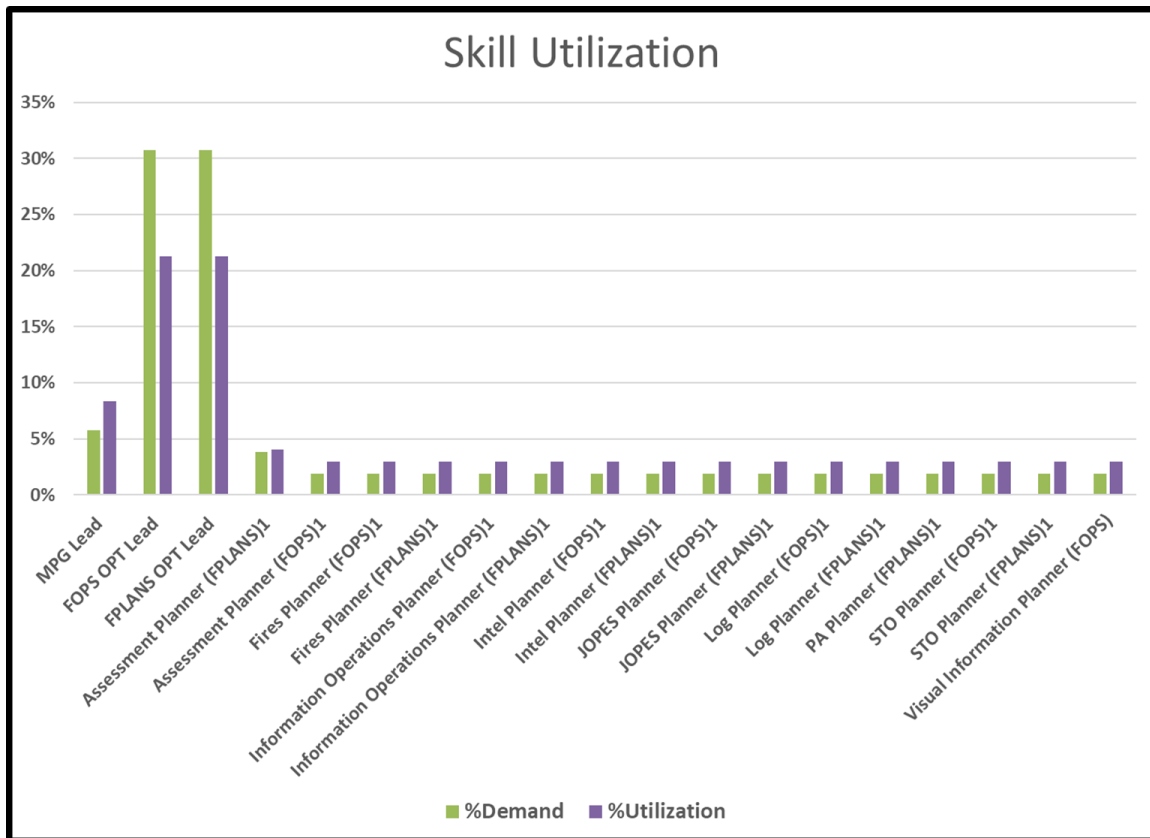


Figure 4.5. Skill Utilization: OP 5.3.1

Skill utilization for the OP 5.3.1 baseline case shows that the FOPS and FPLANS OPT leads composed the most demanded and utilized skills for the project. Since each skill in Table 4.2 is only possessed by a single resource, this skill utilization chart also reveals resource demand for the baseline case.

4.2.3 Analysis Conclusions

As the resource pool of Table 4.2 is composed of only single-skilled resources, the relatively low utilization of the remaining skill sets opens up the possibility for cross-training of other staff members in the most-demanded skills. This "dual-hatting" of staff members would decrease their percentage of idle time, while relieving the percentage of maximum utilization exerted by the currently trained staff. However, to realize any positive effects from this overlapping of personnel satisfying multiple positions, staff members would have to develop multiple levels of bandwidth and competence in skills for which they may not be able to achieve an excellent level of proficiency. The end result may be a MOC that meets its project completion metrics in the near term, but at the cost of personnel readiness

(resource utilization levels) and an assumption of a risk for degraded performance or even failure (average skill level).

