BRAC 2005: Analysis Handbook (Rev. 1.01)[1]

DoN Infrastructure and Analysis Team (DON IAT)

June 9, 2005

[1]Deliberative document. For discussion purposes only. Do not release under FOIA

Contents

Chapter 1

Introduction

This handbook describes the analytical tasks associated with the four stages of the BRAC analyses to be used by the DoN and the JCSGs as envisioned for BRAC 2005. We begin with a brief discussion of the analytical processes employed in prior BRAC rounds. We will review the analytical approach used in the previous two rounds of BRAC and then discuss why the methodology was changed for BRAC 2005.

1.1 1993 and 1995 DoN BRAC analytical process

The DoN analytical process used in preceding BRACs paralleled the four phases that are described in this handbook. For each installation type, capacity analysis was used to determine whether excess capacity existed. If excess capacity existed in that type of installation, a military value data collection and analysis was conducted for all installations in that category. For each installation type where excess capacity existed, alternatives generated using an optimization methodology were examined in detail, resulting in a set of recommended actions (see [1] and [4]).

The optimization methodology used by the DoN for BRAC 95[1] was simple: minimize excess capacity while maintaining or improving the average military value of the retained installations. For example, suppose the original configuration consisted of the five bases shown in figure 1.1 with the indicated military values and capacities. If only 23 units of capacity are required, the optimal solution retains bases A and E. This solution has zero excess capacity and an average military value of 80. The methodol-

[1]Reference [7] documents the details of the approach employed by the DoN for BRAC 95

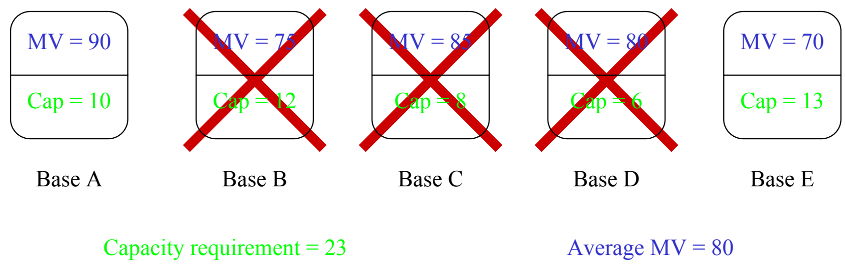

Figure 1.1: BRAC-95 optimization example

ogy also allowed us to generate the second-best and third-best solutions. In this example, the second-best solution retains bases A, C, and D having excess capacity equal to one and average military value equal to 85. Other constraints could be added to this methodology depending on the type of installations addressed, e.g., the number of berthing spaces needed on each coast was added to the optimization model for naval bases.

One important feature of the BRAC 93 and 95 analyses was that capacity was summarized by a single measure that approximated facility assets. For example, at an air station, the capacity was summarized as the number of squadrons that could be housed and supported, and at a naval station, capacity was summarized the the number of cruiser-equivalents that could be berthed. The use of these summary proxies simplified the analyses and proved adequate for capturing the essential features of the installations and environment. A different model was created for each type of installation addressed.[2]

1.2 Joint cross-service analyses in 1995

For BRAC 95, Joint Cross-Service Groups (JCSGs) were created to examine five common activities: depot maintenance, laboratories, test and evaluation, military medical treatment facilities, and undergraduate pilot training. The JCSGs conducted independent analyses to identify excess capacity. The JCSGs developed proposals for how the services could share assets and consolidate workloads, and thus more aggressively reduce the infrastructure. The JCSGs passed these alternatives to the military services for their consideration in developing their recommendations. However, the services were

[2]GAO reviews of the '93 and '95 BRAC rounds can be found in [2] and [5].

not required to accept the JCSG recommendations. GAO concluded that the efforts of the JCSGs met with limited success for a variety of reasons, including time constraints and the precedence of the services recommendations to those of the JCSGs. See [5] and [6] for a GAO review of the joint analyses conducted for BRAC 95. The DOD report to the BRAC Commission [3] also discusses the joint analyses.

1.3 If we are changing, why change?

Since the previous BRAC rounds, DoD and the Military Departments have changed the way that the DoD infrastructure is utilized, thus necessitating changes in the methodology for BRAC 2005. First, it is much harder to classify bases, installations, or activities into exclusive categories since bases, installations, and activities are much more likely to host or perform multiple functions. Many installations host both training and operational functions while many industrial activities produce widely different products.

Second, capacity for these different functions may not be reliably measured in a single dimension. For example, an industrial facility may be able to repair engines and airframes, but it is not possible to create a single-dimensional capacity measure that covers both product areas without introducing the artificiality of weighting the capacity for repairing engines relative to the capacity for repairing airframes. Capacity measurement has become a more critical issue because the reductions in capacity achieved in prior BRAC rounds make further reductions more difficult to achieve without employing a higher-resolution approach for measuring capacity. We have changed the methodology for these reasons. In addition, the new methodology has additional capabilities for generating alternatives that are described in chapter 4.

1.4 Definitions

A few distinctions must be highlighted to ensure clarity. An activity is the largest possible organizational unit used in the analyses. Typically, an activity will be a local command, such as a depot, hospital or military unit. It will normally be located at a single site, often as one activity of many at a base or installation. Functions are rational partitions of the activity's mission or responsibilities. They can be described in several ways. One way is to think of the functions as a partition into product lines–for example, airframe repairs and engine overhauls. Similarly, they can be thought of as

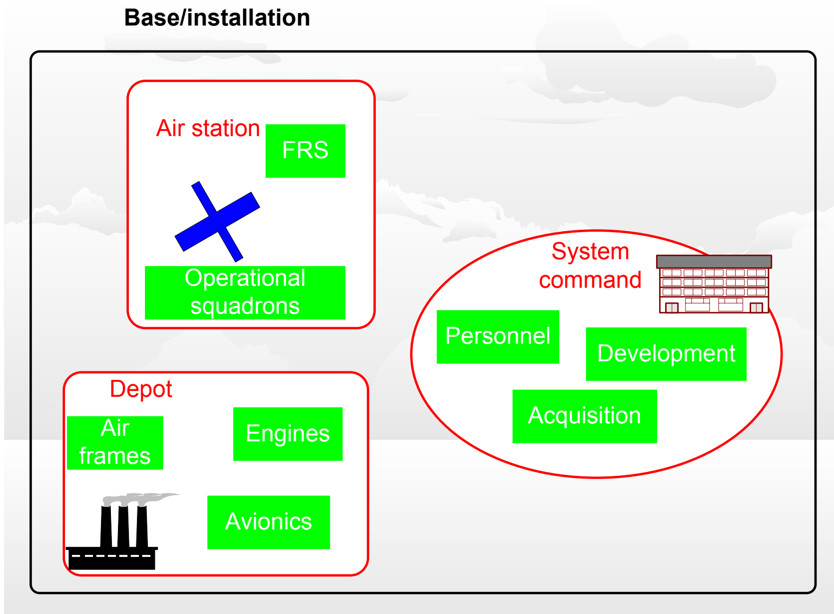

Figure 1.2: Illustration of definitions

outputs–for example, dental care and undergraduate pilot training. A third way to think of them is as subordinate organizations. For example, the laboratory and fabrication shop of a technical center are separate functions within the technical center. Because these are different ways of describing the concept of partitioning an activity, these examples naturally lead to parallel descriptions–the product line is associated with a specific output performed by a particular shop or subordinate organization. Figure 1.2 illustrates these definitions.

The function and activity views provide two different levels for performing the analysis. We can think about which activities should be retained, or we can think about how the functions should be assigned to different activities (and thus close an activity when it is assigned no functions). Later, we will find it useful to link functions with major physical resources.

