Report Documentation Page

Form Approved
OMB No. 0704-0188

1. REPORT DATE	2. REPORT TYPE	3. DATES COVERED
2009	**final**	**00-00-2009 to 00-00-2009**

4. TITLE AND SUBTITLE
Assessing capabilities and risks in Air Force programming framework, metrics, and methods

5a. CONTRACT NUMBER
5b. GRANT NUMBER
5c. PROGRAM ELEMENT NUMBER

6. AUTHOR(S)
Don Snyder; Patrick Mills; Adam Resnick; Brent Fulton

5d. PROJECT NUMBER
5e. TASK NUMBER
5f. WORK UNIT NUMBER

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
RAND Corporation,1776 Main Street,Santa Monica,CA,90407

8. PERFORMING ORGANIZATION REPORT NUMBER
MG-815-AF

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)

10. SPONSOR/MONITOR’S ACRONYM(S)

11. SPONSOR/MONITOR’S REPORT NUMBER(S)

12. DISTRIBUTION/AVAILABILITY STATEMENT
Approved for public release; distribution unlimited

13. SUPPLEMENTARY NOTES
Online access http://www.rand.org/pubs/monographs/MG815/

14. ABSTRACT
The goal of the defense budget is to deliver a portfolio of capabilities to meet a spectrum of uncertain future security environments. Despite progress in recent U.S. Air Force capabilities-based programming efforts, many limitations persist, and there are many disconnects between capability assessments and programming. To help mitigate these limitations, the findings presented here reexamine capabilities-based programming by introducing a new definition of capability metrics and a new set of algorithms for building and evaluating budgeting allocations. The capability metrics relate resources to operational-level activities in national plans, apply across a range of programs, and conform to programming boundaries. The algorithms provide the programmer with a means to quantitatively and reproducibly develop programming options in light of an uncertain future security environment, serving as a means to express capabilities and risks of resource allocations in terms of national planning objectives.

15. SUBJECT TERMS

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT	b. ABSTRACT	c. THIS PAGE			
unclassified	**unclassified**	**unclassified**	**Same as Report (SAR)**	**88**	

This product is part of the RAND Corporation monograph series. RAND monographs present major research findings that address the challenges facing the public and private sectors. All RAND monographs undergo rigorous peer review to ensure high standards for research quality and objectivity.

Assessing Capabilities and Risks in Air Force Programming

Framework, Metrics, and Methods

Don Snyder, Patrick Mills, Adam C. Resnick, Brent D. Fulton

Prepared for the United States Air Force

Approved for public release; distribution unlimited

RAND PROJECT AIR FORCE

The research described in this report was sponsored by the United States Air Force under Contract FA7014-06-C-0001. Further information may be obtained from the Strategic Planning Division, Directorate of Plans, Hq USAF.

Library of Congress Cataloging-in-Publication Data

Assessing capabilities and risks in Air Force programming : framework, metrics, and methods / Don Snyder ... [et al.].
p. cm.
Includes bibliographical references.
ISBN 978-0-8330-4608-6 (pbk. : alk. paper)
1. United States. Air Force—Appropriations and expenditures. 2. United States. Air Force—Procurement 3. United States. Air Force—Planning. 4. Program budgeting—United States. 5. United States. Air Force—Operational readiness. 6. Risk assessment—United States. I. Snyder, Don, 1962-

UG633.2.A85 2009
358.4'03—dc22

2009011322

Cover photo courtesy of DoD/Master Sgt. Ken Hammond, U.S. Air Force.

Published 2009 by the RAND Corporation
1776 Main Street, P.O. Box 2138, Santa Monica, CA 90407-2138
1200 South Hayes Street, Arlington, VA 22202-5050
4570 Fifth Avenue, Suite 600, Pittsburgh, PA 15213-2665
RAND URL: http://www.rand.org
To order RAND documents or to obtain additional information, contact
Distribution Services: Telephone: (310) 451-7002;
Fax: (310) 451-6915; Email: order@rand.org

Preface

To maximize its capabilities, the U.S. Air Force seeks to allocate its appropriated funds in the most efficient and effective ways possible to garner the most capability possible. The challenge in recent years has been to define and quantify capabilities in ways that are useful and informative to programmers. The RAND Corporation was asked by the U.S. Air Force Office of the Deputy Chief of Staff for Logistics, Installations, and Mission Support (AF/A4/7) to develop a methodology to address capabilities-based programming decisions within the purview of AF/A4/7. It was requested that this methodology be as widely applicable as possible. This monograph presents the resulting methodology for capabilities-based programming; a forthcoming companion report will use this methodology to examine one program in detail, the Basic Expeditionary Airfield Resources sets.

The research reported here was initiated in fiscal years 2005 and 2006 as part of the project "Balancing Combat Support Resources" and concluded in fiscal year 2007 as part of the project "Achieving Enhanced Operational Effects with Tailored Combat Support Packages." The research was sponsored by AF/A4/7 and conducted within the Resource Management Program of RAND Project AIR FORCE. The work is intended to help programmers understand how to incorporate capability assessments into programming decisions and the basic steps needed to implement the envisioned capabilities-based programming. This research should be of interest to programmers, analysts, capability and risk assessors, and planners.

RAND Project AIR FORCE

RAND Project AIR FORCE (PAF), a division of the RAND Corporation, is the U.S. Air Force's federally funded research and development center for studies and analyses. PAF provides the Air Force with independent analyses of policy alternatives affecting the development, employment, combat readiness, and support of current and future aerospace forces. Research is conducted in four programs: Force Modernization and Employment; Manpower, Personnel, and Training; Resource Management; and Strategy and Doctrine.

Additional information about PAF is available on our Web site: http://www.rand.org/paf/

Contents

Figures

Tables

Summary

The current overarching goal of the defense budget is to deliver a portfolio of capabilities to meet a spectrum of uncertain future security environments. Over the past several years, the U.S. Air Force has made progress in creating a process for evaluating capabilities and integrating this analysis into programming.

Despite this progress, many limitations persist, and there are many disconnects between capability assessments and programming. One deficiency is that capability assessments remain anchored in subjective, nonreproducible judgments. A second weakness is that there is a disconnect between defined capabilities and the resources to be allocated: dollars and manpower. A programmer faces great difficulties in terms of how to adjust programming following an evaluation of excess or insufficient capabilities, particularly if the relationship between those capabilities and available resources remains obscure. A third weakness is that capability assessments are currently performed against a single plausible future, not a spectrum of possible security environments. The uncertainty of the future—one of the central themes of capabilities-based planning—is therefore not captured by current assessments of capabilities and risks. (See pp. 5–13.)

In this monograph, we present a methodology that redresses these limitations by reexamining how capabilities-based programming is viewed and performed. First, we introduce a new definition of *capabilities* and present capability measures developed specifically to inform programming decisions. (See pp. 15–24.)

The goals are that the new capability metrics

- relate directly to national planning objectives
- relate to program elements, definable parts of program elements, or groups of program elements
- apply broadly across a range of programs.

We define *capabilities* as the set of resources needed to perform an operational-level activity specified in the Defense Planning Scenarios. For example, the set of resources needed to perform a specified major combat operation (MCO)—call it MCO-1—would constitute a one MCO-1 capability. For example, if 17 fire trucks of a particular type are deemed necessary for the MCO-1 contingency, then 17 of those trucks constitute a one MCO-1 capability. Similar metrics can be defined for a number of contingencies, including MCOs, small-scale contingencies, humanitarian relief operations, and steady-state deployments, such as drug interdiction and noncombatant evacuation operations, that might not rise to the level of supplemental funding. In this definition, the capability of a resource is not fixed. It has a value only relative to an operational scenario. Twenty refueling trucks may constitute 0.8 of a particular MCO but 2.3 of a particular small-scale contingency. This definition of *capabilities* naturally ties capabilities to national plans and to operational objectives. (See pp. 15–34.)

The second step is to quantify the resources needed for each deployment in the planning scenarios. Previous RAND work developed a prototype tool that ascertains the resources needed for a deployment based on how many and what types of aircraft are deployed to each base, the sortie rates they fly, and some general characteristics of the infrastructure at each base (Snyder and Mills, 2004, 2006). These characteristics include how much billeting is available, whether there is a fuels hydrant system available, and if the base is exposed to a high, medium, or low risk of conventional or nonconventional attack. This tool is adequate for determining deployment requirements for programming, and it is also useful during execution. However, the tool needs to be formally vetted, implemented, and periodically maintained by the Air Force in order to be used regularly in programming. (See pp. 21–34.)

Third, we develop algorithms that allocate funds optimally across resources for both procurement and sustainment. These algorithms can either examine programming relative to a single-scenario set or develop a program that is robust across a range of scenario sets. The robust optimization maximizes a capability relative to a number of scenario sets, subject to budgetary constraints. This monograph also develops two optimizations for planning using a single-scenario set. All optimizations recommend how to allocate spending between procurement and sustainment. The first determines the minimum cost for meeting all requirements specified in a set of planning scenarios subject to the constraint that spending not fluctuate more than a certain percentage from year to year. The second maximizes the capability relative to a single-scenario set, given a fixed budget specified for each year. (See pp. 35–53.)

These optimizations provide the programmer with analytically based, reproducible insights into how to build a robust program and how effective that program would be against an uncertain future. The algorithms express assessments of capabilities and risks.[1]

Therefore, we recommend that

- when feasible, capabilities be defined in terms of national-level plans rather than Air Force tasks
- a rules-based tool be developed and maintained for generating deployment requirements, given air order of battle–level inputs for planning scenarios
- analytical, reproducible algorithms be developed to assist in the building of a robust program across a range of plausible scenario sets that balance asset levels with sustainment investments, in lieu of programming to meet a single challenging scenario set. (See pp. 65–67.)

---

[1] We use the term *risk* to mean the expected, unrealized capability to perform operational activities in the Defense Planning Scenarios.

Following these recommendations would provide a reproducible, analytical foundation for program development and evaluation. The program would link clearly to planning objectives, and the implications of the program would be expressed in terms of national-level operational objectives rather than Air Force tasks. The methodology would not only encompass and evaluate the effectiveness of a program against a single plausible future, it would also be robust against a range of possible future security environments.

Acknowledgments

We dedicate this monograph to the memory of C. Robert Roll, Jr., who inspired and supported us throughout our research effort.

This work was sponsored by the U.S. Air Force Deputy Chiefs of Staff for Installations and Logistics, Lt Gen Michael Zettler and Lt Gen Donald Wetekam, and by the Directors of Resource Integration Maj Gen Art Morrill, Maj Gen Ronald Ladnier, and Maj Gen Polly Peyer. Without their continuing support, this work would not have been possible. (Ranks and positions are as of the time of our research.)

Feedback from interim briefings of this work helped shape and focus the methodology. We are especially grateful for the feedback from Lt Gen Howie Chandler, Lt Gen Stephen Wood, Maj Gen Gary McCoy, Maj Gen Craig Rasmussen, Maj Gen Mike Worden, Brig Gen Art Cameron, Jacqueline Henningsen, Michael Aimone, Grover Dunn, Kathe Graham, Chris Puckett, William Budden, Col Patricia Battles, Col Amy Bouchard, Col Frank Gorman, Col Connie Morrow, Col Jose Rodriguez, and Dave Sweet.

Lt Col Heather Buono, Lt Col Larry Audet, Lt Col Marty Wiseman, Lt Col Robert Helgeson, and Lt Col Warren Berry served as dedicated, seemingly tireless action officers. They all helped more than they probably realize.

At RAND, we benefited from numerous discussions and feedback; in particular, we thank Mahyar Amouzegar, Gary Massey, Ronald McGarvey, and Robert Tripp. Formal reviews by Michael Kennedy and Keenan Yoho significantly strengthened this monograph.

The authors alone are responsible for any errors or oversights that may remain.

Abbreviations

AF/A4/7	Office of the Deputy Chief of Staff for Logistics, Installations, and Mission Support
AF/A5X	Office of the Director for Operational Plans and Joint Matters
BEAR	Basic Expeditionary Airfield Resources
CONOPS	concepts of operations
CRRA	Capabilities Review and Risk Assessment
DoD	U.S. Department of Defense
FYDP	Future Years Defense Program
GAMS	General Algebraic Modeling System
IPL	integrated priority list
J8	Joint Staff Directorate for Force Structure, Resources, and Assessment
LP	linear programming
MAJCOM	major command
MCL	Master Capabilities Library
MCO	major combat operation
O&M	operations and maintenance

OAF	Operation Allied Force
OEF	Operation Enduring Freedom
OIF	Operation Iraqi Freedom
OSD	Office of the Secretary of Defense
OSD/PA&E	Office of the Secretary of Defense for Program Analysis and Evaluation
PE	program element
POM	Program Objective Memorandum
PPBE	Planning, Programming, Budgeting, and Execution
PPBS	Planning, Programming, and Budgeting System
PSM	Process Sequence Model
START	Strategic Tool for the Assessment of Required Transportation
TPFDD	time-phased force deployment data
UTC	unit type code
VBA	Microsoft® Visual Basic® for Applications

CHAPTER ONE

Introduction

Perhaps no act each year defines the U.S. Air Force's policy goals more than its decision about what to fund in its Program Objective Memorandum (POM). The Air Force POM is the set of programs and budget appropriations requests that the service submits to the Office of the Secretary of Defense (OSD) and that becomes, with some modifications, the Air Force portion of the nation's defense budget. Which programs make it into the defense budget and how their priorities have been determined have evolved over the 60-some-year history of the department.[1]

During the tenures of presidents Truman and Eisenhower, the White House established an overall target top-line ceiling for defense spending. This ceiling was determined by two factors: a desire to maintain a balanced budget and a sense that the defense budget should be a fixed portion of the gross domestic product. The defense budget was constrained more by the overall economy than by what emerged from a national assessment of external security threats or desired capabilities. Using this ceiling, OSD then allocated proportions of the budget for each service. These proportions did not change much from year to year—the Air Force share was generally around 47 percent. Each service determined its programming independently within this top line.

A consequence of this approach was that programming and budgetary priorities were not driven by strategic plans, and defense efforts lacked interservice coordination. In the 1950s, for example, the Army,

[1] The following discussion of the early years of defense programming is derived largely from Korb, 1977; Kanter, 1979; and Stevenson, 2006.

the Navy, and the Air Force all pursued duplicative programs to develop intercontinental ballistic missiles. Furthermore, there was no logical process to prioritize programs, either within or among the services.

When Robert McNamara assumed the duties of secretary of defense in 1961, he decided to remove many of the programming decisions from the services and centralize them in OSD. He implemented these changes by instituting a radical reformation of the defense spending process—the Planning, Programming, and Budgeting System (PPBS). The planning portion established the goals, objectives, and force levels; the programming portion defined which programs would carry out these plans; and budgeting estimates were made by each service to determine the overall costs of executing the programs. Planning and programming decisions rested in OSD, not the services. Individual service budgets were replaced by budgets to carry out the overall programs in 10 mission categories. The formal input into programming from the services consisted of change requests, which were adjudicated within OSD.

During the Nixon administration, the secretary of defense, Melvin Laird, decentralized aspects of the process, but not so much as to return to the extent of decentralization in the 1940s and 1950s. Laird retained decisions on plans and objectives within OSD but granted the services responsibility for building the programs to meet those plans and objectives. OSD limited itself to reviewing those programs and making changes. Each service's submission took the form of the newly created POM.

From this juncture until the end of the Clinton administration, planning objectives were derived from operational (war) plans. After the creation of the regional commands with the Goldwater-Nichols Act of 1986 (see Lederman, 1999), these plans were maintained by the unified regional commands. Operational plans detailed how the combined services might respond in specific geographic regions to specific potential adversaries. The service POMs were built to organize, train, and equip the forces to meet these combatant commander plans. As the geopolitical environment changed in the last two decades of the 20th century, these plans were updated to reflect the most probable engagements.