THIS PAGE INTENTIONALLY LEFT BLANK

CHAPTER 5:

Conclusions and Future Work

In this chapter, we summarize our work and provide conclusions. This is followed by suggested future work.

5.1 Summary and Conclusions

In order to articulate the near- and long-term operational impacts to command and control (C2) for a staff that continues to succeed by “doing more with less” in the face of limited resources, we must find a way to quantify and objectively measure their efforts.

In this thesis, we develop a method for quantifying the operational impacts of manpower, personnel, training, and education (MPT&E) on C2 resilience in maritime operations centers (MOCs) by considering their operations as Combinatorial Multi-mode Resource Constrained Multiple Project Scheduling Problems (CMRCMPSPs).

We detail how the structured nature of the relationships between a MOC’s mission essential tasks, supporting tasks, and their corresponding procedures, allows for the intuitive conversion of its staff operations into a collection of task-on-node type network models. We also show that the defined responsibilities of core process driver positions serve as the resource requirements needed to consider staff operations as a Resource Constrained Project Scheduling Problem (RCPSP). Finally, we describe how each task can potentially have more than one configuration of resource requirements, each capable of completing the task within its own duration.

To properly model the “real world” behavior of the MOC’s diversely talented personnel, we consider task resourcing requirements for various quantities of combinatorial sets of skills that resources possess. Like the people they are representing, these resources have the ability to serve a certain number of tasks simultaneously and possess the skill for any number of process driver positions; each with a level of proficiency based on training, education, and experience.

To represent the “real world” behavior of an operational MOC, we develop a discrete

event simulation, named the Dynamic Resourcing Allocation Analysis Simulation Took Kit (DRAASTK), which simultaneously schedules multiple projects based on task precedence relationships, durations, and skill requirements. These schedules are assigned resources to satisfy their skill requirements from a common pool of resources, each of whom possess a set of user-defined priority rules to resolve conflicts arising from limited resource availability.

We compare the outputs of DRAASTK for the project schedule and conflicted resources to non-resource-constrained analysis results, and we identify the skill deficiencies that cause any divergence from the unconstrained, ideal schedule. These identified skills provide us insight into the impact of current shortfalls, and the areas where additional resources or cross-training could benefit operational performance.

DRAASTK also provides the skill and resource utilization percentages required by the resulting schedule, and the average skill levels applied to perform each task. Through this utilization and skill level information, we have a justifiable method to quantify and articulate the risk and impact that levels of manpower, personnel, training, and education have on a staff's operational performance.

5.2 Future Work

Our method for analyzing staff operations gives us insight into the extraordinary levels of effort their personnel put into ensuring their organization's success, especially in the face of limited resources. However, quantifying and measuring the operational impacts of such perseverance to C2 resilience in the "real world" is extremely difficult when the efforts of those very same people work to mitigate any empirical evidence of negative impacts in the short and near-term. Therefore, the focus of our suggestions for future work lean towards efforts to better understand the organization's structure and improvements to model utility.

Our procedure for converting MOC supporting task into task-on-node network models that can be analyzed, is dependent upon the precedence relationship translations and expected duration determinations of that supporting task's procedures. Presently, the supporting task OP 5.3.1: *Conduct Mission Analysis* (developed by Jones et al. 2018, in coordination with MOC staff SMEs), is the only available example with credible data. Like Jones et al. (2018) also suggest, the remaining supporting task should be developed in a similar manner and made available to study for insight into various scenarios.