1.5 Hypothetical examples

We will use an example of seven depots to illustrate the application of the methodologies described in this handbook to a typical JCSG problem. The basic data for the depots is given in appendix B. The basic data for a moderately complex example of the application of these methodologies to the DoN problem is given in appendix C.

1.6 Document organization

We first discuss capacity analyses. We then describe military value analysis, generating alternatives with an optimization model, and scenario analysis. The appendices include general guidelines for identifying the data needed for BRAC analyses, descriptions and data for the two main examples employed in the handbook, and a primer on the military value calculation. The mathematical description of the optimization model used to generate alternatives and the code used to actually implement the model are also contained in appendices. Code examples are given for a simple example and for the hypothetical examples identified above. We also describe the computer environment we created to perform the necessary modeling. A description of fuzzy sets, measuring value with fuzzy functions, and definitions of types of fuzzy functions that we employ in the military value analysis are included in the last two appendices.

Chapter 2

Capacity analysis

In general, we will think of capacity in terms of the availability of resources, such as land, air space, buildings, workshops, and technical centers, to support operational forces. Capacity analysis is the process used to gather information on the current base structure and its resources and compare current resources to the resources required to support the projected force structure. The difference between available resources and the required resources is a direct measure of excess capacity. The initial determination of capacity is a macro-level analysis. It provides a very simple determination of total excess capacity without regard to where excesses may be located. However, the data collected for the capacity analysis will also fill a more important role later in the BRAC process. Later steps of the BRAC process focus on developing options for reducing excess capacity and, thereby, freeing resources for other uses.

In this chapter we present an overview of the initial capacity analysis, with two examples that illustrate the determination of excess capacity. We then clarify some of the definitions and comment on ambiguities that have to be faced in determining capacity. Finally, we discuss the role that capacity data plays in later steps of the BRAC process.

2.1 Initial capacity analysis

For each category of activities, capacity measures are selected to reflect an appropriate set of *throughputs*. Throughput generally refers to a rate of production over some period of time. The units of throughput at an air base may be the number of squadrons that can be housed and supported. At a training center, the units of throughput could be the number of personnel trained in a fiscal year. Some categories of activities may have multiple throughputs. For example, the throughputs of an aviation depot could be man-years of effort in avionics, air-frame, and engine repair.

The initial capacity analysis determines the maximum level of throughputs capable of being produced with the current physical base structure. Like capacities are summed over each activity within a category and then compared to the capacity needed to support the overall throughput required to support the force structure. These throughput requirements derive, directly or indirectly, from the planned force structure. Excess capacity is deemed to exist for a category of activities when total current capacity of a particular type is greater than that required to support the throughput requirements.

The actual determination of capacity is based on a knowledgeable assessment of capabilities of the individual activities. Capacity will reflect the limited availability of one or more substantial physical resources that are most likely to present the bottleneck to production. The limiting resource is the one that will determine capacity. A substantial resource means one that is not easily expanded. Later in the BRAC process, there will be more explicit consideration of the individual resources and their potential expansion, reallocation, or disposal.

The initial capacity analysis is both a screening and an informational tool. In prior BRAC rounds, the capacity analysis was primarily a screening tool. Those categories of activities with no significant excess capacity were excluded from further analysis; categories with excess capacity were evaluated in depth for possible closure or realignment. In the current round of BRAC, the finding of no excess capacity is not sufficient to exclude facilities in that category from further analysis. Activities across all categories may be evaluated further to investigate the potential for consolidation or relocation. Consideration of activities across categories distinguishes this round of BRAC from previous BRAC rounds. The initial assessment of capacity highlights the potential magnitude of the opportunities for realignments and closures. The data collected will also be an essential element of the scenario generation step (Chapter 4).

2.2 Capacity analysis definitions and assumptions

In practice, the initial assessment of capacity can prove to be more complex than it might at first appear. There are a number of definitional issues that have to be resolved. What do we really mean when we say that capacity is the *maximum level* of throughput from the *current base structure*? The issues include questions such as:

- Whether skilled labor should considered in determining capacity,
- The meaning of surge requirements and surge capacity,
- When to consider planned or possible expansion of facilities.

These definitional issues are addressed below. In addition, we introduce some inherent challenges to measuring capacity that arise when there are several throughputs at a single activity that compete for the use of some key resources.

Normal capacity is a measure of potential throughput using current physical infrastructure resources, as distinct from input resources such as labor and materials, under normal (sustainable) working conditions. It should be assumed that the workforce and material needed to sustain throughput are available.

Surge capacity is the potential throughput if current physical resources are used as intensively as realistically possible. Surge capacity and requirements address the ability to provide sufficient operational support in the time between the initial identification of a need for increased throughput and the time when additional capacity can be created. It usually refers to using the current resources more intensely (e.g., increasing the staffing, working additional shifts and more days per week, running the equipment at higher speeds). Furthermore, surge might involve a usage level that cannot be sustained over a long period of time. The surge capacity should be determined by how realistically the throughput could be increased, given some assumptions on workforce, materials availability, and equipment maintenance. Specifically, it should be assumed that the workforce necessary to achieve normal capacity is already in place. The amount of additional labor that could be applied should be based on a realistic assessment of how much overtime that workforce can provide, and how readily available the required skills are in the short-term labor market. In addition, the intensity of usage of the physical capital should be based on a realistic assessment of how long that throughput rate can be sustained and how long it will take for additional physical capital to be in production. Furthermore, the time to expand the physical capital through rental, the reconstitution of any mothballed resources, and the construction of new facilities should be incorporated as part of the analysis. The materials required to meet the surge capacity should be assumed available because the focus of the analysis is on the physical plant's throughput capacity.

Excess capacity. Excess capacity could be evaluated in two ways:

1. the excess of normal capacity over normal requirements

2. the excess of surge capacity over requirements during surge periods

The minimum of these two values would be the relevant measure of excess capacity, i.e., the capacity that could be eliminated without impairing military readiness. It is not appropriate to evaluate excess capacity as the excess of normal capacity over surge requirements.

2.2.1 Workforce assumptions

There is often debate as to whether specialized workers should be considered, along with facilities, in determining normal capacity. The answer is no. BRAC analyses focus on facilities alone, and should incorporate a long-term perspective. Labor may be constrained in a short-term horizon, but, over time, can be expanded. Consider the error introduced by incorporating labor constraints in normal capacity. Suppose, for example, skilled mechanics use only half of the available physical capacity in aviation depots. If reported capacity were adjusted down to reflect the labor useage, the excess capacity in facilities would be obscured. That could lead to missing an opportunity to dispose of facilities and consolidating the specialized workforce in those that remain.

2.2.2 Current base structure and the potential for expansion

The stated intention of the initial capacity analysis is to assess the capacity of the current physical structure. We may face a number of ambiguities in making the assessment. What about construction or renovation already underway? The suggested practice is to consider these as complete. What about mothballed facilities or those in need of repair? It may be appropriate to consider these as if they were in operating condition to the extent that restoring the facilities does not require substantial time or expense. It is difficult to say exactly where the boundary between facilities that count and those that don't should lie. No other expansion of the primary plant is considered in initial capacity analysis. However, data on the potential for expansion and facility restoration should be collected for use in the later scenario generation analysis.

2.2.3 Multiple throughputs competing for resources

One of the most complex situations arises when several throughputs are competitors for key resources. For example, suppose the throughputs of both the engine repair and airframe repair shops depend upon the services of the same machine shop. How much of each throughput can be provided will depend upon how the services of the machine shop are allocated. There is no definitive answer for how much capacity exists for each throughput. We recommend calculating capacity assuming that access to such shared resources is allocated according to the normally planned throughput.[1] Later BRAC steps allow for a more careful analysis based on individual resources that will avoid the ambiguities that are inherent in trying to determine a simple measure of capacity.