When Donald Rumsfeld became secretary of defense in 2001, he modified the PPBS process to better prepare for a less certain future threat environment. The process was renamed the Planning, Programming, Budgeting, and Execution (PPBE) system to reflect that the execution component is on par with the others. We discuss this new system in more detail in Chapter Two, but the key change introduced was to abandon programming intended to meet the needs determined by operational plans maintained by combatant commanders in favor of programming to develop a portfolio of capabilities able to meet an uncertain future security environment (DoD, 2001). The logic was that, more so than during the Cold War, the location and identity of U.S. adversaries were uncertain, and, thus, robust programming that could meet a range of potential adversaries was a more secure posture than deterministic programming around a limited set of specific threats.

Over time, then, the emphasis in how the defense budget is constructed has shifted considerably. It began after the Second World War with allocating money to the services according to fiscal constraints, then leaving each service the freedom to program as it saw fit within strategic guidelines. During the past several decades, planning was more centralized, with the services programming to meet deterministic operational plans. These plans were designed around potential engagements with specific adversaries in specific geographic regions. The current budgeting process reflects less certainty about the nature of threats, and hence strives for robust programming in the form of a portfolio of capabilities to meet an uncertain set of adversaries in any region. This strategy should better position the United States to meet uncertain future threats. But how can the Air Force build a robust POM around a portfolio of capabilities that meets these goals? How can a programmer[2] match capabilities with resource requirements? These are the current programming challenges.

In this monograph, we discuss general approaches to capabilities-based programming in the Air Force and, specifically, develop a meth-

---

[2] When we refer to programmers throughout this monograph, we mean all those involved in the building of the Air Force POM, at both the major commands (MAJCOMs) and the Air Staff.

odology for capabilities-based programming for agile combat support resources. A future companion report will present a proposed budget and a capabilities and risk analysis for the Basic Expeditionary Airfield Resources (BEAR) program.

CHAPTER TWO

Air Force Programming and Capability Assessments

The Air Force has instituted processes for developing its programming around a set of capabilities.[1] In this chapter, we review the current process for developing the budget and the process for assessing capabilities in the Air Force. We follow these discussions with a critical examination of how these two processes interact.

Current Air Force Planning and Programming

Each year, the Air Force establishes priorities and sets budgets for scores of programs that constitute its roughly $111 billion portion of the presidential budget submission to Congress.[2] The size and complexity of the Air Force gives rise to a comparably complex budgeting process that goes on continuously and engages numerous staff, from the MAJCOMs to the Air Staff. Decisions regarding what to include and how to balance programs within the budget determine the capabili-

---

1 The Air Force defines a *capability* as the "combined capacity of personnel, materiel, equipment, and information in measured quantities, under specified conditions, that, acting together in a prescribed set of activities can be used to achieve a desired output" (Air Force Instruction 10-604, 2006, p. 3).

2 This figure refers to the "blue" portion of the fiscal year 2008 presidential budget and excludes that portion of the Air Force budget not under the control of the Air Force (i.e., the National Foreign Intelligence Program, Special Operations Command, and Defense Health Program).

ties that the Air Force garners and the risks[3] it assumes for national defense.

The current system for creating the U.S. Department of Defense (DoD) contribution to the presidential budget, in which the Air Force participates, is the PPBE process. This system divides the budget-building process into four phases:

- planning, which provides guidance for devising strategies to meet the nation's defense needs, expressed as military objectives
- programming, which translates the planning objectives into specific packages of resources allocated to specific agencies, called *programs*
- budgeting, which assigns the best estimates of costs to these programs
- execution, in which obligated money is spent to carry out the programs.

The last three phases are largely the responsibility of the services, in this case, the Air Force.

Various organizations specify and report military planning goals on a regular basis, including the White House (National Security Strategy), OSD and the Joint Chiefs of Staff (National Military Strategy, Quadrennial Defense Review, Guidance for the Development of the Force, Guidance for the Employment of the Force, and Joint Strategic Capabilities Plan), and the Office of the Secretary of the Air Force (Annual Planning and Programming Guidance). Collectively, these documents describe a planning environment fundamentally changed from that of even a few years ago. Planning objectives in the recent past revolved around operational plans drawn up to address threats from specific adversaries in specific locations. Recognizing that planning must reflect current uncertainties in the security environment, objectives now focus on maintaining a portfolio of capabilities.

---

[3] We use the term *risk* to refer to the expected, unrealized capability to perform operational activities in the Defense Planning Scenarios.

This is not to say that evaluation of specific threats has been removed from the planning process—a spectrum of threats and contingencies still determines the nature and balance of required capabilities. It is the emphasis that has shifted, from an optimal set of capabilities to a robust set. Planning for optimal capabilities focuses on specific threats; planning for a robust set of capabilities is focused on effectiveness against a range of conflicts. This change in planning perspective has direct consequences for programming.

Under the current PPBE process, the Air Staff is responsible for building the Air Force POM with assistance from the MAJCOMs. Subject to fiscal guidance, the Air Staff develops a set of program elements and a level of funding for those program elements to enable the Air Force to organize, train, and equip the forces to meet overall planning goals. Air Force guidance comes largely from the Annual Planning and Programming Guidance document, and the requests from the combatant commanders come in the form of integrated priority lists (IPLs). The organization within the Air Staff that oversees the building of the POM is the Air Force Corporate Structure.

The Air Force Corporate Structure is organized into four tiers. The lowest and the first step in the process of moving the POM through the corporate structure is carried out in the Air Force Panels. These are mission- and mission support–specific panels that balance programming needs at the mission level. Currently, the Air Force top-line ceiling is divided among the panels, and each panel attempts to optimally balance its resources across its programs. This structure is in contrast to the process of the recent past, in which the MAJCOMs were given a slice of the ceiling to balance across their missions.

The next step is the Air Force Group (chaired by the Deputy for the Directorate of Programs under the Deputy Chief of Staff for Strategic Plans and Programs), which conducts the first Air Force–wide review of the budget. The Air Force Board (chaired by the Director of Programs under the Deputy Chief of Staff for Strategic Plans and Programs and the Deputy Assistant Secretary of the Air Force for Budget) provides a senior-leader perspective, and the Air Force Council (chaired by the Air Force Vice Chief of Staff) finalizes the Air Force programming.

After the corporate structure has finalized the programming, a further refinement of costs is assigned in the budgeting process, which may entail some minor changes to the programming. The final POM and justifications for the POM for a given fiscal year are then submitted to DoD about a year before the fiscal year begins. DoD may adjust or contest aspects of the programs. The Air Force can argue its case for the programming via a reclama. In the first week of February, DoD then submits the Air Force budget and associated justification books to Congress as part of its contribution to the president's budget. Congress reviews the budget over the spring and summer and may request clarification or justification for programming in the form of inserts for the record (or questions for the record). Congress determines the final programming in the form of an appropriations bill and an authorization bill. The Air Force then executes this programming.

Decisions made by the Air Force throughout this process are influenced by a number of factors. Not all of these factors are objective assessments of Air Force capabilities. One strong influence is institutional inertia. Building a new POM each year through a bottom-up review of requirements is untenable. Hence, previous programming in the Future Years Defense Program (FYDP) strongly influences the current-year POM build. Political concerns and competition among organizations within the Air Force also play a role. Also factoring heavily are the inevitable subjective judgments of experts and senior leaders, as well as the relative persuasive abilities of those who champion programs and articulate their merits.

Some of this subjectivity and rivalry is unavoidable and, perhaps in some instances, even beneficial. Yet a variety of circumstances point to the value of injecting quantitative, objective assessments of capability into the Air Force PPBE process, among them the need to adjudicate among competing programs; the need to provide a robust set of capabilities (and minimal risks) for a given, finite budget; the desire for these capabilities to be balanced among the functional areas; and the need to provide quantitative, objective expressions of the consequences of programming decisions to DoD and Congress. In part to address these issues, the Air Force began the Capabilities Review and Risk Assessment (CRRA) process.

Current Capabilities Review and Risk Assessment

The Air Force recently began formally assessing its capabilities, both programmed and executed, using the CRRA process.[4] The purpose of the CRRA is to identify all the capabilities required of the Air Force and to quantitatively assess their current states. The effort is undertaken through two perspectives.

When viewed from an operational perspective, capabilities are organized into concepts of operation (CONOPS) (see Air Force Instruction 10-2801, 2005). The Air Force defines seven CONOPS: global strike; global persistent attack; nuclear response; homeland defense and support to civil authorities; global mobility; space and command, control, communications, computers, intelligence, surveillance, and reconnaissance; and, underpinning and supporting these six, agile combat support. The organizational structure of the Air Staff that oversees the CRRA follows these operational groupings.

When viewed from a functional perspective, capabilities are organized in the Master Capabilities Library (MCL).[5] The MCL attempts to define an exhaustive set of mutually exclusive Air Force capabilities. The library lists capabilities as tiers, ranging from broad categories down to increasingly specific constituent capabilities. Each broad capability is divided into subcapabilities until a level is reached at which a measure of effectiveness can be assessed. An example will help clarify.

Version 6.0 of the MCL includes eight broad capability groups. These are "Battlespace Awareness," "Joint Command and Control," "Net Centricity," "Force Application," "Focused Logistics," "Force Protection," "Force Management," and "Training."[6] The fifth broad capability, "Focused Logistics," contains a subcategory (indenture 5.5)

---

[4] The primary office of responsibility for the CRRA is the Office of the Director for Operational Plans and Joint Matters (AF/A5X).

[5] The current version at the time of this research was version 6.0, July 2006. The MCL is to be updated for each PPBE even year by September 1. See Air Force Instruction 10-604, 2006, p. 9.

[6] These top-level categories in version 6.0 of the MCL follow the joint functional concepts defined in Chair of the Joint Chiefs of Staff Instruction 3170.01C, 2003, and correlate with the Joint Capability Areas and the areas covered by the Functional Capability Boards.

called “Support the Mission, Forces, and Infrastructure.” This category, in turn, has a tree of further indentures leading down to, for example, indenture 5.5.1.4.2, “Maintain Utility Infrastructure.” Each capability in the MCL is so subdivided until a level is reached that can be meaningfully quantified and represented by a numerical measure of effectiveness.

The CRRA uses the MCL as the starting point for analysis of capabilities and risks. How these assessments are performed has evolved and matured over the past several years. Currently, the central element in the capability assessments is a set of Process Sequence Models (PSMs). PSMs are process maps that indicate the interrelationships of activities that constitute a mission area, such as opening and establishing bases. They are essentially examples of decision networks or influence diagrams. Nodes in the network are activities, or tasks, that must be completed for the mission. Nodes are assigned probabilities of success, and simulations indicate which nodes are most critical, as well as areas of most frequent failure.

These models topically correlate to the CONOPS structure but link to the MCL. For example, for agile combat support CONOPS, there are 10 PSMs that do not reach into the other CONOPS areas but link together elements of the MCL that pertain to agile combat support.

Aside from judgments about what to include in the PSMs and how to link the nodes together into a network, inputs into the PSMs include a probability of success and probability of occurrence for each node. These probabilities are validated by functional assessment teams. Also included are the desired operational outcomes, which derive from the Defense Planning Scenarios developed by the Office of the Secretary of Defense for Program Analysis and Evaluation (OSD/PA&E) and the Joint Staff Directorate for Force Structure, Resources, and Assessment (J8). The analysis is carried out on the current capabilities and future capabilities as specified in the Air Force POM.

The output of the PSM analysis indicates which nodes have the largest effect on the operational outcome. In this way, resource limitations are linked to indicate the proficiency or sufficiency of a capability in a network. In this view, an F-16, for example, is not in itself a capability. Rather, the aircraft, its support equipment, the intelligence needed for a mission, and all the other elements necessary for the F-16 to perform its mission form the overall capability. Only when all these elements are in place and operating is the capability available, and to increase the level of the capability available, it is necessary to invest in the limiting element. It is this kind of insight that the CRRA endeavors to deliver.

A Critical Review of Current Capabilities-Based Programming

As currently implemented, the CRRA provides an expression of the capabilities that the Air Force possesses and the risks it assumes. It has evolved and matured over the past several years. During that maturation, several of the early weaknesses of the CRRA have been ameliorated. Initially, the calendar of the CRRA and the PPBE process were out of phase, so the outputs of the CRRA could not be inputs into the PPBE. These calendars are now synchronized. Earlier assessments of capabilities in the MCL were done independently, with no attention to systems-like interactions of the tasks. For example, there was no apparatus to determine how one capability might impact another. This weakness has been addressed, though imperfectly, with the introduction of the PSMs. Nevertheless, some limitations remain.

In the CRRA, capability assessments remain bound by the subjective judgment of subject-matter experts. Although the risk calculator and PSM analysis are reproducible algorithms, their inputs come from subject-matter experts. These experts have varying familiarity with the subject area, the CRRA process itself, the PPBE process, and the DoD planning environment. A limited number of experts from the field are available to make these assessments. Hence, each expert must weigh in on a wide variety of issues. No expert is capable of assessing accurately the full range of capabilities that are needed. More importantly, because they are *functional* experts, these representatives are, in general, not thoroughly familiar with how resource levels might change in future years in the POM or with the details of the Defense Planning

Scenarios, much less how to assess how much of what resources would be needed to carry out those plans. Thus, this subjectivity leads to lack of repeatability in the CRRA process.

The capabilities are grouped and defined in the CRRA around CONOPS and Air Force functions. The PPBE, on the other hand, is built around program elements and organized around panels. The capabilities assessed and the risks defined in the CRRA do not correspond to these PPBE elements. The CONOPS and panels are misaligned, and capabilities and program elements are not clearly related. These mismatches cause the CRRA to provide the programmers with little detailed insight into how to adjust what they program (i.e., dollars and manpower in program elements) to achieve desired operational effects.

Another consequence of the lack of a relationship between money invested and capabilities acquired is that target levels for capabilities cannot be fiscally constrained. For many capabilities, increasing the quantity or quality of the capability is nonlinear with respect to cost: Getting marginally more capability can be increasingly costly. For example, consider the mission-capability rate of an aircraft. If the rate is quite low, it can be raised with relatively small investments of money, perhaps by increasing the availability of a few critical spare parts. Further raising the rate will become increasingly expensive, up to a point beyond which any amount of money will not increase the mission-capability rate. Not linking capabilities to cost in the form of cost-capability curves limits the programmer's ability to establish the best position to occupy along the cost-capability tradespace in light of desired operational effects.

Further, despite the CRRA's capabilities focus, the process retains some characteristics of the deterministic, threat-based planning that it is meant to replace. The CRRA evaluates how well Air Force functions can achieve a deterministic future as specified by selected scenarios[7]

---

[7] We use the term *scenario* consistently with the definition in U.S. Department of Defense Instruction 8260.01, 2007, p. 6: "An account or synopsis of a projected course of actions or events. For the purpose of this Instruction, the focus of scenarios is on strategic and operational levels of warfare." We use the term *contingency* to describe the individual events that make up a scenario. In this monograph, *deployments* refers to the action of sending those

from the Defense Planning Scenarios. In essence, the combatant commanders' operational plans have been replaced by the Defense Planning Scenarios, with input from the combatant commanders in the form of IPLs. Maintaining a strong connection to plans is inevitable and, although perhaps not in the spirit of capabilities-based planning as some interpret it, perhaps necessary.

The critical aspect of basing programming on a portfolio of capabilities rather than specific threats is the robustness. By *robustness*, we mean the ability to meet a spectrum of threats given the uncertainty of the future security environment. In this sense, the limitation of the CRRA is not that it ties capability assessments to plans (threats), but that it ties them to one set of plans rather than evaluating them against a portfolio of plans (threats).

For a combination of these reasons, perhaps in concert with a certain lack of transparency of the entire process, the CRRA has yet to provide many novel insights into Air Force capabilities or risks, and the confidence in its conclusions has been mixed. Improvements have been made as the CRRA evolves, and further maturation can correct many of these deficiencies.