To better model the actual efforts of a MOC staff, further effort should be placed towards developing the level of granularity for resources and the skills they possess. The MOC standardization manual (Department of the Navy 2014) currently list process drivers as the standard basic element for both resources and the skills required to complete procedures across all MOCs. However, this does not identify all of the personnel actually involved, as the process drivers are merely the leaders for groups of personnel working to perform the functions that support completing a procedure. While the positions and functions of process drivers is meant to be standard across the fleet, this configuration of supporting personnel is unique to each MOC. Any proper analysis for the impacts of MPT&E to the C2 resilience of a specific MOC should incorporate this level of staff composition fidelity into its analysis.

For a more “real world” analysis of C2 resilience, development of input granularity shouldn’t be limited to the sole consideration of personnel resources. As discussed in Section 4.1.1, the development of alternate modes for procedures may not be relevant in the context of analyzing the impacts of MPT&E. However, to gain deeper insights into the impacts of a broader range of resourcing considerations, such as network or communications suites, alternative modes should be developed that link procedures to the equipment needed by personnel to perform their duties, and the alternatives available to them. For instance, OP 5.3.1 requires use of the joint worldwide intelligence communications system (JWICS). The system allows staffs to efficiently coordinate information at the Top Secret level and complete tasks within expected time frames. However, in the event that JWICS is not available, another system that can provide the same level of high-assurance protection could be utilized, but at a different monetary and/or temporal cost whose affect can be analyzed.

The discrete event simulation we develop to process problem models with the logic of our method is adequate enough to provide useful insights into MOC operations as we describe them. However, the complexity of future simulation efforts should be addressed to allow for a more robust analysis of resourcing impacts. For instance, when transitioning from Phase 0 operations to contingency missions, a MOC staff is often augmented with additional personnel to dampen the stresses of the increased operational tempo (Department of the Navy 2014). DRAASTK does not consider resource availability schedules as an element of the resource pool, so their incorporation would allow for the evaluation of a broader range of connected supporting tasks.

Since variability in task duration does not necessarily help to highlight the merits of our method, DRAASTK is a deterministic simulation. However, future work could incorporate stochastic elements in both initial, and possibly even in resource-dependent, task duration. This would help to gain a more realistic view of the system operation, and grant insights into elements of risk management and mitigation.

As with many modeling efforts, future research geared towards gaining more empirical data of MOC operations is most beneficial. A wider field of data, combined with a more complex simulation solution that follows the logic of our method and incorporates many of these future research elements, would help develop a richer bank of objective data for quantifying effort and risk. This data is needed to articulate the risks assumed by, and justify the resourcing request of, staffs that continue to succeed in the face of limited resources while “doing more with less.”

APPENDIX: Terms

We define the following terms for use throughout our work.

Available Time. The system time when either a resource becomes available for utilization by a task or a task becomes available for scheduling.

Bandwidth. The number of skills a resource can perform simultaneously. For example, a resource may have the skills of project leader, project analyst, and communications coordinator with a bandwidth of 2. Therefore, that resource has the ability to perform any two of those skills at a given time, but would become unavailable for utilization until one of those two tasks is completed.

Duration. The time required for a mode of a task to complete once it is scheduled to start.

Finish Time. The system time where a task completes its duration and any levels of bandwidth utilized by the task are released to the associated resource.

Mode. Alternate combinations of skills for accomplishing the same task.

Name. A unique identifier for either a task or a resource.

Predecessors. The set of tasks that must be completed before a task is able to be scheduled.

Processed. Condition where a task has been scheduled and the finish time has passed the system time.

Program. Collection of all projects considered.

Project. A collection of tasks which possess precedence relationships, duration, and resource requirements to accomplish work.

Required Skills. A set of skills a mode of a task requires to complete work.

Resource. An element utilized by a task for a given duration, to perform a skill and complete work.

Skill. The type of work required by a task and performed by a resource.

Skill Count. The number of different skills a resource can perform.

Skill Level. The level of competence a resource possesses for a skill. It is a subjective value with a relative worth that diminishes as the value increases. For example, a skill level of 1 would be a skill that a resource is deemed to be excellent at performing, while a value of 2 would be less than excellent, 3 less than 2, and so on. However, a skill level of 0 or none indicates that the resource does not possess that skill.