2.3 Simplified capacity analysis examples

Now, we use two examples to illustrate some potential pitfalls of capacity analysis and a means of estimating excess capacity when critical resources must be shared by multiple functions. The first example demonstrates the importance of appropriately defining capacity to address important resource-level data and develop matching requirements data.

2.3.1 Capacity analysis for shipyards

This example will show how failing to collect and utilize resource-level data can lead to the wrong conclusions. In addition, the example will demonstrate how measuring capacity and requirements differently can lead to the wrong conclusions.

Drawing inappropriate conclusions may result in a poor set of alternatives and a significant waste of effort during the scenario analysis phase. In some cases, the alternatives generated may omit good feasible solutions. This will limit consideration to alternatives that might not be as attractive as those that should have been considered, and result in poor outcomes. In other cases, the alternatives generated may include infeasible solutions. These might appear attractive and be considered in the scenario analysis. Evaluating alternatives that are ultimately infeasible can be avoided by better data collection early in the process. Because of the extensive data required and limited time available, we try to examine only alternatives that

[1]Section 2.3.2 demonstrates how to make this calculation.

are very likely to be feasible. A reasonable amount of up-front effort in collecting the right data can have a major pay-off later.

Consider the ship overhaul work in an infrastructure consisting of three shipyards. Assume that the work in a shipyard can take place on the ship while it is in drydock, on the ship while it is moored to a pier, or on equipment removed from the ship and repaired in a shop. Suppose that the current annual workload is as shown in table 2.1 below. Work is measured in the number of direct labor hours (DLH) that are performed.

Site	Drydock	Pierside	Shop	Total
A	3,000	1,500	3,000	7,500
B	2,000	1,500	2,000	5,500
C	2,000	500	2,000	4,500
Total	7,000	3,500	7,000	17,500

Table 2.1: Current annual workload (thousands of direct labor hours)

Capacity is measured as the number of direct labor hours that could be performed annually at the facility, *assuming that the facility is fully utilized, that is, with sufficient labor that the equipment is not idle*. Suppose that the drydock, pierside, shop, and total capacities are as in table 2.2.

Site	Drydock	Pierside	Shop	Total
A	4,500	3,000	4,500	12,000
B	3,000	4,000	3,000	10,000
C	2,000	1,000	3,500	6,500
Total	9,500	8,000	11,000	28,500

Table 2.2: Capacity (thousands of direct labor hours/year)

Suppose the total annual work requirement, which includes work that can be done at the pier, is the same as the current workload, 17,500 DLH, distributed across drydocks, pierside, and shops as in table 2.1.

Non-comparable requirement and capacity data

First, we will examine how capacity data that uses a different definition of capacity than the requirements can result in an inaccurate assessment of viable alternatives. Suppose that the work requirement is to provide a total of 17,500 DLH per year. The allocation of the work requirements among drydock, pierside and shop work is not specified–just that the retained infrastructure accommodates a total of 17,500 DLH each year. However, assume that the capacity data call only collects drydock and shop capacity. Thus, the reported capacity is as shown in table 2.3. We are only allowed to use the collected data, which exclude pierside work.

Site	Drydock	Shop	Total reported
A	4,500	4,500	9,000
B	3,000	3,000	6,000
C	2,000	3,500	5,500
Total	9,500	11,000	20,500

Table 2.3: Reported capacity (thousands of direct labor hours/year)

First, we note that the actual, yearly workload appears feasible even with the underreported capacity: the total reported capacity (drydock and shop) is more than the current yearly requirement. Thus, the data will appear to be consistent, and we might not notice the error in reporting shipyard capacity.[2] However, the understated total capacity implies a greatly overstated utilization, suggesting that the shipyards are operating near capacity, when in reality, the difference between the total capacity (28,500 DLH/year) and requirement (17,500 DLH/year) is significant. Given the requirement and the reported capacities, there are no two sites that can handle the workload: Sites A and B have a combined reported capacity of 15,000, sites B and C have 11,500, and sites A and C have 14,500. Thus, the requirement, and an incomplete capacity measure, forces all three sites to be retained.

However, if some of the work can be done pierside, as in the current allocation, the total capacity enables two sites (site A and either one of sites B or C) to meet the demand. Thus, given incomplete capacity data, feasible alternatives are not even considered.

Omitting resource level constraints

Second, we will examine the importance of incorporating different resources into the capacity and requirements. Suppose that the problem above is resolved, and that the total capacities include the pierside, as well as shop and drydock capacity. Assume that breakdowns among the shipyard type

[2] This is not necessarily the case. If there were more work done pierside, and less unused drydock and shop capacity, then it would not even be feasible.

capacities were not reported, and are not incorporated into the requirement, even though they do matter. That is, the only data we have are the total requirement, and the total capacity by shipyard. However, assume also that the workload in the drydock must be performed in the drydock, and shop work must be performed in the shop.

Given the assumed information available–that is just aggregate capacity and requirements, we would consider three possible combinations as feasible: shipyards A, B and C together meet the requirement, as do A and B together, and A and C.

However, if we incorporate the requirements for resources by type, a different feasible set results. Because 7,000 DLH must be available in the drydocks, sites A and C together do not provide sufficient capacity.[3] Though the aggregate capacity of the two sites is more than is required, the capacity is not in the right places. There is insufficient drydock capacity to meet the requirement, though the shop and pierside capacity mask this deficiency. Inadequate data could lead to a scenario retaining inadequate facility capacity.

These examples demonstrate the importance of collecting capacity data that is comprehensive in the resources used, and that is designed to match the requirements that must be met.

2.3.2 Capacity analysis for a depot example: multiple resources and products

We use a depot example to demonstrate the calculation of excess resource capacity for a case having multiple shared resources used to produce multiple products. We use the data for seven fictional depots given in appendix B. There are four products in this example: airframes, tanks, turbines, and electronics. The requirement for these products is derived from the future force structure.

We assume a very simple force structure consisting of just aviation squadrons and tank battalions. Each aviation squadron generates airframe overhauls, turbine overhauls and electronics work. Each tank battalion generates tank overhauls, turbine overhauls and electronics work. The current and future force structure are given in table 2.4.

The product demands for squadrons and battalions are given in table 2.5. The demand for air frames and tanks are easily derived from the current workload and force structure. These imply that an air wing demands eight

[3] A and C also do not provide the necessary 3,000 DLH in shop capacity.

Force component	Current force structure	Future force structure
Air wings	4	5
Tank battalions	6	8

Table 2.4: Current and future force structures for depot example

Product	Current workload	Workload per wing	Workload per battalion	Future requirement
Air frames	32	8	0	40
Tanks	24	0	4	32
Turbines	230	20	25	300
Electronics	3500	500	250	4500

Table 2.5: Depot product demands

airframe overhauls per year, and a battalion demands four tank overhauls per year. The demands for turbines and electronics are not so clearly defined by the aggregate data because both tank battalions and air wings generate demands for the same products, though not equal. We assume that additional data or analysis reveals that the demands per air wing and tank battalion are as provided in table 2.5. Applying these demands to the future force structure results in the required capacity. The required capacity is a key input to the analysis.

There are four resource types needed in this production example: test ranges, fabrication shops, hangars, and test facilities. Hangars are needed to perform work on airframes. Fabrication shops are needed to perform work on airframes, tanks, and turbines. Test ranges are needed to test airframes and tanks. Test facilities are needed to complete work on all four commodities. The quantities of these resources available at each depot are given in table B.1. Production of a unit of each commodity type at a depot may require the utilization of each of the shared resources at the depot. The amount of each resource required to produce one unit of a product at a depot is given in table B.3.