---

resources to perform a contingency operation outside the United States. When we focus on agile combat support, a deployment requirement is nearly synonymous with a contingency requirement.

CHAPTER THREE

Linking Programming Decisions with Capability Assessments

In this study, we sought a process for achieving the key goals of capabilities-based programming that possesses four core attributes. First, the driving force in determining what gets programmed and at what levels should be how programming adjustments affect operational objectives, not how they impact Air Force tasks. The role of the Air Force is to organize, train, and equip its forces to support national security objectives as outlined in plans. Therefore, the Air Force POM should be constructed to maximally support national-level planning objectives and presented to senior national security leadership in those terms.

Second, the method should be analytically based, reproducible, and responsive within budgetary time frames. It is only through careful analysis that the correspondence between resources and capabilities can be established—and established in a reproducible form. Fragments of such analysis exist for a number of resources throughout the Air Force. In the area of combat support, one example is how the levels of spare parts affect aircraft mission-capable rates. By making the process analytically rather than subjectively based, we do not suggest that programmers abdicate their expert roles in favor of the outputs of algorithms. Rather, we advocate that programming decisions be informed and supported by an analysis of capabilities.

Third, capabilities must be linked directly to what is programmed: dollars and manpower. No matter how accurate and thorough capability assessments might be, if the programmer is at a loss to understand how capabilities relate to program elements, it is unlikely that the POM will be reasonably affected by those assessments. Further, programming does not take place in a fiscally unconstrained environment. Adding capability in one area inevitably affects the ability to deliver capability in another. Such trades have an impact on operational effects by requiring that capabilities be tied to costs (in dollars or manpower) in the form of cost-capability curves. Only when such linkages are quantified do programmers possess adequate tools to identify the operational effects of programming adjustments.

Fourth, the process should embrace the reality that the future is uncertain. The process should not be driven by deterministic plans, whether drawn from combatant commanders' plans or OSD's Defense Planning Scenarios. In these uncertain times, the Air Force POM should be robust enough that the capabilities that it generates are able to meet a wide range of possible threats. The programmer therefore needs an apparatus for evaluating how well a POM will perform against different futures. During programming trades, investments that reduce risk across a wide spectrum of threats should be favored over those that mitigate a small number of less likely threats.

The keystone to satisfying these goals lies in how capabilities are defined and measured. Capability metrics should relate directly to plans; be tied to program elements, groups of program elements, or definable subsets of program elements; and be broad enough to apply across a range of programs. The methodology described in this monograph was developed to address programming issues in the area of agile combat support. For example, do the funded levels of medical support and civil engineering programs provide comparable levels of capability? Or, how do increases (or decreases) in funding levels in fuels support programs change capabilities relative to comparable funding changes in civil engineering? Are sustainment investments sufficient to support all assets acquired? How can resource levels be best set to meet an uncertain future security environment?

For the remainder of this monograph, we focus specifically on capability assessments for agile combat support capabilities. Nevertheless, many of the basic principles apply more broadly and should help structure capabilities-based programming decisions across the Air Force.

Defining Capabilities for Programming

The hallmarks of a good measure of capability are that it is intuitively understandable and that it meets the goals described in the previous section. In this monograph, we define *capabilities* as the set of resources needed to perform an operational-level activity. For example, the set of resources needed to perform a specified major combat operation (MCO), call it MCO-1, would constitute a one MCO-1 capability. For example, if 17 fire trucks of a particular type are deemed necessary for the MCO-1 contingency, then 17 of those trucks constitute a one MCO-1 capability. Similar metrics can be defined for a number of contingencies, including MCOs, small-scale contingencies, humanitarian relief operations, and steady-state deployments, such as drug interdiction and noncombatant evacuation operations, that might not rise to the level of supplemental funding. The capability of a resource is not fixed. It has a value only relative to an operational scenario. Twenty refueling trucks may constitute 0.8 of a particular MCO but 2.3 of a particular small-scale contingency.

This definition is a somewhat elastic use of the term *capability*, but it parallels how the Air Force expresses unit-level capabilities with unit type codes (UTCs). UTCs are initiated by specifying a needed capability via a mission-capability statement. A pilot unit is assigned to determine what manpower and equipment are needed to achieve the specified capability. In this way, a capability and a set of resources are equated. Sometimes, the UTC is used to refer to the capability, other times to the resources. In the same spirit, we use the term *capability metric* to refer both to the operational capability of a set of resources and to that resource set itself, depending on the context.

The current directive from DoD is to program using a set of scenarios called the Defense Planning Scenarios.[1] These are composed of homeland security scenarios and scenarios for MCOs, small-scale contingencies, and steady-state deployments. Each of these scenarios is a unit of capability in the nomenclature presented here. That is, for each of these scenarios, the set of resources needed to perform that scenario

---

[1] U.S. Department of Defense Instruction 8260.01, 2007.

can be determined, and, in that context, the set of resources is equivalent to the capability to conduct that operation.[2]

This definition of *capabilities* meets the goals outlined in this monograph. Operationally defined capability measures naturally tie resource availability to desired operational outcomes. By linking capabilities to resources, capabilities are also naturally linked to costs, both in dollars and manpower. To address uncertain future threats, the analysis of capabilities should consider not just one set of scenarios playing out in a specific time frame, but the full spectrum of scenarios defined in the Defense Planning Scenarios. Finally, how to ground this process in reproducible analysis is the subject of the next section. Before taking up that point, it is instructive to contrast these capability measures with a similar one currently used in the Air Force.

Consider, for example, a commonly used metric for measuring the capabilities that combat support resources bring to the warfighter: the number of bare bases that can be opened and established.[3] While this metric is useful in other contexts, it does not capture the breadth of the objectives included in planning. To see why, consider the data in Figure 3.1.

The figure shows the average number of fuels and bare-base support items used in three recent operations: Operation Enduring Freedom (OEF), Operation Iraqi Freedom (OIF), and Operation Allied Force (OAF). It is not important at this stage to know the specific function of each of the assets. The focus here is on the wide variation in the requirements for these resources per base for different operations. The variance arises principally from two factors: the usage of the base and the existing base infrastructure.

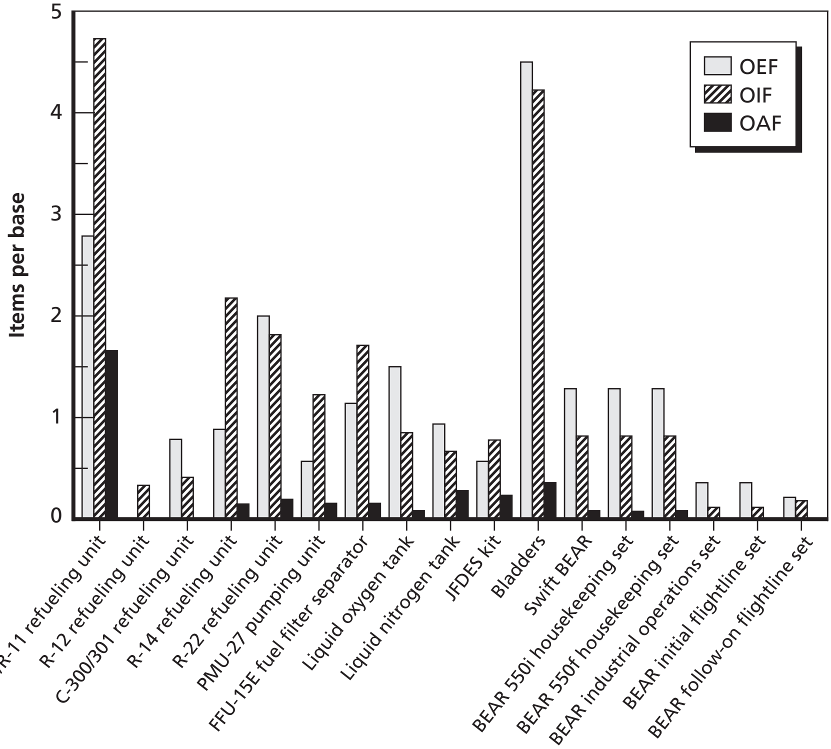
Figure 3.2 shows the great variance in use, expressed in terms of aircraft types and numbers. The figure depicts 30 locations to which the Air Force has recently deployed in support of OIF and OEF. An intrinsic characteristic of these bases is that there is a mix of aircraft

---

[2] We consider a resource a capability only when it is mission capable. We take up the issue of sustainment costs to maintain sets of resources as capabilities later.

[3] Here, we use *open* and *establish* in the sense characterized by force modules (see Secretary of the Air Force, 2006).

Figure 3.1
Items per Base for Three Recent Operations

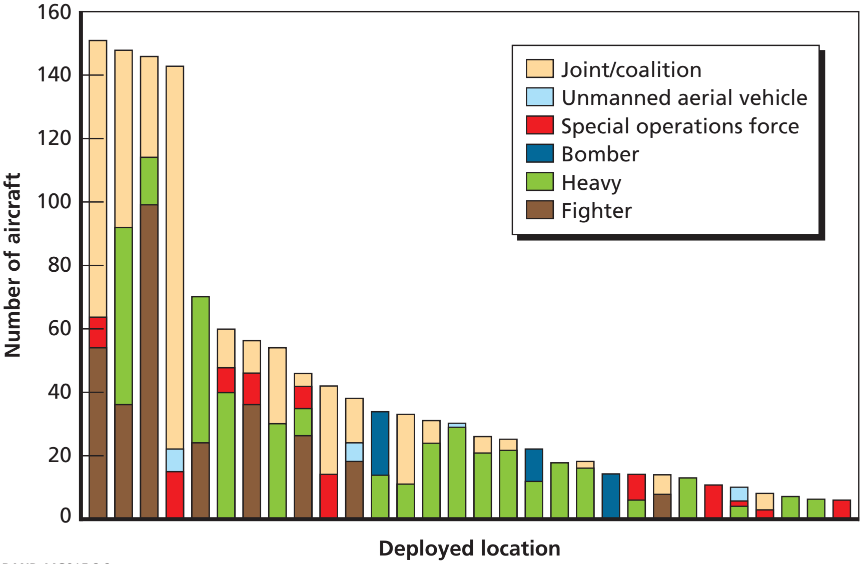


RAND *MG815-3.1*

types, and a large fraction of sites support a number of aircraft from other services and coalition partners. Further, it is striking that there is not a limited number of "typical" bases, or natural sets of bases with similar numbers and types of aircraft; virtually every base is unique.

The amount and quality of prior combat support infrastructure to support these functions also vary considerably, not only from base to base within a theater, but also from theater to theater. The latter effect can be seen clearly in Figure 3.1. OEF and OIF took place in the U.S. Central Command area of responsibility, an area of numerous austere bases and no permanent U.S. presence. OAF, in contrast, took place in the U.S. European Command area of responsibility, a theater with a considerable permanent U.S. presence and virtually no austere bases.

Figure 3.2
Aircraft Mix for 30 Recent Deployed Locations



RAND *MG815-3.2*

Hence, there is no *typical* base to which the Air Force deploys. The number of bases that can be supported varies depending on the type of engagement and the location. These observations suggest a metric that emphasizes operational rather than base-level considerations. For example, capability might be expressed as how many, say, OIF-like operations a resource can support. If resource capabilities are expressed in such terms, rather than metrics with narrower scope, expressions of capabilities of resources as diverse as medical support, civil engineering support, and suppression of enemy air defenses can be examined and traded on a comparable basis that relates directly to planning-level objectives.

The challenge, then, is to determine what resources are needed to perform these Defense Planning Scenario operations. First, there are what we call the *deployment requirements*. These are the resources needed to perform one of these scenarios. Turning again to Figure 3.2, it is necessary to calculate what resources are required for each of the different bases depicted, given the varying infrastructure and operational demands.

Deployment requirements alone are insufficient for achieving all the desired outcomes. Some resources will inevitably break and be in reconstitution at any given time, and some resources are set aside for training or home-station operations and are used for deployments only as a last resort. These additional resources need to be programmed. We call the sum of the deployment requirements and those needed to cover breakage and training *programming requirements*. The next section treats the calculation of deployment requirements, and the following section discusses programming requirements. The next chapter pulls these together in various prototype algorithms.

Matching Resources to Capabilities

We now turn to the heart of the analysis—how to determine what resources are needed to provide a given capability level. Deployment requirements for agile combat support resources can be determined in three ways. First, one can assemble the necessary scores of subject-matter experts and have them interact with the operational experts to create a time-phased list of UTCs, called the time-phased force deployment data (TPFDD).[4] TPFDDs are very expensive to produce in terms of both time and labor. As many as 60 experts may need to be assembled, with the work iterated over the course of weeks or months, to arrive at a viable solution. Part of the difficulty is that requirements for one functional area often depend on others. For example, areas such as medical support and civil engineering require knowledge of the base population as an input to determine their own requirements, but the base population can be determined only by summing all the requirements across all functional areas. This approach is perhaps the most accurate way to estimate deployment requirements, but is not practicable for the examination of the portfolio of possible scenarios

[4] Pronounced "tip-fid."

in capabilities-based programming for an uncertain future security environment.

The second approach is a step toward rectifying this problem but still falls short. Over the past several years, the Air Force has determined the set of time-phased UTCs necessary to support operational activities at an austere location. These groups of UTCs are called *force modules*. Force modules leverage efforts already expended by subject-matter experts, relieving them of reproducing the same analysis each time. Yet, as shown in Figure 3.2, not only are many operations executed from nonausterе bases, but there is no typical base at all. Force modules need to be tailored for each location, and to do that, a set of subject-matter experts must be assembled. Although some time-savings are realized, again, this effort exceeds what is practicable for the flexible treatment of a portfolio of scenarios.

There is a third way, one that we advocate: Extract a set of rules for what resources are needed in deployments, and keep this rules-based algorithm current. This is the approach developed by RAND in the form of the Strategic Tool for the Assessment of Required Transportation (START) (see Snyder and Mills, 2004). The tool calculates a set of UTCs needed to support operations at a deployed location. It takes as inputs characteristics of the aircraft and the location. For the aircraft, inputs are the type and number of aircraft at the location, whether they are bedded down there or use the location as an en-route station, the sortie rate, and the mission type. For the location, inputs are the level of conventional and nonconventional threat to which the base is exposed (high, medium, or low) and some aspects of the infrastructure, such as how much billeting is available, whether a fuels hydrant system is available, and so forth. With this air-order-of-battle level of input, a list of UTCs to support such operations can be produced rapidly. We use this tool to determine the resources needed to meet the full set of the Defense Planning Scenarios.

A natural complication in assigning capabilities to resources merits some discussion. In most cases, resources and capabilities are not uniquely paired. Consider the following four possibilities.

First, a resource or set of resources may provide a unique capability, and that capability may be met uniquely by that resource or set of resources. In mathematical terms, this is a one-to-one and onto (or bijective) mapping of the resource and the capability. Because there is usually more than one way to do anything, examples of strict bijective mappings are few. One example might be explosive ordnance disposal services to a deployed force. If this function is to be provided organically within the Air Force, there is a set of UTCs for this function, and none other can substitute. Nor can these UTCs easily substitute for other Air Force functions.[5]

Second, a resource may be able to provide more than one distinct capability. An example might be an F-16CJ, which can perform suppression of enemy air defenses or combat air patrol.

Third, a capability may be met by more than one resource. For example, a reconnaissance capability might be met by a manned U-2 aircraft, an unmanned RQ-4A Global Hawk, or space-based assets. In another example, fuel delivery may be provided by trucks or hydrants. And, because the Air Force shares many deployed locations with other services or coalition nations, historically, some capabilities are met by resources not organic to the Air Force.

Fourth, the relationship between capabilities and resources might be a mixture of any of these three types.