Start Time. The system time where a mode of a task utilizes a level of bandwidth of each resource it uses to perform a skill it requires for completion.

Successors. The set of tasks that cannot be started until completion of the given task.

Task. The lowest level of work element, which cannot be divided further and influences attributes of the project such as cost, duration, or resource utilization. Projects consist of a set of tasks where synonymous terms are activities, jobs, or operations. While activity is the most common term used for this element of a project in project scheduling problems, the OPNAV M-3500.42: *Maritime Operations Center Standardization Manual* utilizes the task terminology.

List of References

- Davidson P (2011) *Robust analysis of the joint strike fighter integrated training center pilot scheduling*. Master's thesis, Naval Postgraduate School, Monterey, CA, <http://hdl.handle.net/10945/5610>.
- Department of the Navy (2008) Maritime operations at the operational level of war. NWP 3-32, Washington, DC, <https://portal.nwdc.navy.mil/ndls>.
- Department of the Navy (2013) Navy tactics, techniques, and procedures: Maritime operations center. NTTP 3-32.1, Washington, DC, <https://portal.nwdc.navy.mil/ndls>.
- Department of the Navy (2014) *Maritime operations center standardization manual*. OPNAV M-3500.42, Washington, DC, <https://www.secnav.navy.mil/doni/SECNAV%20Manuals1/3500.42.PDF>.
- Gen M, Cheng R, Lin L (2008) *Network Models and Optimization: Multiobjective Genetic Algorithm Approach (Decision Engineering)* (Springer-Verlag London Limited, London).
- Joint Chiefs of Staff (2018) Joint operations. JP 3-0, Washington, DC, https://www.jcs.mil/Portals/36/Documents/Doctrine/pubs/jp3_0ch1.pdf?ver=2018-11-27-160457-910.
- Jones C, Hughes R, Oliver M (2018) *Maritime operations center resourcing decision aid*. Master's thesis, Naval Postgraduate School, Monterey, CA, Calhoun dissertations and theses database.
- Kolisch R (1995) *Project Scheduling Under Resource Constraints: Efficient Heuristics for Several Problem Classes* (Physica-Verlag, Heidelberg).
- Meade W, Zaricor S, Morgan L, Fennessey D, Peot M, McGowan L (2016) MOC certification report for U.S. Seventh Fleet. Technical Report 3500.42-20 DEC 2016, US Fleet Forces Command, Norfolk, VA.
- Pinha D (2015) *Short-Term Resource Allocation and Management*. Doctoral dissertation, Department of Industrial and Management Systems Engineering, Statler College of Engineering and Mineral Resources at West Virginia University, Morgantown, WV.
- Pinha D, Ahluwalia R, Senna P (2016) The combinatorial multi-mode resource constrained multi-project scheduling problem. *International Journal of Supply and Operations Management* 3(3):1391–1412, <http://dx.doi.org/10.22034/2016.3.03>.

- Pritsker A, Watters L, Wolfe P (1969) Multiproject scheduling with limited resources: A zero-one programming approach. *Management Science* 16(1):93–108, <https://www.jstor.org/stable/2628369>.
- Snyder C (2013) *A User's Manual to the PMBOK® Guide*, 5th ed. (John Wiley & Sons, Inc., Hoboken, New Jersey).
- Vanhoucke M (2012) *Project Management with Dynamic Scheduling: Baseline Scheduling, Risk Analysis and Project Control* (Springer-Verlag, Berlin Heidelberg).
- Wauters T, Kinable J, Smet P, Vancroonenburg W, Vanden Berghe G, Verstichel J (2016) The multi-mode resource-constrained multi-project scheduling problem: The mista 2013 challenge. *Journal of Scheduling* 19(3):271–283, <https://doi.org/10.1007/s10951-014-0402-0>.
- Williams R, Scott D, Sarchet D, Morgan L, McGowan L (2016) MOC certification report for U.S. Pacific Fleet. Technical Report 3500.42-27 DEC 2016, US Fleet Forces Command, Norfolk, VA.

Initial Distribution List

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