We would like to measure the capacity of these seven depots to produce the four product lines so that we may compare the capacity to the requirement given in table 2.5. Unfortunately, because of the shared resources, there are many different mixes of the four products that could be produced at the seven depots. If we take the relative numbers of required production from table 2.5 as a measure of the value of each unit produced, we can impute the values for each type of production as shown in table 2.6.[4]

Product	Value
Air frames	112.50
Tanks	140.63
Turbines	15.00
Electronics	1.00

Table 2.6: Imputed product values

We can estimate total capacity across the seven depots by maximizing the value of the total production. We do that by solving the following optimization problem:

$$\text{maximize} \sum_{i \in I} v_i \left(\sum_{j \in J} x_{ij} \right)$$

subject to:

$$\sum_{j \in J} x_{ij} \geq R_i \qquad \forall i \in I$$

$$\sum_{i \in I} u_{ijk} x_{ij} \leq b_{jk} \qquad \forall k \in K, \forall j \in J$$

$$x_{ij} \geq 0 \qquad \forall i \in I, j \in J,$$

where I is the set of product lines, e.g., airframes, tanks, turbines, and electronics, J is the set of depots, and K is the set of resource types. The value of producing one unit of commodity type i is given by v_i (these are the values shown in table 2.6). The decision variable, x_{ij}, indicates how much of commodity type i is to be produced at depot j. The amount of resource type k required to produce one unit of commodity type i at depot j is given by u_{ijk} and the total amount of resource type k available at depot j is given by b_{jk}. We require the solution to meet or exceed the requirements given in table 2.5, where R_i is the requirement for commodity type i and we must not use more of the resources at a depot than are available.

Maximizing the total value of the production across all seven depots results in the production of each commodity type shown in table 2.7. These

[4]In doing this, we are assuming that the total requirement for each of the four products are equally valued. In the absence of a market mechanism to establish relative values, such an arbitrary approach may be appropriate.

results indicate that there is significant excess capacity in the seven depots relative to the requirement. There should be opportunities to close depots and realign the workload, thereby releasing resources for other uses.

Product	Maximum production
Air frames	59
Tanks	116
Turbines	466
Electronics	40,642

Table 2.7: Maximum production quantities (repairs/year)

2.4 The role of the capacity data call in later steps of the BRAC process

The initial capacity analysis is intended as a first-cut assessment. Chapter 4 takes a more detailed look, as part of the development of alternative closure and realignment scenarios, at methods for reducing excess capacity. Where the initial capacity analysis provides a system-wide assessment of excess capacity, the generation of alternative scenarios will explicitly consider the location of individual resources. Where the initial capacity analysis considers resources as fixed and tied to current use, the scenario generation analysis allows for the expansion and reallocation of resources.

The greater detail of the scenario generation analysis leads to data requirements that may not be immediately apparent in formulating the capacity analysis. Expandability, for example, is not relevant to the initial analysis of excess capacity, but needs to be considered later. Similarly, there are resources that are unlikely to be considered as relevant in the initial capacity analysis, but that can prove to be an important consideration when generating alternative scenarios. Any resources that are likely to limit the movement of functions between bases require some consideration. It is essential that the data collected allows for identification of resources by location, even though this may not seem essential to the calculation of excess capacity.

Chapter 3

Military value analysis

A military value must be calculated for each activity or for each activity/function combination, depending on the focus chosen. This section illustrates a method for calculating military value that is an extension of the method used by the DON for the 1995 BRAC process and that conforms to the OSD guidance for BRAC 2005. We use fictional data for two depot activities shown in table 3.1 to illustrate the method. The table shows the depot characteristics of interest, the units of measure used, and the measured values for each of the two activities.

3.1 Structure for calculating military value

Military value is calculated based on the military value criteria and the attributes associated with those criteria. For this example, we will assume there are four military value criteria: readiness (R), facilities (F), mobilization (M), and cost (C). For a particular activity type or activity/function assessment, decision makers will weight these criteria with positive values that sum to 100, as we have done in table 3.2. By assigning a weight of 50 to readiness, the decision makers want 50 points of the possible military value score for an activity or activity/function combination to account for readiness.

The four military value criteria will be further broken down into attributes. All of the attributes for one of the criteria will be given weights that sum to 100. The total weight for each criteria/attribute combination will be the product of the criteria weight and the attribute weight divided by 100 as shown in table 3.3. As with the criteria, these weights indicate the total points available within that criteria/attribute combination. For

Activity characteristics	Measure	Depot 1	Depot 2
Equipped machine shops	K ft^2	9	12
Equipped bench facilities	K ft^2	5	7
Foundry	# furnaces	4	6
Secure outdoor storage	K ft^2	100	100
Water	mg/d	3.5	3.0
Number of shipping and receiving docks		8	7
Annual maintenance budget	% PRV	1.5	2.3
Size of local mfg labor market	annual mfg revenue ($M)	450	350
Local labor skills	% with 5+ years of experience	21	11
Distance to nearest commercial air terminal	nm	25	20
Distance to nearest railhead	nm	15	20
Distance to nearest interstate highway	nm	10	3
Distance to nearest ocean trans dock	nm	600	50
Local crime rate	crimes/100K	4,684	5,002
Average commuting time	minutes	30	17

Table 3.1: Military value data example

example, the skills attribute associated with the readiness criteria is worth at most 10 points out of the 100 maximum possible score. Note that an attribute may appear with more than one military value criteria. We now have, in this example, nine military value criteria/attribute combinations. The military value questions that we use to measure meaningful characteristics may apply to any subset of these nine criteria/attribute combinations.

3.2 Calculating the weight to be given to a question or characteristic

The maximum score that an activity or activity/function combination may receive for a response to a question or characteristic is based on the pre-

Criteria	Weight
Readiness	50
Facilities	15
Mobilization	20
Cost	15
Total	100

Table 3.2: Military value criteria weights

Criteria	Attribute	Weight within criteria	Overall weight
Readiness	Equipment capability	50	25
	Distance	30	15
	Skills	20	10
Facilities	Equipment condition	60	9
	Security	40	6
Mobilization	Distance	55	11
	Skills	45	9
Cost	Quality of life	50	7.5
	Cost	50	7.5
		Total	100

Table 3.3: Attribute weights

viously discussed weights for military value criteria, their associated attributes, to which of these attributes the question or characteristic may apply, and the relative score given to the question or characteristic by the decision makers. This latter score, a number from 1 to 10, allows the decision makers to distinguish between questions or characteristics that may apply to the same criteria/attribute combinations, but which are inherently different in their importance to assessing military value in the view of the decision makers.

These data are combined according to the procedure we describe in appendix D to derive a weight for each question or characteristic. In general, a question or characteristic that is associated with more of the criteria/attribute combinations that have larger weights and given a higher relative score will have a higher weight.

3.3 Determining the value an activity receives for a question or characteristic

Each activity will have answered data call questions that provide responses corresponding to the questions or characteristics used to assess military value. Responses may be in the form of *Yes/No* answers or actual counts or measures. In the case of a *Yes/No* response, assuming that a *Yes* is the better answer, a *Yes* response would get a value of 1 while a *No* would get a 0. In the case of our depot example, Depot 1 says that it has 9 equipped machine shops while Depot 2 says that it has 12. Each of these responses must be converted to a value between 0 and 1 as well. There are many ways to do this, but we recommend the use of *fuzzy* functions which are described next.