Many resource-capability relationships fall into the third category: A given capability can be provided by a number of different resources or resource sets. That this situation is common is deliberate. It gives the Air Force reduced risk and greater flexibility. The process of relating resource programming to capability assessments must account for these multiple relationships—in particular, the third.

The model developed in Chapter Four strictly treats the first case. This case shows the essence of the issues involved in the programming problem and is the starting point for modeling the other, more complex cases. These other cases may be nonlinear but should still be handled with standard optimization methods. Whether it is desirable to develop these more complicated models depends on how much they would assist the programmer in making the wisest programming trades.

---

[5] With the exception that airmen in these UTCs could serve some generic duties, such as third-country national escort.

The broader the scope of a capability metric, the more likely it is that a specific capability is satisfied by more than one resource. This, too, points toward a preference for operational-level measures of capability in PPBE programming. For example, if the metric of capability were narrow, such as the level of fuel-pumping capability at a base, there would be ambiguity during programming in terms of the appropriate mix of refueling trucks and hydrants. When the capability metric is specified at the operational level, however, this mix is inherently specified. Different operations will require not only different levels of this refueling capability, but also a different mix. Both this effect and the need to examine uncertain futures point to the utility of the programmer's examination of a spectrum of operational-level capability metrics.

Balancing Procurement and Sustainment Decisions

We now turn to the important issue of sustainment. A set of resources is not a capability unless it is mission capable. Resources in general are occasionally unavailable for use, so the total resource level needed to meet a set of scenarios may exceed the sum needed for each scenario taken together. For equipment, this additional quantity is generally due to breakage or insufficient maintenance. For manpower, the additional quantity may be due to training or a need for recovery time after a deployment. In this monograph, we focus only on equipment, but the broader principles apply to manpower.

The rate at which equipment breaks, needs maintenance, or both is strongly determined by the frequency and type of use. The frequency, duration, and distribution of scenarios in time determine not only the deployment requirements, but also the quantities and costs involved in maintenance and repair. Since timing plays such a central role, we develop the concepts of scenario timing in some detail.

To examine quantitatively the impact of the overlap of contingencies in time, we need to establish a nomenclature for resource demands as a function of time. For a particular contingency, numbers of requests for a resource, i, can be summed over specified time intervals beginning at t. We call these requests d_{it}^{+}. If the analysis is at the UTC level, the deployment requests would be specified by a TPFDD and by binning those requests over some time interval (e.g., per month). The + superscript denotes that the item is deploying; a positive amount is therefore required. The total amount of resource i deployed up to time t is then the sum of all such deployment requests:

$$D_{it}^{+} = \sum_{\tau=0}^{t} d_{i\tau}^{+} \quad \forall i,t. \tag{3.1}$$

Once no longer needed for an operation, resources are redeployed in a reverse process relative to the deployment. We use a parallel nomenclature for redeployment[6] requests, d_{it}^{-}. The total amount of i redeployed up to time t is

$$D_{it}^{-} = \sum_{\tau=0}^{t} d_{i\tau}^{-} \quad \forall i,t. \tag{3.2}$$

The sum of the cumulative deployments and redeployments, then, gives the total simultaneous demand for resource i at time t:

$$D_{it} = D_{it}^{+} - D_{it}^{-}. \tag{3.3}$$

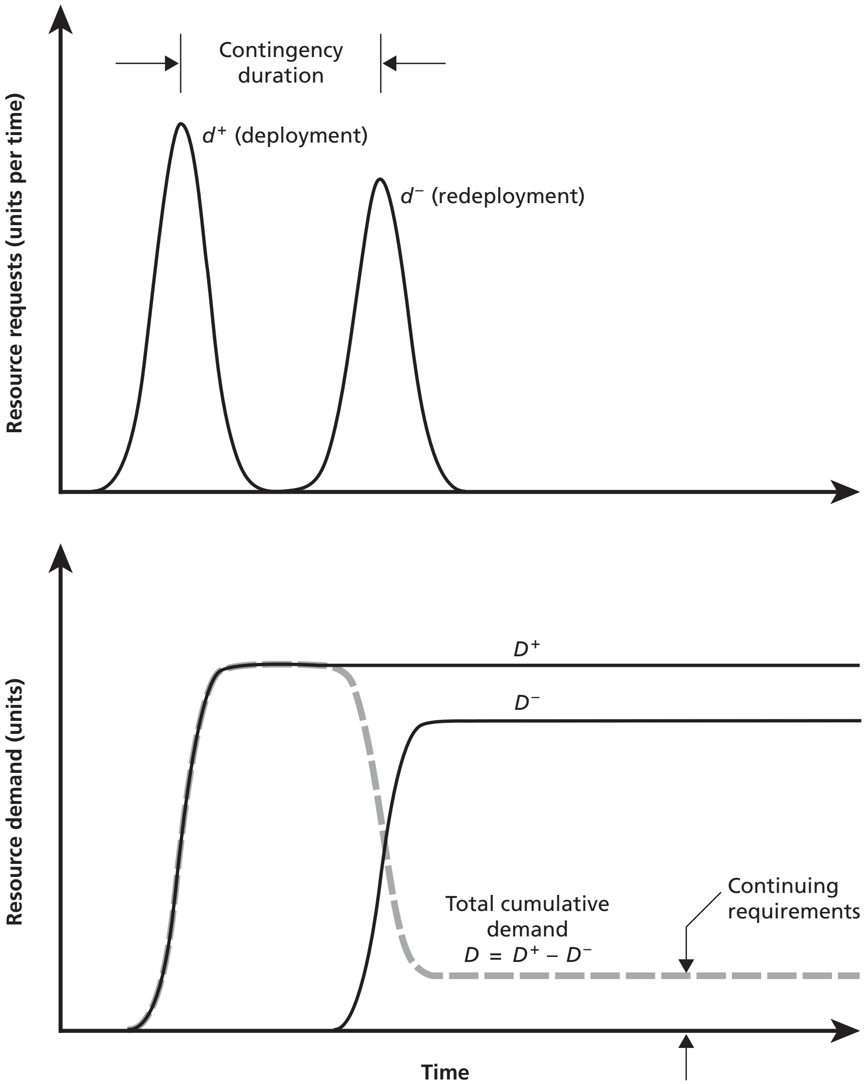
Figure 3.3 shows the relationships among these variables for notional deployment and redeployment requests.

Note that the contingency has a peak, given by the sum of all deployment requests, and a duration that we define (arbitrarily) as the time lapse between the peak in the deployment and redeployment request curves. The notional case shown in Figure 3.3 indicates an instance in which the redeployment requests are fewer than the deployment requests over the planning time horizon (usually the duration of

---

[6] We use the term *redeployment* to mean returning the resource from a deployment for possible reconstitution. We exclude from this the moving of a resource from one contingency directly to another contingency.

Figure 3.3
Definitions of Resource Deployment and Redeployment Demands



RAND *MG815-3.3*

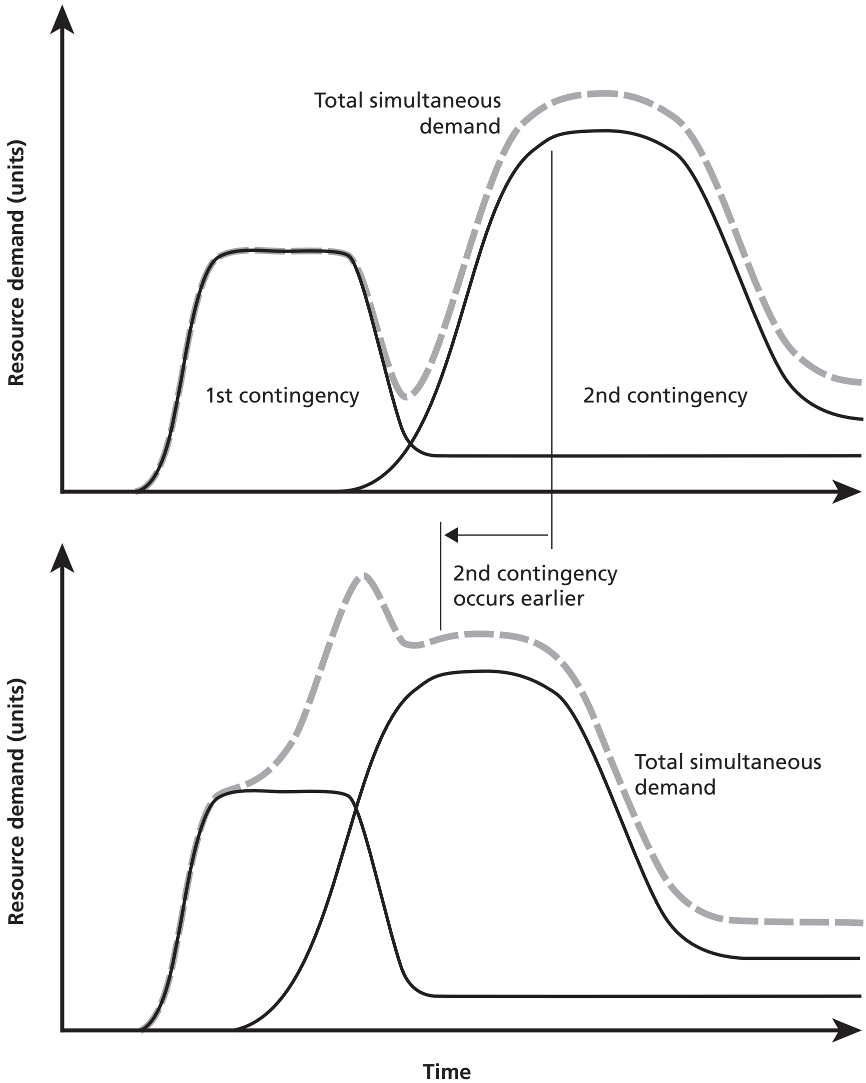
the FYDP). This difference arises from ongoing commitments incurred from the initial contingency and causes D to remain nonzero. In the case shown in the figure, D remains nonzero over the remainder of the planning timeline. A recent case example of this phenomenon is Operation Southern Watch, the enforcement of the southern no-fly zone in Iraq, which required relatively constant resources for roughly a decade following the First Gulf War. We call such requirements *continuing requirements*.

Each resource has a set of cumulative demand curves such as those depicted in Figure 3.3. Values of timing, magnitude, duration, continuing requirements, and so forth will generally be different for different resources. Fully specifying the requirements for a scenario involves specifying a set of such curves for all resources involved in all of the applicable Defense Planning Scenarios.

The Defense Planning Scenarios call for dealing with multiple types of contingencies, and the total demand for a resource is given by a linear combination of all contingency requirements. If these always occurred separated by intervals of time and if no resources required reconstitution, the job of the programmer would be simple—to meet the demand for the largest contingency over the planning horizon (FYDP). This temporal coincidence cannot be assumed, however, because contingencies might occur simultaneously or nearly simultaneously. Figure 3.4 shows the total cumulative demand for a resource determined by two different contingencies. The upper and lower panels illustrate two contrasting cases of temporal propinquity. The upper panel shows a notional case in which the events are sufficiently separated that the peak demand is dominantly determined by one of the contingencies. The lower panel shows a case of some temporal overlap, such that the maximum demand for the resource occurs between the maximum demands of the two contingencies.

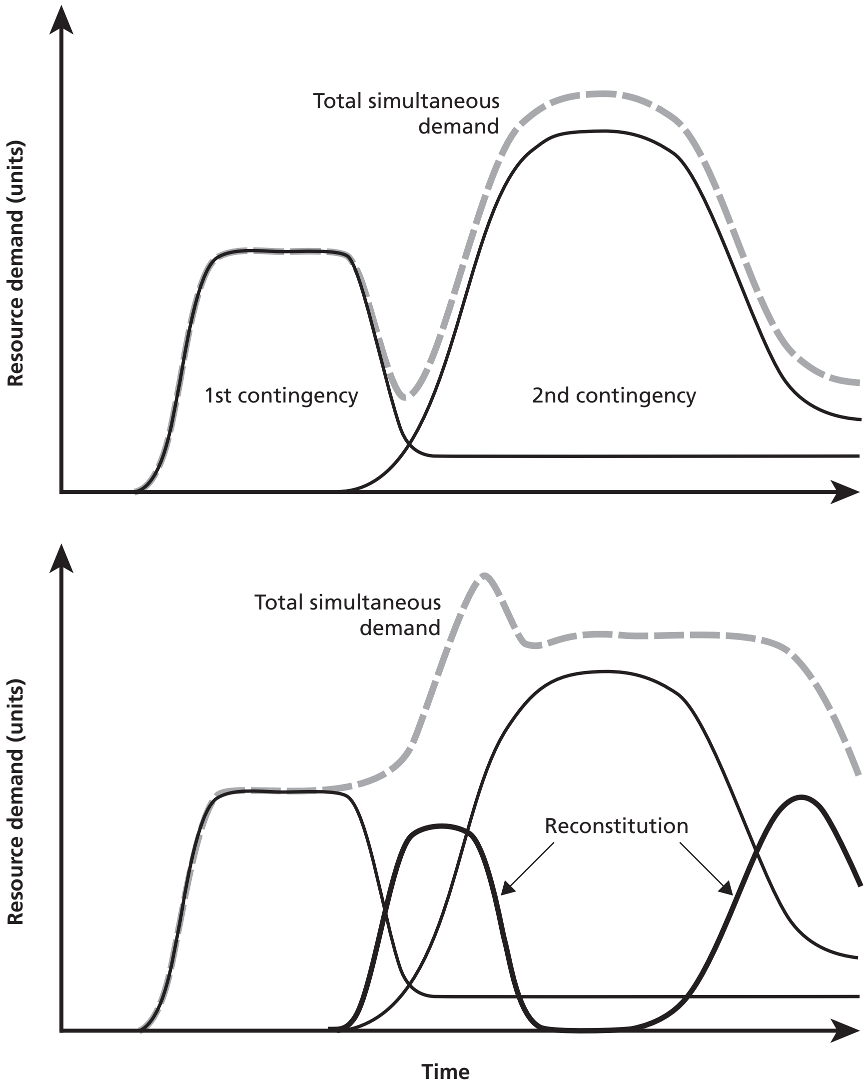
In addition to the temporal overlaps of contingencies, the time to reconstitute resources after contingencies also plays a significant role in determining resource demands. When a resource redeploys, it enters a reconstitution pipeline of duration $l_i^R \geq 0$ (the superscript, *R*, refers to reconstitution lead time), which makes it unavailable for deployment for time l_i^R. As shown in Figure 3.5, filling a reconstitution pipeline is equivalent to extending the duration of the conflict by l_i^R for that resource (compare the lower panels in Figures 3.4 and 3.5).

Figure 3.4
Effects of Timing of Contingencies on the Demand for a Notional Resource



RAND *MG815-3.4*

Figure 3.5
Effects of Finite Reconstitution Time on the Demand for a Notional Resource



RAND *MG815-3.5*

The probability (or frequency) of occurrence of these contingencies is unknown. To work in this environment of uncertainty, the programmer also needs the flexibility to explore different frequencies of occurrence of each contingency type, as well as the timing and duration.

Two additional factors involving time play a role in determining resource requirements and the necessary programming to meet them. First, the length of the planning timeline (FYDP) can influence programming decisions. A goal of meeting the requirements of a challenging scenario by the close of the first year of the FYDP is much more ambitious than meeting the same requirements by the end of the FYDP. Second, the resources needed for a contingency are a subset of the total resources needed at a given time. Also needed is the set of resources required to train the forces to meet future contingencies. This set of resources also includes all those resources needed for home-station operation beyond those available for deployment.

Curves for home-station and training requirements will tend to be more constant over time than demands resulting directly from contingencies, but they may increase or decrease as a function of concurrent contingency operations. For example, deployment of aircraft may decrease aircraft support needs at a home base, but a terrorist attack, such as the one that occurred on September 11, 2001, and the resulting response of OEF and Operation Noble Eagle, may increase home-station force protection requirements concurrent with that same resource being called upon for deployments.