3.3.1 Assigning values using fuzzy functions

There are no *Yes/No* or *1/0* values in the table since binary measures should only be used if the measure associated with the characteristic cannot be quantified or if the metric of interest is inherently discrete. Using a *Yes/No* or *1/0* measure is equivalent to saying the military value is a step function: less than or equal to a certain value gets a score of 1 while a greater value receives a 0. For example, the distance to a range could be a very important characteristic, but saying that a distance less than or equal to 500 miles should get a 1 while any distance greater than 500 miles should get a 0 is not a meaningful method to assess military value. Rather than use the step function, we propose using an approach from the field of fuzzy set theory to obtain a score. Suppose we believe that we are indifferent to any distance less than or equal to 100 miles to the range, i.e., we really cannot distinguish any inherent goodness in being 50 miles from the range versus 100 miles from the range. In addition, suppose we believe a distance of 900 or more miles to the range is totally unacceptable and that distances in between are better or worse depending on how close they are to the ideal of 100 miles. Fuzzy functions are an ideal way to address this situation. A tutorial on fuzzy functions is included in appendix J.

For example, the function on the left of figure 3.1 gives a score between 0 and 1, depending on the distance to the range. This fuzzy function gives a score of 1 to any distance between 0 and 100 miles and a score of 0 for any distance greater than 900 miles. Distances between 100 and 900 miles receive intermediate values in a smooth manner. This avoids the obviously unrealistic simplification of using the step function on the right of the figure

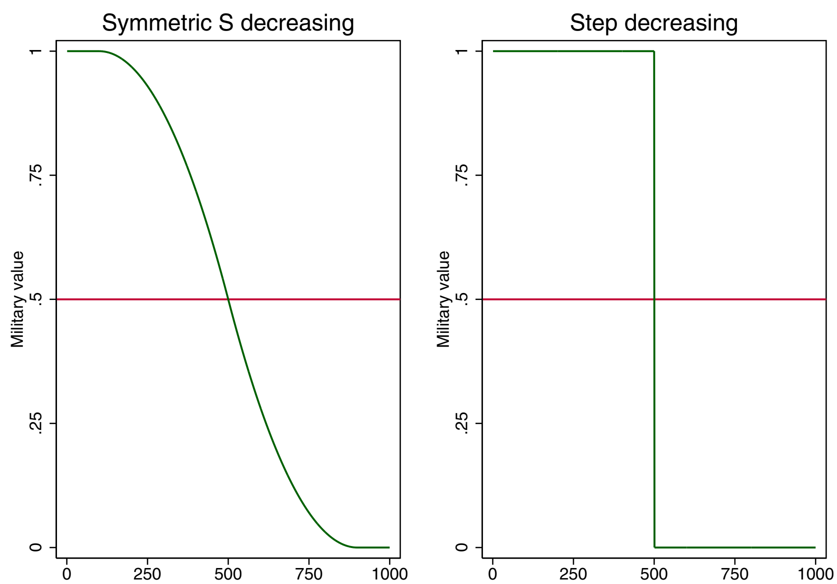

Figure 3.1: Fuzzy function example

of giving a score of 1 to an activity that is 499 miles from the range and a score of 0 for one that is 501 miles from a range.

In table 3.4, using characteristics from the depot activity case, we show how we specify the fuzzy function to be used in scoring the values for the two depots. The last five columns are used to specify the type of fuzzy function[1] and the parameters needed to shape the function. Under quality of life, a crime rate of 1,000 or fewer felonies per person-year will receive a 1 while a rate of more than 6,000 per person-year will receive a score of 0. Rates greater than 1,000 and less than 6,000 will receive intermediate scores.

[1]The fuzzy function types are specified in appendix K. The shape parameter corresponds to the numbering scheme used in appendix K to identify the different types of fuzzy functions.

		Function			
Activity characteristics	Measure	shape	min	med	max
Equipped machine shops	K ft^2	10	0	6	20
Equipped bench facilities	K ft^2	10	1	4	10
Foundry	# furnaces	10	0	3	6
Secure outdoor storage	K ft^2	10	5	60	120
Water	mg/d	7	2	2.5	3.0
Number of shipping and receiving docks	# docks	7	1	3	5
Annual maintenance budget	% PRV	10	1	3	4
Size of local mfg labor market	annual mfg revenue ($M)	10	200	300	500
Local labor skills	% with 5+ years of experience	7	5	25	50
Distance to nearest commercial air terminal	nm	5	10	25	60
Distance to nearest railhead	nm	5	10	25	75
Distance to nearest interstate highway	nm	5	5	12	18
Distance to nearest ocean trans dock	nm	5	20	30	100
Local crime rate	crimes/100K	5	1,000	3,500	6,000
Average commuting time	minutes	5	10	20	35

Table 3.4: Measures for activities and the parameters defining the fuzzy functions

3.4 Calculating military value

The table shown in figure 3.2 summarizes the calculations we have done to determine a military value for each depot. The first column identifies each of the characteristics used to assess the military value for a depot. The next nine columns show how we associated each characteristic with each of the nine criteria/attribue combinations. The *Equipped machine shops* characteristic, for example, is associated with just the *Equipment capability* attribute of the *readiness* criteria. The next column shows the relative weight score we gave to each characteristic. We decided that the *Foundry* characteristic was a more important characteristic than the *Equipped machine shops* characteristic since we gave the first a score of 10 and gave the second a score of

Criteria	**Readiness**			**Facility**		**Mobilize**		**Cost**					
Attribute	**Equipment capability**	**Distance**	**Skills**	**Equipment condition**	**Security**	**Distance**	**Skills**	**Quality of life**	**Cost**				
Criteria/attribute weight	**25.0**	**15.0**	**10.0**	**9.0**	**6.0**	**11.0**	**9.0**	**7.5**	**7.5**			**Miltary Value**	
Activity characteristics	**Characteristic applies to criteria/attribute**									**Score**	**Weight**	**Depot 1**	**Depot 2**
Equipped machine shops	1	0	0	0	0	0	0	0	0	7	6.034	0.691	0.837
Equipped bench facilities	1	0	0	0	0	0	0	0	0	7	6.034	0.653	0.875
Foundry	1	0	0	0	0	0	0	0	0	10	8.621	0.778	1.000
Secure outdoor storage	0	0	0	0	1	0	0	0	0	7	2.800	0.944	0.944
Water	0	0	0	1	0	0	0	1	1	6	9.865	1.000	1.000
Number of shipping and receiving docks	1	0	0	1	0	0	0	0	0	5	7.772	1.000	1.000
Annual maintenance budget	0	0	0	1	0	0	0	0	1	2	2.538	0.031	0.211
Size of local mfg labor market	0	0	1	0	0	0	1	0	1	5	12.385	0.969	0.719
Local labor skills	0	0	1	0	0	0	1	0	0	5	9.500	0.400	0.150
Distance to nearest commercial air trans terminal	0	1	0	0	0	1	0	0	0	7	6.067	0.500	0.778
Distance to nearest railhead	0	1	0	0	0	1	0	0	0	5	4.333	0.944	0.778
Distance to nearest interstate highway	0	1	0	0	0	1	0	0	0	6	5.200	0.745	1.000
Distance to nearest sea water trans dock	0	1	0	0	0	1	0	0	0	4	3.467	0.000	0.255
Local crime rate	0	0	0	0	0	0	0	1	0	6	2.250	0.139	0.080
Average one-way rush-hour commuting time	0	1	0	0	1	1	0	1	0	8	13.133	0.056	0.755
	0.86	0.50	1.00	0.69	0.40	0.37	0.90	0.38	0.58		**100.00**	**63.015**	**74.363**
											Rank	**2**	**1**

Figure 3.2: Spreadsheet display of the military value calculation for two depots

7. The next column, labelled *Weight*, shows the computed weight for each characteristic determined using the calculations we describe in appendix D. These weights, as required, sum to 100, the maximum military value possible. Notice the higher weights for characteristics associated with several criteria/attribute combinations.

The next column shows the calculated value, from a range of 0 to 1, given to each of the responses to the characteristics for Depot 1 as described previously. Likewise for Depot 2 in the last column.

The military value shown for Depot 1 at the bottom of the next-to-the-last column is obtained by summing the products of characteristic weight and the values of the responses corresponding to each characteristic. Likewise for Depot 2.

Chapter 4

Generating scenario alternatives using an optimization methodology

With even a small number of installations or activities, the problem of developing good recommendations can be daunting. Given just 10 installations or activities, there are 175 alternatives that close one, two, or three of the ten activities. It is unlikely that the time and resources needed to do an in-depth analysis of each of these possible alternatives will be available.[1] The optimization methodology described in this section provides a means of filtering the alternatives to find a good subset of these alternatives that can be used to develop scenarios for in-depth analyses in a timely and efficient manner. Final recommendations will come from the in-depth analysis of these scenarios as described in chapter 5.

We discuss the optimization methods for generating scenario alternatives in this section. We illustrate the use of these methods on the depot and DON examples described in appendices B and C, respectively.

[1] The problem of the number of alternatives to consider grows very rapidly with the number of activities under consideration. for example, given 20 activities, there are 1,350 ways to close one, two, or three activities.

4.1 Conceptual description of optimization methodology

The optimization methodology is intended to help generate alternative configurations for the infrastructure. Notionally, the optimization method *finds* the configuration that best meets the decision makers' goals among all the combinations that satisfy any requirements (constraints) that are imposed. Thus, the solutions to the optimization problem can serve as a basis for alternatives to be more thoroughly studied.

Generically, the optimization program consists of an objective function and constraints. The objective function defines the preferences among configurations. In this application, the objective function consists of two components: one for military value (where higher values are preferred) and one for size or capacity (where lower values are preferred). This objective function captures the trade-off inherent in BRAC: that we want to reduce the infrastructure, but in so doing, we must give up some capability. Thus, the military value component of the objective function increases as more places are retained, but at the same time, the *penalty* associated with retained size or capacity also increases.

The constraints define the set of *acceptable* configurations. At a minimum, the constraints in this application will be that the infrastructure must meet the required capacities. Additional constraints can be specified and utilized to further define the set of acceptable configurations.

The model incorporates several features that can be adjusted to obtain different alternative scenarios. The importance of military value relative to size can be adjusted by varying the magnitude of the penalty. This results in different scenarios to be studied. In addition, different proxies for measuring *size* or *capacity* can be used depending upon the decision maker's focus. Military value can be represented at a functional or activity level. In the following sections, we describe the optimization methodology and the design decisions more extensively.

4.2 Choosing the optimization goal

The optimization goal is represented by the objective function. As noted, the objective function in this application consists of a military value component and a size component. The specific definitions of these two components must be made by decision makers, to focus military value and size so that they reflect the environment and decision makers' priorities.

The military value can focus on either an activity-wide military value, or the individual functions of an activity. Size can focus on reducing the number of sites, the amount of resources retained, or both. Different combinations of the choice for the focus of military value and the goal for reducing the infrastructure result in four different methods as shown in table 4.1. The optimization methods are closely related because the intent of each is to develop solutions to the trade-off between keeping military value and reducing the infrastructure.

	Military value focus	
Reduction focus	Activity	Function
Reducing activities	Method 1	Method 3
Reducing capacity	Method 2	Method 4

Table 4.1: Optimization methods

Each method emphasizes different features and can produce different configurations of activities and functions as solutions. Thus, the choice of methods is an important decision. In some cases, decision makers might determine that only one combination of size and military value should be used. In other cases, more than one definition of either might be suitable. Determining the optimal solutions under the different approaches adds value by looking at the problem from different perspectives.

The options and implications for these four methods are described next.

4.2.1 Focus of military value assessments

How military value is measured and enters the optimization model is a key decision. The choice is between determining a single military value for an activity or determining a military value that is specific to each activity and function combination. This decision rests on multiple considerations. First, how important are the differences? If an activity-wide measure adequately captures the suitability of an activity to perform all of its functions, then a single value might be appropriate. On the other hand, if the activity is very well-suited for a particular function, but we want to consider it for performing some alternative function, then a function-specific military value might be important.

Second, how costly and difficult will it be to determine these military values? The burden of determining military values by functions is a consideration. It may be that determining a separate military value for every function requires so many resources that the cost of the analysis and the quality of the resulting values is in doubt. Determining an overall value may result in a position that is easier to defend.

These considerations are important. If the need for function specific values is high, then it may be necessary to pay the cost to derive the needed results. In addition, it may be necessary to determine the relative *importance* of each function and weight the functional military values accordingly so that the more important functions will play a larger role in determining the outcomes.

4.3 Procedures for generating alternatives

The purpose of the optimization methodology is to generate several alternatives that decision makers can evaluate in more depth. There are a number of ways that alternatives can be generated which we explore here. A full mathematical description of the methodology is given in appendix E. Appendix F shows the methodology coded using a well-known modeling language commonly used to solve mathematical programming problems.

4.3.1 Exploring tradeoffs between military value and infrastructure reduction

Varying the importance of reducing infrastructure provides different tradeoffs between military value and infrastructure, which are associated with different solutions. For example, the highest total retained military value is achieved with no infrastructure reduction, which corresponds to keeping everything open. At the other extreme, a solution with the most infrastructure reduction will most likely have a much lower total retained military value. The points in between may be obtained by changing the penalty on retained infrastructure in the objective function. They may also be obtained by varying a constraint on the retained size or capacity and eliminating the penalty on size or capacity. A number of alternatives are generated because each point represents a different solution that represents a particular tradeoff between retained military value and retained infrastructure.

4.3.2 Modeling expansion possibilities

The methodology will allow the exploration of resource capacity expansion at an activity. We have added this capability because we have seen several cases where a small-to-moderate increase in the amount of a resource available at an activity will generate good alternative scenarios that the methodology would not otherwise discover. Models can be run with resource capacities at the sites that reflect the current facilities or they can be run with resource capacities that represent future potential for the site.

4.3.3 Generating the second-best and third best solutions

The optimization methodology may be used to generate the 2nd and 3rd best solutions in addition to the best solution. This is accomplished by excluding the best (or best and 2nd best) solutions from the set of feasible solutions and running the optimization program again. The resulting solutions provide a set of high quality alternatives for consideration.

4.3.4 Using the four different methods to generate alternatives

Obtaining solutions using the different methods generates different alternatives. Each method is associated with a particular combination of retained military value (activity or function) and reduced infrastructure (numbers or capacity). Using different methods provides a measure of the robustness of a particular solution. If different methods give radically different solutions, then the analyst will know that the problem is very sensitive to the method used indicating that further analysis is required.

4.4 Simple example of optimization methodology

To illustrate how the proposed optimization methodology works, we will use another simple, five-site example. The example consists of five sites with different military values and capacities. These are summarized in table 4.2. The capacity requirement is 23. We interpret these military values as the *activity* military values. Thus, we will apply methods 1 and 2 to the problem. Note that the average military value of the five sites is 72.60.