To summarize, resource demands are determined by the nature of the anticipated contingencies, their locations, and the training required for readiness. For each Defense Planning Scenario (and to satisfy its associated training and readiness component), the total demand for a given resource varies as a function of the timing, which has five important components: (1) the potential temporal propinquity of contingencies, (2) the time available to prepare for each contingency, (3) the duration of the contingencies, (4) the reconstitution time necessary to recover from previous contingencies, and (5) the frequency of occurrence of each contingency type. Programming decisions that integrate capability assessments must be able to handle this spectrum of factors.

Given a set of resource demands as functions of time, the programmer needs to consider a range of attributes of each resource to determine how to distribute programming funds to meet these anticipated requirements with a robust set of capabilities. Next, we discuss these resource attributes.

Salient Resource Attributes for Procurement and Sustainment Decisions

The Air Force monitors countless attributes of its equipment resources and catalogs these properties in numerous databases. We sought a minimal list of attributes necessary to determine optimal programming decisions and that balance investments between procurement and sustainment across functional areas.

Resource attributes should possess broad applicability to most resources, be expressed by similar measures, and capture the first-order properties that most influence a programmer as he or she makes programming decisions and trade-offs based on the capabilities that those resources provide. For example, the rate and the general state of wear or aging of equipment factor into when an asset needs to be replaced. The natural units of measurement vary with equipment type. Wear or aging might be naturally measured in units of time (as for a tent), the number of deployments (as for a fuel bladder), or number of miles driven (as for a fuel truck). A program element manager needs data collected in a natural form for each asset. For the purposes of trading these assets, however, the programmer needs a common scale. In this instance, the assets will have an expected lifespan in time, number of deployments, or miles. A common measure is useful for comparison: a value for wear or aging scaled by the lifespan.

Further, for general economies in both data collection efforts and modeling complexity, this list of attributes should be kept to the minimum needed to maintain the level of fidelity of the trades being modeled. That is to say, a sound modeling technique does not require increasing the number of input parameters or complexity unless it is accompanied by a commensurate increase in insight. We have identified four such attributes: attrition rate, procurement time, reconstitution time, and costs (of procurement, operations and maintenance [O&M], and reconstitution).

Attrition Rate

While in use, material items, to varying degrees, reach a condition wherein they are no longer mission capable. At this juncture, they are either reconstituted or, if they are beyond a state at which it is feasible for them to be refurbished, condemned. The point at which each item type arrives at one of these states varies. Some items, such as fuel bladders, are used once and discarded. Other items have lifespans determined by age, mileage, or frequency of use.

Although expected lifespans are clearly of central importance in programming decisions, these data can be quite difficult to estimate because they depend strongly on wear or aging. The collection of such data is inconsistent across the agile combat support areas. In general, there are few data indicating what drives attrition (e.g., time, frequency of use), what the expected life cycle is, or where each resource resides in that life cycle. Despite this dearth of data, it is not possible to balance procurement and sustainment costs without such insights.

Times for Procurement and Reconstitution

Material items not condemned after use during deployments enter a reconstitution pipeline for some length of time. This time effectively extends the deployment duration for an additional period during which the resource is unavailable for use. If l_i^R is the time spent in reconstitution for resource i, the amount of resource i entering the reconstitution pipeline at time t is $d_{i,t}^-$, and (assuming that the resource was promptly reconstituted) the amount leaving is $d_{i,t-l_i^R}^-$. The total amount of resource i in reconstitution at time t is then

$$\Re_{it} = D_{it}^- - D_{i,t-l_i^R}^- . \tag{3.4}$$

The longer the reconstitution time, the more of that resource that must be procured to fill the reconstitution pipeline. The time for reconstitution for a given resource is not a constant and can vary for a number of reasons. First, dollars must be spent to reconstitute a resource. A decision could be made to defer reconstitution and use these funds to increase capability in another area, thus extending the reconstitution time (in the extreme case, to infinity). Second, there are circumstances in which a monetary investment can increase reconstitution capacity, thus buying a decrease in the reconstitution time and reducing the total inventory required to achieve a specified capability level. We have not included this latter option in the current model. Incorporating these trades into the model makes the algorithm significantly more complicated. Given that trading among capacity, sustainment, and procurement is less frequent than trading between sustainment and procurement, we have opted to leave this option for future work.

Costs

Because programming decisions are constrained by fiscal guidance, the key attribute for trading among resources for most programming decisions is cost. For equipment resources, we consider three cost types: (1) costs to acquire new items, (2) O&M costs, and (3) costs to reconstitute items after use during deployments (and recapitalization).[7] The budgets for these activities are interrelated. Purchasing an item in one year incurs O&M costs until that resource is retired. Furthermore, O&M expenditures on one item can be deferred (lowering its availability) in exchange for purchasing another (raising the inventory) to match capabilities.

All of these costs can vary according to conditions in the industrial base. Procurement costs, in particular, may vary according to buying patterns over time. If a sole supplier has to shut down a production line between Air Force buys, the pricing may increase substantially. These effects need to be considered in determining optimum programming over the FYDP. In this monograph, we consider costs to be constant, not a function of the state of the industrial base.

Finally, although it was beyond the scope of this study, modernization issues also play a role in programming trades. Some resources

---

[7] There is a fourth cost—costs to modernize the inventory; that is beyond the scope of this study.

have a finite lifespan due to such factors as technical obsolescence (e.g., computers, other electronics) and marginal utility or shifting priorities in light of changing world conditions. It is undesirable to make large capital investments in resources that have a high likelihood of being phased out in the near future. It is more desirable to buy these items "just in time." Quantifying the likelihood of obsolescence for all resources is not possible, but, for resources that are obvious candidates for faster obsolescence (e.g., computers) or slower obsolescence (e.g., tents), these factors should enter into programmers' decisions about procurement priorities.

CHAPTER FOUR

Algorithms for Capabilities-Based Programming

A Methodology for Capabilities-Based Programming

We now have all the ingredients to build a capabilities-based POM for agile combat support equipment. The programming goals are set by a portfolio of Defense Planning Scenarios that define operational-level capability metrics. Resources are tied to these scenarios via a rules-based approach that assigns UTCs required from air order of battle–level inputs. Linking capabilities to resources ipso facto links capabilities to programmable units and costs. These costs derive from both the need to procure new assets and the need to sustain existing assets. Procurement costs derive from deployment requirements, reconstitution pipelines, and current stock levels. The sustainment costs derive from the frequency of use specified in the Defense Planning Scenarios and empirically determined attrition rates. The factors that drive sustainment costs also determine the reconstitution pipeline—just one way in which all these ingredients mutually interact in a complex programming system.

The challenge for the programmer is to disentangle and balance all of these factors—and to balance them not just within a given program element, but also among the full set of program elements that constitute the Air Force POM. In this chapter, we develop algorithms that synthesize these ingredients into capabilities-based POMs. These algorithms can also be used to evaluate how a candidate POM would perform against a set of desired capabilities (operational scenario sets). We develop three approaches, each of which provides a different kind of insight into programming decisions. These approaches are distinguished by how they treat the future planning objectives and whether the optimization maximizes capability or minimizes costs.

The first approach minimizes costs (spending for procurement and sustainment) while fulfilling all the capabilities required by a set of planning scenarios, subject to constraints on spending fluctuations from year to year in the FYDP. In this case, the set of planning objectives includes some subset of the Defense Planning Scenarios that constitutes one possible future for which the United States might prepare.

The second approach maximizes capabilities defined by a set of planning scenarios subject to fiscal constraints. In this case, spending limitations may cause shortfalls in the ability to carry out all the desired capabilities, or spending may be in abundance, leading to a surfeit in capabilities as defined by the planning objectives.

Both of these approaches build a program against a deterministic future. While providing some important insights, especially if done multiple times with different sets of planning scenarios, these approaches do not capture the full essence of robust planning for an uncertain future security environment. We call these *single-scenario set* approaches.

The third approach develops a robust program in the face of an uncertain future. This algorithm maximizes capabilities against a portfolio of possible futures *simultaneously*, subject to fiscal constraints. Whereas the second case maximizes capabilities against one future, the third case does so simultaneously against a portfolio of futures. We call this a *robust* approach.

The remainder of this chapter presents in more detail each of these programming approaches. The following chapter shows how these algorithms can be used to inform programming decisions.

Modeling Approach

Each of the approaches outlined in the last section involves the simultaneous need to seek minimal or maximal values of several variables subject to constraints. Such problems lend themselves to the analytical technique of optimization. The dual nature of the objectives suggests two modes of optimization: one that minimizes the net-present value of costs subject to meeting all anticipated capability requirements and another that maximizes the global minimum of capability (over time and resources) subject to budgetary constraints (e.g., specified budgets, constraints on yearly variations in program budgets).

For all modes of optimization, anticipated capabilities must be specified. Given uncertainties in the types, locations, and timing of contingencies, we leave these as user-specified inputs in our model. Contingencies can be specified according to capability metrics, from either the Defense Planning Scenarios or, for exploratory analysis, historical contingencies (e.g., OIF). This flexibility allows the programmer to explore the implications of various planning forecasts on programming and vice versa.

We employ linear programming (LP) to find an optimal[1] choice of purchase decisions given a predetermined set of contingencies. Solving the optimization with deterministic demand and an LP formulation lends itself to rapid solutions to industrial-scale problems. Thus, LP satisfies our desire to look across a broad range of Air Force resources and provides rapid analysis to the programmer.

Our approach is lissome enough to deal with the intrinsically nonlinear components of this problem by using linear constraints—in particular, the feedback between procurement decisions and pricing resulting from procurement patterns over time that can affect the state of the industrial base. We believe, nonetheless, that the advantages of maintaining linearity in the mathematics outweigh any benefits that may accrue by introducing a pricing nonlinearity. These pricing issues can still be addressed by adjusting linear parameters, such as setting one price for constrained procurement (e.g., forcing a certain minimum purchasing level at all times) and another price for unconstrained procurement (e.g., allowing procurement to vary from zero to any value within overall budget constraints). This allows the programmer

---

[1] By *optimal*, we mean the best programming that meets the specified planning objectives given the modeling assumptions. It is not optimal in the sense of considering all factors, such as political implications.

to explore the effects of variable pricing due to the state of the industrial base but maintains the enormous advantages of linearity.

The LP approach assigns continuous rather than discrete values to all variables (e.g., purchasing, inventory levels). Hence, within the algorithm, it is possible to buy a fractional amount of an item. This approximation is acceptable for assets with large inventories, such as those considered in this monograph. Algorithms that force integer solutions may be solvable only for smaller problems. Given the large inventory levels considered and the desire to analyze numerous resource types simultaneously, admitting continuous variables outweighs the trivial benefit that integer calculations would add. We note that, in more complex cases, such as when a capability can be met by more than one resource, either more complicated models may be needed (nonlinear or integer models) or, if these are intractable, continuous linear models will need to be combined with expert judgment.

Structure of the Prototype Software

This prototype programming optimization tool merges code written in Microsoft® Excel® with Visual Basic® for Applications (VBA) and the General Algebraic Modeling System (GAMS).[2] As described earlier, the calculation is a linear programming optimization. This computation is done in the GAMS engine and uses an Excel shell as a graphical user interface.

The tool's flow comprises four main steps. First, the user specifies a set of planning contingencies that create anticipated resource demands as a function of time. This step is followed by a choice of the mode of optimization, of which there are three: (1) costs can be minimized subject to always meeting the objectives of a single future, (2) capabilities can be maximized against a single future subject to budget constraints, or (3) capabilities can be maximized against a portfolio of futures. An additional option is to perform no optimization at all. This latter mode

---

2 GAMS is a product of GAMS Development Corporation.

is useful for examining the implications for a specified POM on future capabilities against sets of possible contingencies.

Next, VBA code assembles and records this information in text files and GAMS code. GAMS code is selected during execution without the need for user intervention. After these inputs, GAMS loads the data, performs the optimization, and assembles and records its outputs. Finally, Excel, using VBA code, formats and displays the final results. The remainder of this chapter describes these steps in greater detail.

Resource Demands

The user specifies the total demand for resources by building linear combinations of resource sets (deployment requirements) that can provide the capabilities specified by the capability metrics in Chapter Three. These capability metrics are lists of resource units, generally at the UTC level, as a function of time. In some cases, the demand over time may be a constant. Capability metrics can be drawn from the Defense Planning Scenarios, home-base requirements, training requirements, historical operations (e.g., OEF, OIF, OAF), and force modules.[3]

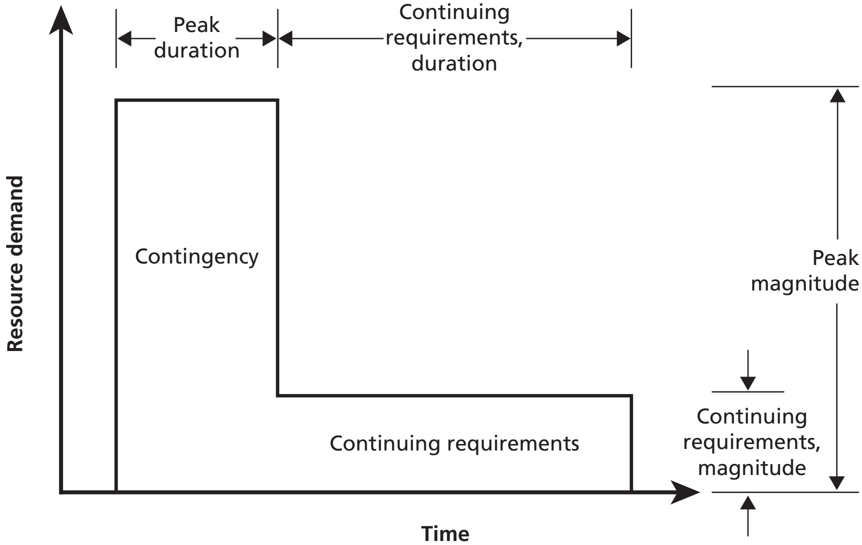
The resource requirements corresponding to these capability metrics can be parameterized in timing and scale. We call each sum of the linear combination of these parameterized resource sets (capabilities) a *demand stream*. For example, for each capability metric, the user may specify when the contingency starts, as well as four adjustable parameters: the magnitude of the peak requirement (the sum of all deployment requests), the duration of that peak (the time elapsed between first deployment and first redeployment), the magnitude of the continuing requirement, and the duration of the continuing requirement.

Figure 4.1 depicts these adjustable parameters graphically for a notional contingency. All parameters scale the demand stream with the capability metric for each resource. Hence, each resource will have

---

[3] The force module metric includes the open, establish, and command-and-control modules.

Figure 4.1
Contingency with Adjustable Parameters



RAND *MG815-4.1*

a distinct demand stream that will, in general, be unique. This parameterization of the capability metric allows the programmer to explore a wider range of demands than specified by the metrics themselves. The total demand for an asset at any time is the sum of all the specified demand streams.

Resource States and Attributes

During optimization, a decision is made in each time step (one month in duration) regarding how much to procure and reconstitute, if any, of each of the resources. Once a resource is procured, future money is obligated for its sustainment costs. That is to say, the model forces O&M spending on all existing assets that are available to deploy or are deployed.[4] Future modifications might relax this constraint, making

[4] O&M costs are not assessed on broken assets (i.e., assets awaiting or in reconstitution).

O&M a decision variable. The time horizon for the model is set at six years, the duration of the FYDP in even years.[5] These procurement and reconstitution decisions are based on resource attributes and anticipated demands, as described in Chapter Three. To clarify the details, we now follow a resource through the algorithm.

At any time, each resource is uniquely in one of four states: awaiting deployment, deployed, awaiting reconstitution, or in reconstitution. A resource is awaiting deployment if it is in storage or at a home station but not being used. We do not currently distinguish between storage (e.g., war reserve materiel) and unit assets. We charge O&M expenses on all assets not awaiting or in reconstitution. An item is deployed if it is being actively used in one of the user-specified scenarios (whether in a contingency, supporting home-base operations, or training at home or abroad). A resource is in reconstitution if it is in any way being reconditioned or replaced or if it is awaiting such reconstitution and is unavailable for immediate use. During each time step, the amount of resources in each state will generally change. These changes occur through either active decisions of the algorithm or passive factors.

The active decisions are how many assets, if any, to procure in each time step and whether to reconstitute broken assets. For some capabilities, existing assets may be out of balance with what is required to support plans. If an asset is in surplus relative to plans, when that asset is redeployed, it may be desirable to forgo O&M costs or not reconstitute it and, instead, use these funds to buy capability in another area. For example, if 20 units of an asset exist and projections of needed capabilities never call for more than 15, we allow the model to suspend reconstitution on that asset until the mission-capable stock level reaches 15, and instead allocate those funds to maintaining or procuring an asset that is in short supply relative to plans. This option seems reasonable for many agile combat support assets, and hence we include this option in the model. Some assets (for example, assets of great capital expense)

---

[5] Although output results are only for the period of the FYDP, the model runs are carried out beyond the FYDP by six more years. This extension ensures that the consequences of decisions made in the latter periods of the FYDP are considered. Without such a feature, the model might choose to procure nonoptimal assets in the last time steps, since it would not have to consider the sustainment costs of those assets.

may always undergo reconstitution upon redeployment, and for those cases, the programmer may wish to suspend this option in the model.

Likewise, it may be desirable to sell an asset and use those funds to increase the capability of another resource. The algorithm currently does not provide options for such fungible trades. This limitation can easily be relaxed in future versions.

Passive effects on resource states are that assets can be terminally removed from the inventory by attrition. We have included two attrition rates. One is a constant rate assessed during each period on all assets in all states except reconstitution. To keep the model simple, we assess this kind of attrition by incorporating this cost in the operations and sustainment costs of the item. Note that modeling in this manner forces the attrited assets to be replaced—and replaced with sustainment funds. For some resources, a programmer might wish to relax this assumption, in which case the model can be expanded to incorporate this attrition separately.

The other attrition rate is a one-time assessment of terminal breakage at the time of redeployment. This rate is an estimate of what fraction of a resource is generally returned or salvaged after a deployment. Planning figures for such a rate are difficult to estimate, as each deployment differs considerably in terms of the wear and damage to materiel, and the United States sometimes makes the strategic decision to leave some assets behind for host-nation use as a goodwill gesture.

Optimization Modes

Minimizing Costs

The first optimization option is to minimize costs in terms of net-present value while meeting all planning requirements at all times. We minimize Y, the sum of the discounted costs over all time:

$$Y = \sum_{t} \left(1+\gamma_t\right)^{-t/12} B_t, \tag{4.1}$$

where γ_t is the annual real discount rate[6] and B_t is the allocated budget at time t (measured in months). Notation is summarized in Table 4.1. The minimization is subject to budget and several ancillary constraints. The budget,

$$B_t = \sum_i \left[c_i P_{it} + o_i \left(S_{it} + D_{it} \right) + r_i R_{it} \right] \quad \forall t, \tag{4.2a}$$

for each time step is the sum of the procurement, maintenance, and reconstitution costs for all items. Under the same formalism, another option for the budget that distinguishes O&M costs for deployed and nondeployed assets is

$$B_t = \sum_i \left[c_i P_{it} + o_i^S S_{it}^+ + o_i^D \left(D_{it} - S_{it}^- \right) + r_i R_{it} \right] \quad \forall t. \tag{4.2b}$$

This latter expression must be used with care. It is valid when $o_i^S \geq o_i^D$, and therefore is most useful when the programmer wishes to assume that the O&M costs of deployed assets will be paid out of supplemental appropriations, in which case $o_i^D = 0$.[7] To ensure some consistency in annual budgets, we introduce notation for a time index in years, t^*, and annual budgets, $\bar{B}_{t^*}$. The annual budgets can be smoothed over time with the constraint that spending in all years must not deviate from that of the prior year by more than a user-specified "float" parameter, f:

$$\left(1 - f\right) \bar{B}_{(t-1)^*} \leq \bar{B}_{t^*} \leq \left(1 + f\right) \bar{B}_{(t-1)^*} \quad \forall t^* > 1. \tag{4.3}$$

Without this constraint, all purchasing would be done "just in time" for the contingencies, and budgets would fluctuate accordingly.

---

[6] We use the (10-year-based) annual real discount rate of 2.6 percent from Office of Management and Budget, 2008. Current rates are updated annually.

[7] It is possible to distinguish O&M costs of deployed and nondeployed assets more generally. In some cases, this may be an informative distinction. For the assets analyzed here, the scenario inputs and costs of procurement and reconstitution drive the decisions more so than do the O&M costs, so we have opted for a simpler formalism. The reason that $o_i^S \geq o_i^D$ must hold in Equation 4.2b is that otherwise, the term S_{it}^- would have a higher weight than S_{it}^+ in the budget constraint, and the lowest feasible value for S_{it}^+ would no longer be ensured.

Table 4.1
Notation for Single-Scenario Algorithms

Symbol	Meaning	Units
b_t	permissible budget at t	$
B_t	allocated budget at t	$
$\overline{B}_{t^*}$	annual budgets, $\overline{B}_{t^*} = \sum_{t=12t^*-11}^{12t^*} B_t$	$
c_i	purchase price of i	$
d_{it}^+	amount of i deployed at t	
d_{it}^-	amount of i redeployed at t	
D_{it}^+	$\sum_{\tau=0}^{t} d_{i\tau}^+$	
D_{it}^-	$\sum_{\tau=0}^{t} d_{i\tau}^-$	
D_{it}	$D_{it}^+ - D_{it}^-$	
f	parameter to smooth temporal budget fluctuations	
i	asset index	
k_i^R	condemnation rate of i at the time of redeployment	
l_i^P	purchase lead time for i	months
l_i^R	reconstitution lead time for i	months
m_i	optimization capability metric	
o_i	O&M cost of i	
o_i^D	O&M cost of i when deployed	
o_i^S	O&M cost of i when not deployed	
P_{it}	amount of i purchased at t	
q_{it}	programmed buy of i at t	

Table 4.1—Continued

Symbol	Meaning	Units
r_i	reconstitution cost of *i*	$
R_{it}	amount of *i* entering reconstitution at *t*	
S_{it}	available amount of *i* at *t*	
S_{it}^{+}	positive component of S_{it}	
S_{it}^{-}	negative component of S_{it}	
S_{it}^{*}	amount of *i* awaiting reconstitution at *t*	
t	time index (months)	
t^{*}	time index (years)	
X	global minimum of capability $\forall i,t$	
Y	total budget in net-present value	$
γ_t	real discount rate for each period	
τ	index of summation over time	

The residual stock level for each time period is set by

$$S_{it} = S_{i(t-1)} - d_{it}^{+} + P_{i\left(t-l_i^P\right)} + R_{i\left(t-l_i^R\right)} \quad \forall i,t > 1. \tag{4.4}$$

Note that, in expressions that include references to a previous time step, the initial time step is specified by an initial condition, not the general expression. For example, in this case, S_{it} is given by the user-specified initial stock level, not by Equation 4.4. This stock level must remain non-negative if all requirements are met, but to preserve generality, we divide the stock level into positive and negative components in order to avoid assessing O&M costs on negative stock levels when solutions with shortfalls are admissible. Hence,

$$S_{it} = S_{it}^{+} - S_{it}^{-} \quad \forall i,t, \tag{4.5}$$

with the constraints that

$$S_{it}^{+} \geq 0 \quad \forall i,t \tag{4.6}$$

$$S_{it}^{-} \geq 0 \quad \forall i,t. \tag{4.7}$$

The amount of stock awaiting reconstitution is defined as

$$S_{it}^{*} = S_{i(t-1)}^{*} + \left(1 - k_{i}^{R}\right) d_{it}^{-} - R_{i(t-1)} \quad \forall i,t > 1. \tag{4.8}$$

The model decides as part of the optimization to spend money to reconstitute these items or to allow them to await reconstitution until it is optimal to do so. Hence, we get the additional constraint that the number of items selected for reconstitution cannot exceed those awaiting reconstitution:

$$R_{it} \leq S_{it}^{*} \quad \forall i,t. \tag{4.9}$$

We further allow the model to force purchases at the user's discretion. For example, the user might specify that a certain number of units of a given asset must be purchased in a certain fiscal year. We call these forced purchases q_{it}, which gives rise to the obvious constraint that the total purchase decisions must equal or exceed these forced purchases:

$$q_{it} \leq P_{it} \quad \forall i,t. \tag{4.10}$$

Note that the forced purchases can be zero, but they cannot exceed any budget constraints. Finally, the decision variables—the number of acquisitions, P_{it}, and the number of reconstitutions, R_{it}, must be nonnegative:

$$P_{it} \geq 0 \quad \forall i,t \tag{4.11}$$

$$R_{it} \geq 0 \quad \forall i,t. \tag{4.12}$$

Finally, to ensure that all requirements are met at all times,

$$S_{it} \geq 0 \quad \forall i,t. \qquad (4.13)$$

Maximizing Capability

Optimization of PPBE programming that maximizes capability subject to budget constraints is a bit more complicated. The two principal decisions in developing the formalism for optimizing capability are the form of the objective function and the choice of the capability metric used to weight each resource. This metric is the entity that enables a comparison of disparate resources on an equal basis, and is needed to make informed programming trades. Consider, first, the choice of the form of the objective function.

Deciding on the objective function is fundamentally about valuing when a shortfall of capability is most dire. If, in the judgment of the programmer, a small but chronic depletion in a capability is deemed worse than a large, acute shortfall during an MCO, a natural choice of objective function would be maximizing the average asset position across all periods for all assets. Such an optimization function has an obvious deficiency, however: It permits increasing the time-averaged capabilities by accumulating enormous stockpiles of cheap assets while allowing significant shortfalls of expensive assets. Such a solution is clearly not consistent with Air Force objectives. Furthermore, we feel that acute shortfalls at times of highest demand generally take priority over small, acute shortfalls during times of peace.

These considerations lead to an objective function that maximizes the global minimum capability of all assets over all time, and this is the objective function of the prototype tool used for the illustrative calculations in the next chapter. Yet, even this objective function has a notable limitation. When the global minimum residual capability is a negative quantity, all the effort of the optimization focuses on increasing this quantity. The residual supply of all other assets at all other points in time is not explicitly considered so long as it does not produce a residual capability more negative than the present global minimum. In particular, decisions to purchase assets or reconstitute assets at times past the global capability minimum do not affect the global minimum; thus, they are not optimally constrained. Due to this aspect of the optimization, we cannot strictly interpret the minimum residual capability across all assets at other points in time besides the global minimum as truly optimized. It is possible that there is yet room to increase the residual capability of these assets at other points in time, but the optimization has not taken care to do so because it would not improve the objective (maximizing the global minimum). If the global minimum (maximum demand) occurs near the end of the planning horizon (FYDP), these problems are mitigated.[8]

Stated mathematically, we maximize

$$X \leq \frac{S_{it}}{m_i} \quad \forall i, t, \tag{4.14}$$

where S_{it}—defined in Equation 4.4—is the residual stock level of i at time t, and m_i is the *optimization capability metric*.[9] This quantity, X, is the new objective function, replacing what in Equation 4.1 was the objective function from the optimization to minimize cost.

Like the objective function, there is a range of reasonable choices for the optimization capability metric, m_i. The optimization capability metric determines the relative value of each asset compared to one another. One of the capability metrics discussed in Chapter Three, such as the resources needed for a particular operation from the Defense Planning Scenarios, could be used. Although practical in some contexts, these metrics would preferentially weight the relative mix of resources according to the chosen operation, which is insufficiently general.

We prefer a metric that reflects the requirements specified by plans over the entire planning horizon (FYDP). Obvious choices are the maximum demand for each resource over all time and the average demand for each resource over all time. The total demand over time differs from the average demand only by a constant and hence is math-

---

8 A fruitful avenue of research could be a two-step optimization to eliminate these issues in future analyses.

9 Note that, unlike the previous optimization, S_{it} can be positive or negative, depending on the actual asset level relative to demand.

ematically indistinguishable from the average as an objective function. The maximum and the average demands emphasize different shortfalls of capability.

The effects of the optimization capability metric choice depend on the choice of objective function, so selection of these two should emphasize consistent priorities. Since we are using an objective function in which decisions are driven largely by the global minimum in capability (usually at the time of maximum demand), differences in the weighting of the metric at this juncture are the most important, and the metric should prioritize large, acute shortfalls at times of high demand over small, chronic shortfalls. The maximum demand over time does this more so than the average, so we use this optimization capability metric as our default.

The constraints of the capability optimization are the same as those of the optimization for minimizing the net-present value, with the exceptions that the budget-smoothing constraint (see Equation 4.3) and the constraint that all requirements be met are both relaxed. In place of the budget-smoothing constraint, we use the constraint

$$B_{t^*} \leq b_{t^*} \quad \forall t, \tag{4.15}$$

where b_{t^*} is the permissible budget at time t^*. The permissible budget, b_{t^*}, can be specified in two ways: A budget for the first year can be set, plus a float parameter akin to Equation 4.3 that constrains the budget to fluctuate no more than a certain percentage from year to year, or it can be fixed for each year by the user.

Robust Programming

The two algorithms described above create programs that minimize costs or maximize capabilities across resources for a single-scenario set. It is also possible, using the same general approach, to construct a program that maximizes capability over *multiple-scenario sets*. Because this third algorithm creates a program that is robust to uncertain futures by optimizing across rather than within scenarios, we call this *robust optimization*. This robust optimization seeks to answer the following question: Given fiscal constraints, how should spending be distributed over programs to achieve the maximum, balanced capabilities across these programs under a number of possible futures?

Whereas the single-scenario set capability optimization maximized the global minimum for a single-scenario set, we now try to maximize all global minima. While the two single-scenario set models share many constraints, the robust model uses only the constraint (see Equation 4.11) that requires purchases to be positive, the constraint (see Equation 4.15) that constrains budgets, and the equations and inequalities stated in Equations 4.16 through 4.20. Some notation in the robust model is the same as declared in Table 4.1; new notation is listed in Table 4.2. Stated mathematically, we specify a budget similar to the single-scenario optimization that maximizes capability and maximizes the weighted sum of all global minima:

$$Z = \sum_{\eta=1}^{n} X_{\eta} w_{\eta}, \tag{4.16}$$

where η is the scenario index, n is the total number of scenarios, Z is the robust capability score, and w_{η} is the weighting assigned to scenario η.[10]

The global minimum for each scenario, X_{η}, is the maximum shortfall or minimum amount of remaining stock of each asset divided by the optimization capability metric of that scenario,[11] or

$$X_{\eta} \leq \frac{\left(\bar{S}_{it} - \bar{d}_{\eta it}\right)}{m_{i\eta}} \quad \forall i, t, \eta. \tag{4.17}$$

One could also choose to maximize the global minima of X_{η} over all η. We have chosen to maximize the weighted sum (Equation 4.16) instead because it will yield better performance across a range

[10] This weighting function allows the programmer to see the implications of favoring one or more scenarios over others.

[11] When all scenarios are weighted equally, the values of capability metrics are considered equivalent. For example, a value of 0.2 against one scenario is equivalent to 0.2 of any other.