Sites	A	B	C	D	E
Military Value	65	68	70	75	85
Capacity	4	7	10	13	15

Table 4.2: Five-site example: military values and capacities

Sites (1 - open, 0 - closed) A	B	C	D	E	Total military value	Total capacity	Fraction of capacity eliminated	Average military value	Graph label
1	1	1	1	1	363	49	0.00	72.60	W1, W2
0	1	1	1	1	298	45	0.08	74.50	X1
1	0	1	1	1	295	42	0.14	73.75	
1	1	0	1	1	293	39	0.20	73.25	
1	1	1	0	1	288	36	0.27	72.00	
1	1	1	1	0	278	34	0.31	69.50	X2
0	0	1	1	1	230	38	0.22	76.67	Y1
0	1	0	1	1	228	35	0.29	76.00	
1	0	0	1	1	225	32	0.35	75.00	
0	1	1	0	1	223	32	0.35	74.33	
1	0	1	0	1	220	29	0.41	73.33	
1	1	0	0	1	218	26	0.47	72.67	
0	1	1	1	0	213	30	0.39	71.00	
1	0	1	1	0	210	27	0.45	70.00	
1	1	0	1	0	208	24	0.51	69.33	Y2
0	0	0	1	1	160	28	0.43	80.00	Z1
0	0	1	0	1	155	25	0.49	77.50	BRAC 95
0	0	1	1	0	145	23	0.53	72.50	Z2

Table 4.3: Five-site example: data for feasible solutions

Given this simplified structure, the 18 feasible alternatives can be enumerated, as they are are in table 4.3. In the table, each row corresponds to a possible configuration, where a 1 means the site is retained in that configuration, and a 0 means it is closed. Corresponding to each configuration is a total military value (the sum of the military values of the retained sites), a retained capacity (the sum of the capacities of the retained sites), and a fraction of the capacity reduced.

4.4.1 Method 1 results for the five-site example

Recall that for method 1 we are penalizing the number of retained sites. Using different size penalties for the method 1 optimization model produces the configurations in the table labelled W1 through Z1. All eighteen configurations are plotted in figure 4.1. In this figure, we have plotted the reduction in retained sites against the retained total military value. We have also indicated the other solutions that the method 1 solution finds when we look

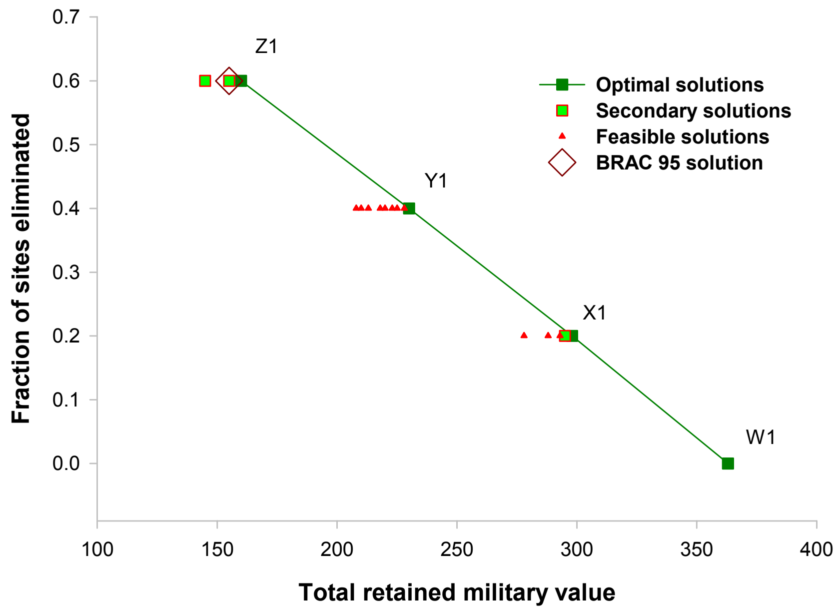

Figure 4.1: Method 1 results for the five site example

for the second- and third-best solutions. These are labelled as secondary solutions in the graph.

Solution W1 in the figure is the five-site solution that retains the entire infrastructure. It corresponds to no penalty, and thus the military value is 363. As the penalty for retaining sites is increased, we move to solution X1, giving up military value, but increasing the reduction in size. Solution X1 retains a total military value of 298 and a capacity reduction of eight percent. The other points near X1 are the other possible four-site combinations. Each of the other four-site combinations have lower total retained military values, and thus are not as attractive as the combination represented by solution X1. Solution Y1 and the associated cluster of points are the three-site combinations, resulting from increasing the penalty on sites retained even more. Solution Y1 is the optimal three-site combination and retains a total military value of 230 and a capacity reduction of about 22 percent. Finally, the last cluster around solution Z1 contains the two-site solutions. Here, only three combinations are feasible. Other two-site combinations might have higher military value, but they do not retain the required 23 units of capacity.

If we apply the BRAC 95 methodology, we would obtain the two-site

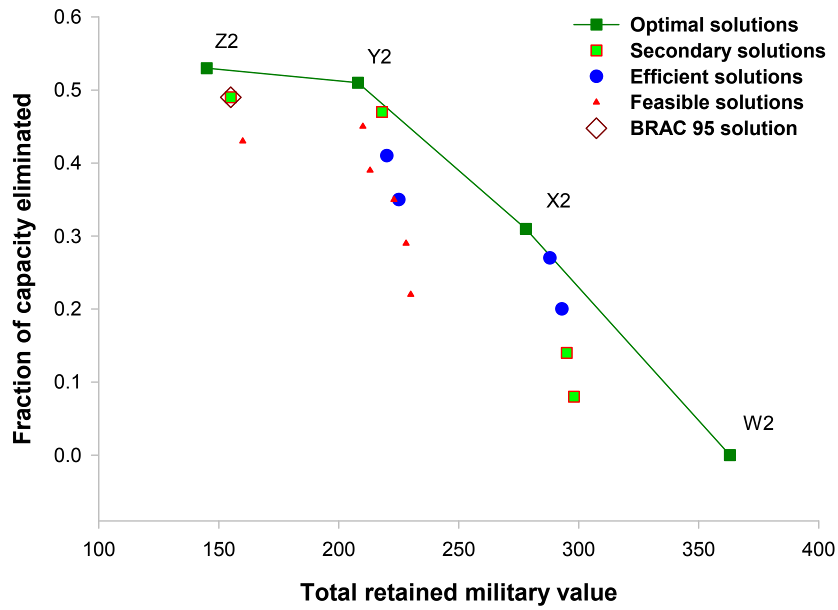

Figure 4.2: Method 2 results for the five site example

solution that retains sites C and E. The BRAC 95 solution is not a method 1 solution.

4.4.2 Method 2 results for the five-site example

For method 2 we are penalizing the retention of capacity. Using different size penalties for the method 2 optimization model produces the configurations in the table labelled W2 through Z2. All eighteen feasible configurations are plotted in figure 4.2. In this figure, we have plotted the reduction in retained capacity against the retained total military value. We have again indicated the other solutions that the solutions found when we look for the second- and third-best solutions. These are labelled as secondary solutions in the graph.

The W2 solution (identical to W1) retains all five sites. By increasing the penalty on retaining capacity, we move to solution X2. Increasing the penalty even further gives us solution Y2. A very high penalty results in solution Z2, which retains the minimum capacity of 23 units. Solution Z2 retains a total military value of 145. The average military value for solution Z2 is only 72.50, which is lower than the average for the original five sites.

This is not a BRAC 95 feasible solution because the average military value of the solution is lower than the starting average military value.

In figure 4.2 we have identified efficient configurations that are not identified as either an optimal solution or as a secondary solution (a second- or third-best solution)with blue dots. These are configurations that are efficient in the sense that there are no solutions that retain the same or more military value while retaining the same or less capacity.

In summary, the optimization methodology evaluates all possible configurations (the infeasible solutions are omitted from the graphs) and identifies those that are most promising, depending on the willingness to trade military value for infrastructure reductions (as simulated by the penalty term). These solutions (and possibly those that are closest to them, if desired) could then be examined in greater detail. This illustrates the filtering function of the optimization methodology–reducing a large number of possible configurations into a smaller number of the most promising configurations.