Table 4.2
Additional Notation for Robust Algorithm

Symbol	Meaning	Units
a_{it}	Average cumulative redeployments for asset i over time t: $\frac{1}{n}\sum_{\eta=1}^{n}\sum_{\tau=1}^{t} d^{-}_{\eta i \tau} = a_{it}$	
X_{η}	distance function	
$d^{+}_{\eta it}$	amount of i deploying at t in scenario η	
$\overline{d}_{\eta it}$	cumulative net demand for i at t: $\sum_{\tau=1}^{t} d^{+}_{\eta i \tau} - d^{-}_{\eta i\left(\tau - l^{R}_{i}\right)}\left(1-k^{R}_{i}\right) = \overline{d}_{\eta it}$	
$d^{-}_{\eta it}$	amount of i redeploying at t in scenario η	
h	length of planning horizon	months
η	scenario index	
$m_{i\eta}$	metric value of i in scenario η	
n	number of scenarios	
$\overline{S}_{it}$	cumulative stock of i at t	
Z	robust capability score	
w_{η}	weight for scenario η (likely = 1)	

of scenarios if those scenarios have drastically different demands. An example will help clarify. Consider a case in which one very demanding scenario has a deep minimum of –2. All other scenarios have shallow minima, between –0.1 and –0.3. If the algorithm maximized these minima, it would allocate all resources to the one scenario with the deep minimum and improve it to, say, –1.6, leaving all the other scenarios with unchanged shortfalls. The formulation in Equation 4.16 would allocate resources to bring this worst case to, say, –1.9 and bring all other minima to zero. The idea is that it is preferable to solve a wide range of shortfalls for a number of scenarios than bring about a marginal improvement to one that cannot be solved, thereby leaving ones that can be solved unimproved.

Given that we are maximizing the sum of global minima across all scenarios, the algorithm could increase the capability in the least demanding scenario to a large surplus while leaving other, more demanding scenarios with a capability shortfall—an efficient way to improve the metric. We would prefer that the model select a policy that would meet all possible futures without shortfalls rather than produce shortfalls in some futures and residual capability in others. To avoid this case, we ignore the contribution to the objective function of any scenarios in which the optimization capability metric is positive by using the constraint

$$X_\eta \leq 0 \quad \forall \eta. \tag{4.18}$$

As a result, the objective function will focus solely on shortages and attempt to minimize the average shortage across the scenarios. For analyses in which demand for assets in all futures can be met without shortfall under programmed budgets, the constraint in Equation 4.18 should be relaxed so that the model searches for programs that will maximize robust residual capability.

The budget,

$$B_t \geq \sum_i \left\{ c_i P_{it} + o_i \left(\bar{S}_{it} - k_i^r a_{it} \right) + \frac{r_i a_{ih} \left(1 - k_i^r \right)}{h} \right\} \quad \forall t, \tag{4.19}$$

for each time step is the sum of the procurement, O&M, and average reconstitution costs for all items. The reconstitution is averaged for the following reason: In this optimization, the algorithm is creating a single program (a scheme for spending budget dollars) that will be applied to multiple scenarios. Because reconstitution is a function of redeployments, and redeployments a function of scenario-specific demands, a reconstitution scheme for one scenario might not be feasible for another.[12] To handle this, unlike the single-scenario set cases, we assumed that all redeploying assets in all scenarios would be sent to reconstitution immediately. Because it is frequently infeasible to fit all the reconstitution spending from a particular redeployment into a month's budget (spikes in redeployment cause spikes in reconstitution), we then averaged the reconstitution cost across scenarios and decremented the budget by this amount.[13] In this way, we included reconstitution costs but could still compare programs on a level playing field.

The cumulative stock level, $\bar{S}_{it}$, is set by

$$\bar{S}_{it} = \bar{S}_{i(t-1)} + P_{i\left(t-l_i^P\right)} \quad \forall i, t > 1. \tag{4.20}$$

As in previous models, $\bar{S}_{i1}$ is given by the user-specified initial stock level.

---

[12] For example, where one scenario had a redeployment of asset *i* at time *t*, another scenario might not have a redeployment there, so any plan to reconstitute the asset from the first scenario would not have a "broken carcass" to reconstitute in the second scenario.

[13] While each scenario is different, the steady-state component, the Baseline Security Posture, underlies all scenarios and is the major driver of reconstitution. Because of this, we feel that taking the average of these components across scenarios is reasonable.

CHAPTER FIVE

Applications to Policy Analysis

> To apply a rule to the letter, rigidly, unquestioningly, in cases where it fits and in cases where it does not fit, is pedantry. Some pedants are quite successful; they understood their rule, at least in the beginning (before they became pedants), and chose a good one that fits in many cases and fails only occasionally. To apply a rule with natural ease, with judgment, noticing the cases where it fits, and without ever letting the words of the rule obscure the purpose of the action or the opportunities of the situation, is mastery.
>
> —*George Polya, 1957, p. 148*

George Polya (1887—1985), one of the 20th century's most accomplished mathematicians, wrote the above quote in reference to general principles that he conjectured for solving problems in mathematics. The spirit of his advice is equally applicable to the use of any model of capabilities analysis in Air Force programming. Programming decisions are necessarily based on a range of factors that go beyond formulaic rules, including sensitivity to political concerns. Together, these considerations require the judgment of a programmer.

Yet the programmer needs insights into the impact of programming decisions on Air Force capabilities, in forming the Air Force POM, assessing the risks it might incur, and defending it to OSD and Congress. The more analytical and reproducible this process of capability assessment, the more easily it can be implemented into programming and the more useful it would be during the often short time available for building and defending the POM. What is needed is a balance of objective capability assessments and subjective considerations. In this final chapter, we show some examples of how a programmer would glean insights from the analysis presented in this monograph and conclude with some overall observations and recommendations.

Insights into Programming Policy

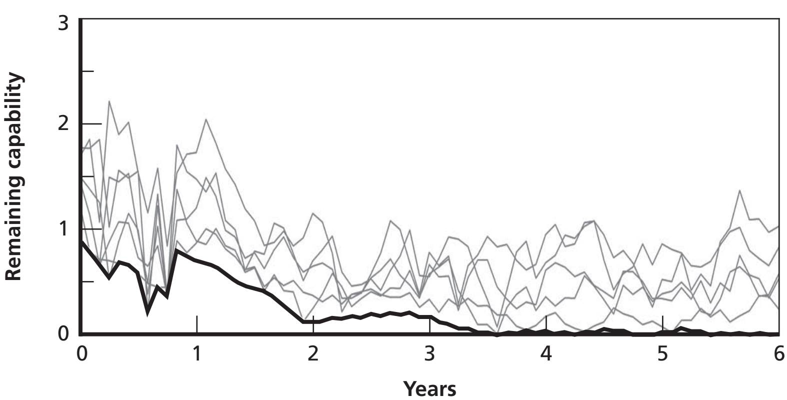
This section provides some examples of how the methodology described in this monograph can be used to guide programming decisions. The results presented here are all notional due to the sensitivity of real capability assessments.

Single-Scenario Set Cases

First, we discuss some of the utilities of the algorithms for a single-scenario set: (1) to minimize costs (with respect to a single-scenario set) and (2) to maximize capabilities as defined by a single-scenario set, subject to fiscal constraints. We begin with minimizing costs.

Figure 5.1 depicts capabilities from optimal programming of a related set of resources as a function of time. These resources may span one program element or several. This notional programming meets

Figure 5.1
Notional Optimization to Minimize Cost



RAND *MG815-5.1*

all the requirements of a single-scenario set at minimal cost subject to the constraint that spending not vary by more than a certain percentage from one year to the next. The ordinate of the plot shows the capability remaining over time during the FYDP beyond that needed to perform the scenario set specified by the plans. Hence, when a curve is at zero, that resource at that time exactly meets the requirements in the planning scenario set. If it is positive, it has more capability than needed for the scenario set. Because this optimization always meets any requirements, the curves must be non-negative. If a curve were negative, as we will encounter later, it would reflect a shortfall of that resource with respect to the scenario set.

The magnitude of the values on the ordinate depend on the choice of metric. Any capability metric can be selected as such a metric. This metric might be the remaining capability relative to a particular MCO, small-scale contingency, humanitarian relief operation, or any other contingency for which deployment requirements are known or can be determined. Note that the choice of metric will change only the *magnitudes* of the remaining capabilities. Whether the curves are in the positive (or negative) portions of the plot is independent of the choice of metric. Examining a range of metrics permits the programmer to see the quantitative impact of the proposed program relative to different types of contingencies.

For a related set of resources in Figure 5.1, the limiting resource forms the lower bound. For that set of resources, the aggregate capability is no better than the worst-performing element, so the overall capability of the resource set is given by the bold curve that marks the lower bound. When this bold curve is above zero, more of the resource is available than is needed for the specified scenario set *at that time*. A positive value does not necessarily imply an excess capability; positive remaining capabilities are sometimes needed to ensure that there are no shortfalls in the future.

Plots such as the one presented in Figure 5.1 show which resources are in excess relative to the scenario set (always possessing positive remaining capability) and which are critical (touch zero at some point). The underlying data, available to the programmer, indicate the balance of investments necessary for procurement, reconstitution, and O&M.

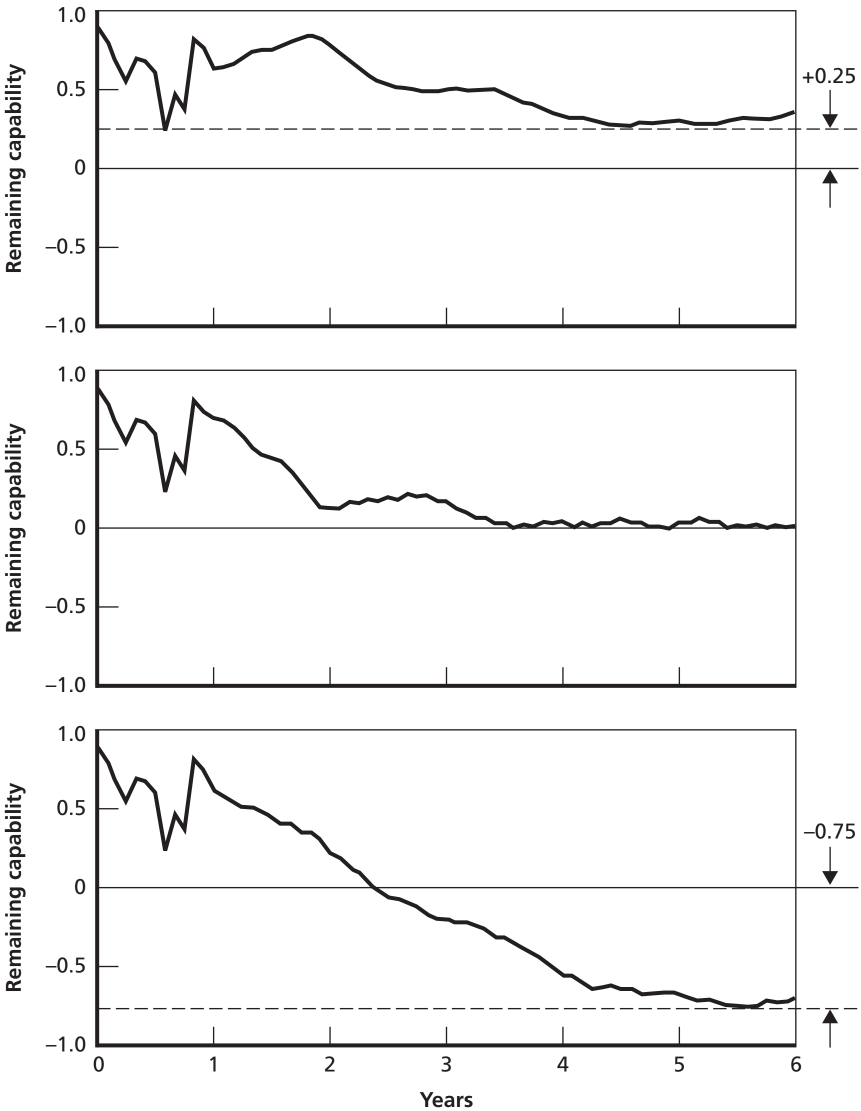
This information helps the programmer determine and defend not only the appropriate authorized asset levels, but also the sustainment dollars to ensure that those assets are real capabilities (mission capable) and not sitting, unavailable, due to lack of support.

This analysis can be extended to the case of maximizing capability relative to the scenario set, one example of which is shown in Figure 5.2. The center panel of the figure is the same plot as in Figure 5.1, except that the curves for each individual asset are suppressed—only the lower bounding curve is shown. The point of this analysis is to explore what risks would be accepted by spending less than the optimal values shown in the center plot, as well as to determine the additional capabilities acquired by spending more.

It is instructive to examine the case of adding and removing the *same amount* of money relative to the optimal solution shown in the center panel of Figure 5.2. The upper panel indicates the optimal programming solution if some additional money, say $10 million per year, is added relative to the program in the center panel. The lower panel shows the optimal programming if money is removed, say the same $10 million per year, relative to the program shown in the center panel. In general, the result will be as indicated in the figure: The same amount of money added to a program buys less additional capability than removing that money assumes in risk.

The reason for this nonlinear response is that the lower bounding, thick curve in the figure determines the overall capability of a set of related resources. If a program is out of balance (i.e., the remaining capabilities of individual resources are scattered widely above the lower bounding curve), buying additional capability is relatively cheap because only one or two resources might need to be purchased (or reconstituted) to push up the lower bounding curve. As more and more capability is acquired, the lower bounding curve moves upward, and more and more resources cluster at or near that lower boundary. Overall, the program is more balanced, which is good, but pushing the curve up further requires buying some of nearly all the resources and hence becomes more expensive. Said another way, in a healthy, balanced program, increasing the capability requires buying some of a lot of resources, since the resources are interdependent. However, under-funding in just one critical resource can render a whole resource set ineffective.

Figure 5.2
Notional Optimization to Maximize Capability

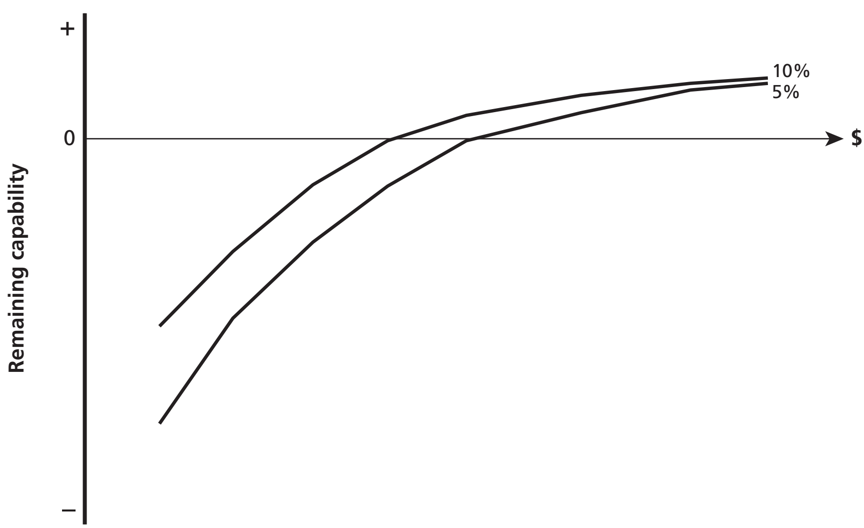


RAND *MG815-5.2*

Figure 5.2 and the underlying analysis used to generate such plots not only identify critical assets and the interdependence of resources as capabilities, but also illuminate the status of a program. If a program currently looks like the one depicted in the upper panel, it is an excellent candidate to find an offset—taking some money out of the program assumes little risk. If the program looks like the one depicted in the lower panel, it is an excellent candidate for an infusion of money, given that a small infusion of money can buy a lot of additional capability.

Figure 5.2 indicates the additional capabilities bought or risks assumed by three alternative budgets over the FYDP: one at the optimal level, one above, and one below. A full range of alternative budgets can be easily explored, yielding a curve that relates costs with capabilities. Two such curves are shown in Figure 5.3. The abscissa is the amount of money spent on the program, and the ordinate is the remaining capability as defined by the minimum of the lower bounding curve (like those in Figure 5.2) over time.

Figure 5.3
Notional Cost-Capability Curves



RAND *MG815-5.3*

Two curves are shown, each with a different constraint on the percentage change in funding from one year to the next. One of the curves is for a 5-percent constraint; the other is for a 10-percent constraint. The curve for the looser, 10-percent constraint is always above that of the tighter, 5-percent constraint. The numbers are not as important as the general principle: The more this smoothing constraint is relaxed, the more capability that can be purchased and sustained *at any funding level*.

Figure 5.3 also depicts another important point, generalized from Figure 5.2. In general, adding a dollar beyond the optimal level buys less additional capability than taking away a dollar assumes in risk (unrealized capability). This nonlinearity can be seen in both curves in Figure 5.3. The slope below the abscissa is steeper than the slope above it.

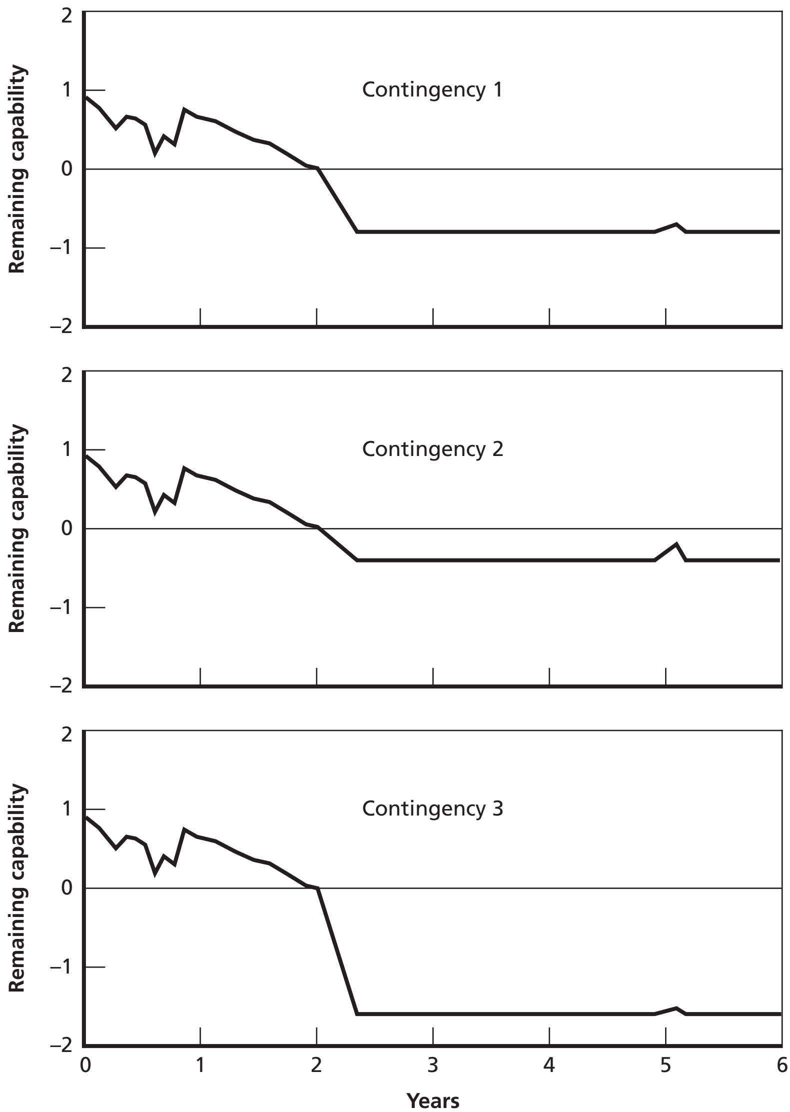
For a final example of insights, we turn to another way to assess the risk of a POM. Figure 5.4 shows the risk assumed by the occurrence of a contingency above and beyond that in the scenario set specified in the plans used to build the program. In this case, it is assumed that the program was constructed to meet all requirements in a scenario set at minimum cost. At year two in the FYDP, a contingency occurs in addition to the scenario set in the plans and, hence, a shortfall arises for at least one resource.

The advantages to the programmer are twofold: It is possible to both see the magnitude of the shortfall for a range of contingencies relative to a range of metrics and drill down to the necessary level of detail to determine which resources cause the shortfall. If a subset of resources repeatedly falls short, it is an excellent candidate for additional programming dollars if any are available.

Robust Programming

In the robust programming optimization, the goal is to maximize capability across a range of scenario sets. Whereas the single-scenario set optimization cases discussed in the previous section are quite useful for assessing capabilities and risk in a POM, we advocate robust programming over single-scenario set optimization for building a POM. Table 5.1 shows the power of a robust optimization. Although these data are again notional, the general principles and trends reflect those of reality.

Figure 5.4
Assessing Risk for Contingencies Beyond Those in Planning Objectives



RAND *MG815-5.4*

Table 5.1
Notional Robust Optimization

Scenario Set	POM Strategy					Robust
	1	2	3	4	5	
1	0.10	0.05	–0.26	0.01	–0.04	0.02
2	–0.10	0.02	–0.18	–0.42	–0.16	0.01
3	–0.50	–0.55	–0.43	–0.62	–0.56	–0.49
4	–0.36	–0.42	–0.55	–0.27	–0.33	–0.29
5	0.05	0.02	0.06	0.04	0.13	0.03

NOTE: The budget for each strategy is $100 million.

In the table, the columns represent various programming strategies in the POMs, developed to satisfy differing planning objectives. Each program is allocated the same budget to spend but buys and sustains a different mix of resources, depending on its objectives. Each row depicts a possible scenario set that might unfold in the future. Entries in the matrix indicate how well a programming strategy (column) scores against a possible future (row). The values capture the worst-performing time (capability minimum) during the FYDP. Positive values indicate that all requirements are met at all times, with some extra capability in reserve. Negative values indicate a shortfall at some juncture during the FYDP.

Consider first the POMs (columns), labeled 1 through 5. These are single-scenario set optimizations against planning scenario sets (rows) 1 through 5. In other words, the column for POM 1 shows an optimization to maximize capabilities against scenario set 1 with the spending constraint given. Column 2 is a POM that optimizes to maximize capabilities against scenario set 2 with the same spending constraint, and so forth.

There are some general characteristics of these single-scenario optimizations. Each of the POMs, 1 through 5, optimizes with the same budget. Hence, the best performer for each scenario set (1 through 5) must be the POM that is optimized in each row. That is, the best of the five POMs for scenario set 2 must be POM 2, and so forth. These values run down the main diagonal of the matrix.

Note that some of the POMs perform satisfactorily against their intended scenario set (e.g., POMs 1, 2, and 5), and others less so (POMs 3 and 4). This difference is because each of the scenario sets is not equally challenging for the resources available. Scenario set 3 is the most challenging, and the money appropriated is insufficient to meet all requirements of that scenario set. It is tempting to suppose that if a POM is built to deal with the worst-case scenario (scenario set 3, in this example), it will prepare better for other, less challenging scenario sets than would a POM that was built to address a less challenging scenario set. This is not the case in general.

The reason is that the proportions of resources needed for each scenario set are generally different. The most demanding scenario is the one that requires the greatest number of assets, but that scenario might require a different mix of resources than other, less demanding scenarios. A common instance would be scenarios in different theaters or against different adversaries. An example from agile combat support might be the fuels equipment needed to support air operations. It would be a very demanding case to support helicopter operations out of numerous austere bases. Such operations would require a lot of fuel bladders, pumps, and C-300 trucks. Being able to support such a scenario does not ensure the ability to operate KC-135 tankers out of more established bases.

Other examples abound. Fighter pilot training during the Cold War provides an interesting, broader example of this phenomenon. An underlying assumption in Cold War planning was that, if the United States could prevail over the Soviet Union—the most demanding adversary at the time—it was adequately prepared to meet lesser adversaries. Focusing on fighting the Soviet Union placed a heavy emphasis on the nuclear mission. A consequence was insufficient training for conventional air-to-air combat, a deficiency that was revealed during the Vietnam War.

This deficiency was rectified only by increasing the priority of training in conventional air-to-air tactics. During the period from 1965 to 1968, the Navy had a kill ratio of 2.4 to 1; after starting the

U.S. Navy Fighter Weapons (or Top Gun) School, the ratio climbed to 13 to 1 between 1970 and 1973 (Lambeth, 2000, p. 48). Preparing for the worst case did not equate to preparing well across all possible cases (Lambeth, 2000, pp. 36, 59–69).

Rather than plan for the supposed worst case and use the resources optimal for that case to meet all other cases, we advocate robust programming. Robust optimization, outlined in Chapter Four, is represented by the "Robust" column in Table 5.1. This optimization seeks to maximize capability against the full range of scenario sets, 1 through 5. Note that, for any of the possible futures considered, the robust solution outperforms all the single-scenario set optimizations except for the POM that builds specifically for that scenario. Yet, unlike the single-scenario set optimizations, when it has a positive remaining capability, this excess is not dramatically high. This is the key to how it achieves a robust solution: It avoids as much as possible an excess capability beyond what is needed for any of the scenario sets, instead applying those resources to mitigating shortfalls in other possible futures.

Conclusions and Recommendations

Programming tools, such as those described in this monograph, provide a guide, not a solution, to programming dilemmas. Uncertainty in many of the input factors and incommensurables—notably, risk—requires intervention by decisionmakers. But subjective decisions alone are insufficient to build a program that spends money and allocates manpower effectively and efficiently. The methodological programming approach and the tools developed in this research provide reproducible, quantitative guides for how to build a program to achieve specified capabilities and how to evaluate how well a program performs against various future security environments.

Three key elements make this analysis possible. The first is defining how to measure capabilities in a way that facilitates programming decisions. To guide programming, capability measures need to have several attributes. In some clear, reproducible manner, the capability measure must be related to program elements or clearly definable subsets of program elements. Second, the capability measures need to relate to planning objectives so that plans directly define and shape programming. And third, the capability measures must articulate capability in general terms that apply across programs, not specific or idiosyncratic terms that apply to one program or function. Otherwise, trading among capabilities and programs is neither reproducible nor quantifiable.

Current Air Force capability metrics fail, in general, to capture these attributes, which point toward using aggregated measures of how a resource provides a marginal contribution to operational-level objectives, such as the MCOs, small-scale contingencies, and steady-state deployments that constitute the Defense Planning Scenarios. Defining capabilities in this way naturally ties capabilities to plans.

Our first recommendation is, when feasible, to define capabilities in terms of OSD-level plans rather than Air Force tasks.

Tying capabilities to programs leads to the next key element: to determine the resource sets needed to provide those operational-level capabilities. For agile combat support resources, deployment requirements can be resolved at the air-order-of-battle level. What is needed to do these calculations rapidly and repeatably is a rule set for UTCs: How many of each UTC is needed, and what is the interdependence of the UTCs, to support specific numbers of given aircraft types, flying at given sortie rates, and flying out of locations with given infrastructures? This and prior RAND research has demonstrated the feasibility of such a rules-based tool with a prototype model (see Snyder and Mills, 2004) and argued that the returns on such a tool would extend far beyond programming (see Snyder and Mills, 2006). To be useful in regular programming and execution decisions, this model needs to be formally vetted, implemented, and periodically maintained.

Our second recommendation is to develop and maintain a rules-based tool for generating a requirements TPFDD given air order of battle–level inputs for planning scenarios.

These first two elements ensure that the ingredients exist to build cost-capability curves for sets of related resources, which is the foundation for the third key element: a set of algorithms to (1) assess the impacts of capability trades and (2) develop a robust POM in the face of uncertain future security environments. A set of resources does not represent a capability unless sufficient sustainment efforts maintain those resources in a mission-capable state. The primary contribution of the algorithms is to ensure the appropriate balance of investments in procurement and sustainment so that maximal capabilities can be realized. The algorithms must balance mission-capable resources within a POM so as to be available to meet a range of possible futures.

Our third recommendation is to develop a robust program across a range of plausible scenario sets that balances asset levels with sustainment investments, in lieu of programming to meet a single challenging scenario set.

Uncertainty abounds in programming. Input data, such as life expectancy of resources, potential obsolescence, and when modernization might be most expedient, are all very difficult to glean. Further, how the future may unfold is impossible to forecast. It is tempting to avoid modeling in the face of these uncertainties because the modeler must commit to decisions about the values of these parameters. However, any programming strategy makes assumptions about these inputs. Assumptions made without analysis are simply implicit and less reproducible. Analysis provides reproducible, quantifiable justification for the budget in terms of national-level objectives. A combination of astute analysis and perspicacious judgment can build a programming strategy to ensure a robust and nimble set of capabilities that meet the challenges of uncertain future security environments.

Bibliography

Air Force Instruction 10-2801, Air Force Concept of Operations Development, October 24, 2005. As of October 3, 2008:
http://www.e-publishing.af.mil/shared/media/epubs/AFI10-2801.pdf

Air Force Instruction 10-604, Capabilities-Based Planning, May 10, 2006. As of October 3, 2008:
http://www.e-publishing.af.mil/shared/media/epubs/AFI10-604.pdf

Chair of the Joint Chiefs of Staff Instruction 3170.01C, Joint Capabilities Integration and Development System, June 24, 2003.

DoD—*see* U.S. Department of Defense.

Kanter, Arnold, *Defense Politics: A Budgetary Perspective*, Chicago, Ill.: University of Chicago Press, 1979.

Korb, Lawrence J., "The Budget Process in the Department of Defense, 1947–77: The Strengths and Weaknesses of Three Systems," *Public Administration Review*, Vol. 37, No. 4, July–August 1977, pp. 334–346.

Lambeth, Benjamin S., *The Transformation of American Air Power*, Ithaca, N.Y.: Cornell University Press, 2000.

Lederman, Gordon, *Reorganizing the Joint Chiefs of Staff: The Goldwater-Nichols Act of 1986*, Westport, Conn.: Greenwood Press, 1999.

Office of Management and Budget, "Discount Rates for Cost-Effectiveness, Lease Purchase, and Related Assets," in *Guidelines and Discount Rates for Benefit-Cost Analysis of Federal Programs*, Washington, D.C., Office of Management and Budget Circular A-94, January 2008. As of October 3, 2008:
http://www.whitehouse.gov/omb/circulars/a094/a94_appx-c.html

Polya, G., *How to Solve It: A New Aspect of Mathematical Method*, 2nd ed., Princeton, N.J.: Princeton University Press, 1957.

Secretary of the Air Force, *Guide to Bare Base Development*, Air Force Handbook 10-222, Vol. 1, February 1, 2006. As of October 3, 2008:
http://www.e-publishing.af.mil/shared/media/epubs/afh10-222v1.pdf

Snyder, Don, and Patrick Mills, *Supporting Air and Space Expeditionary Forces: A Methodology for Determining Air Force Deployment Requirements*, Santa Monica, Calif.: RAND Corporation, MG-176-AF, 2004. As of October 3, 2008: http://www.rand.org/pubs/monographs/MG176/

———, "Air Force Deployments: Estimating the Requirements," *Air Force Journal of Logistics*, Vol. 30, No. 2, Summer 2006, pp. 4–9. As of October 3, 2008: http://www.aflma.hq.af.mil/lgj/Vol%2030%20No%202%20Main%20WWW.pdf

Stevenson, Charles A., *SECDEF: The Nearly Impossible Job of Secretary of Defense*, Washington, D.C.: Potomac Books, 2006.

U.S. Department of Defense, *Quadrennial Defense Review Report*, Washington, D.C.: Office of the Secretary of Defense, September 30, 2001. As of October 3, 2008: http://www.defenselink.mil/pubs/pdfs/qdr2001.pdf

U.S. Department of Defense Instruction 8260.01, Support for Strategic Analysis, January 11, 2007.