4.5 Constraining solutions to satisfy strategic requirements and policy imperatives

These models can be customized to incorporate important features in generating alternatives. Specific strategic requirements or policy imperatives can be addressed by adding constraints to the optimization program. For example, the Navy may have a policy imperative of ensuring that sizable fleets can be homeported on both coasts. Adding this as a constraint ensures that the solution meets that requirement. It may be necessary to include requirements in a model that support the transformational goals of DoD or achieve joint objectives for training or basing. These types of constraints may be added to a model.

The model solutions are always constrained to meet certain minimal capacity requirements. These requirements do not have to be a single set. There can be multiple sets of requirements corresponding to changing requirements over time. The resulting solution will accommodate all of the requirements through time. Similarly, higher capacity requirements can be established to ensure that the resulting configuration can meet surge requirements.

4.6 Other considerations

The choices regarding military value and infrastructure concerns are policy issues. The choice reflects how decision makers perceive military value and the primary objective of BRAC to reduce infrastructure. These choices actually encompass a number of different alternatives. Two in particular may be of interest. By adding a constraint on the number of sites that are open, the optimization will find the solution with the highest average value of sites for that number of sites.

An intuitive method based on rank-ordering activities by military value was considered. Activities are added to the solution in order of their military value ranking until the capacity requirement is met. (An analogous approach is to start with everything in the solution, and drop from the solution the lowest activity until dropping one violates the capacity constraint.) This greedy approach is relatively straightforward, but may result in a solution that can be improved upon by having higher total military value or lower excess capacity. This may happen because the stopping rule excludes consideration of potentially attractive alternatives. Because the solution can be improved using the other methods presented, we did not consider it further.

4.7 Examples used to demonstrate the methodology

We use two fictional examples to illustrate the different methods. We show how the choices for determining military value and infrastructure reduction can affect the outcome. However, as with any example, these are only illustrative and conclusions based on them may be misleading. A different set of values could lead to different conclusions. Thus, the decision should be based on understanding the issues involved.

4.8 Methodology applied to a JCSG decision problem

This JCSG-type example has seven depots that repair airframes, tanks, turbines, and electronics. The data for this example are given in appendix B. These data were used previously in chapter 2 to demonstrate a means for estimating capacity for a multi-dimensional case involving multiple products that are produced using shared resources. The military values for the

	Product types			
Depot	Air frames	Tanks	Turbines	Electronics
Alpha	32		131	1,937
Beta	19		107	4,215
Charlie	8		228	7,194
Delta		35.7	0	6,906
Echo		51	0	2,080
Foxtrot		29	0	10,203
Golf				8,108
Max production capacity	59	116	466	40,642
Requirement	40	32	300	4,500

Table 4.4: Depot product capacities

depots and the depot/function combinations are given in table B.4. For this example, reducing infrastructure is equivalent to freeing up resources such as test ranges, fabrication shops, hangars, and test facilities used to repair airframes, tanks, turbines, and electronics.

The methodology used in chapter 2 to estimate total capacity for all seven depots for the four product types also produces the capacity of each depot for each of the products. These capacities are given in table 4.4. Table 4.4 shows the total capacity across the seven depots as well as the requirement for each product type. This table represents just one of many possible production assignments and other production possibilities that exist, depending on how shared resources are used. For instance, the table shows no turbines produced at depot Foxtrot even though turbines could be produced there. Given the goal of maximizing the value of all products produced across the seven depots according the assumptions of chapter 2, the resources at depot Foxtrot were better used producing other products.

4.8.1 Depot example: method 1

Figure 4.3 presents a summary of the alternatives generated using method 1, which evaluated military value at the activity level and sought to penalize the number of sites retained. Table 4.5 shows the actual depots retained and shows the average military value, total military value retained, and the fraction of resources retained for a given number of retained activities. For example, the best solution that retains three sites (which are Alpha, Charlie, and Delta) retains a total military value of 198, the fraction of

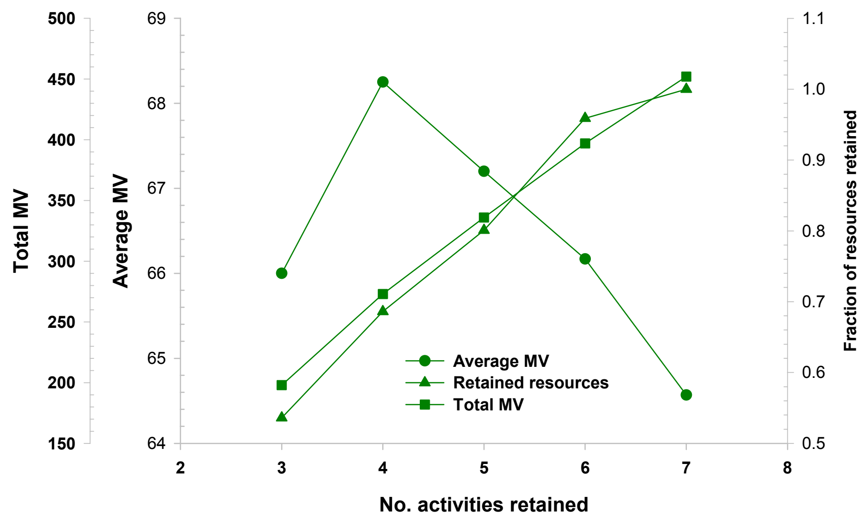

Figure 4.3: Method 1 results

resources retained is about 54 percent, and the average military value is 66.

The different solutions are obtained by varying the penalty associated with retaining a site. The solution retaining three sites results from the highest penalty for retained sites, the configuration that retains seven sites results from the lowest penalty for retained sites. Total military value (which is the sum of the military value of all retained sites) decreases as sites are closed. Average military value typically increases as sites with lower military value are closed first. This improvement in average military value may reverse as the quest for reduced infrastructure becomes dominant. In this regard, note that depot Foxtrot, the depot with the highest military value, is not included in the three-site solution. This is because depot Foxtrot does not have sufficient capacity in combination with any other pair of depots to meet the requirements.

A set of alternatives that decision makers can choose to examine in more depth is created. The decision makers can choose between solutions retaining three to seven activities by considering the trade-off in reduced total military value for a reduction in the resources retained.

Number retained	A	B	C	D	E	F	G	Average MV	Capacity retained
7	1	1	1	1	1	1	1	64.57	1.00
6	1	1	1	1	1	1	0	66.17	0.96
5	1	0	1	1	1	1	0	67.20	0.80
4	1	0	1	1	0	1	0	68.25	0.69
3	1	0	1	1	0	0	0	66.00	0.54
MV	62	61	67	69	63	75	55		

Table 4.5: Solution sets from method 1 for the depot example

Number retained	A	B	C	D	E	F	G	Average MV	Capacity retained
7	1	1	1	1	1	1	1	64.57	1.00
6	1	0	1	1	1	1	1	65.17	0.84
5	1	0	1	0	1	1	1	64.40	0.70
4	1	0	1	0	1	0	1	61.75	0.55
3	1	0	1	0	1	0	0	64.00	0.51
MV	62	61	67	69	63	75	55		

Table 4.6: Solution sets from method 2 for the depot example

4.8.2 Depot example: method 2

In figures 4.4 through 4.6, we have overlaid the results for total retained military value, average military value, and fraction of resources retained, respectively, from using method 2 on top of the results from method 1.

For example, as we move from 7 retained depots to 6, penalizing the number of activities retained (method 1) results in a smaller reduction in excess capacity as well as a smaller reduction in the total retained military value compared to penalizing the retention of excess capacity (method 2). Method 1 removed activity Golf that has the smallest activity military value of all of the activities. Method 2 removed depot Bravo that has a higher military value than depot Golf, but results in the retention of fewer resources. The solutions obtained with method 2 are shown in table 4.6.

4.8.3 Depot example: methods 3 and 4

For methods 3 and 4, we switch to measuring the military value of performing each function at a depot. These military values are given in table B.4.

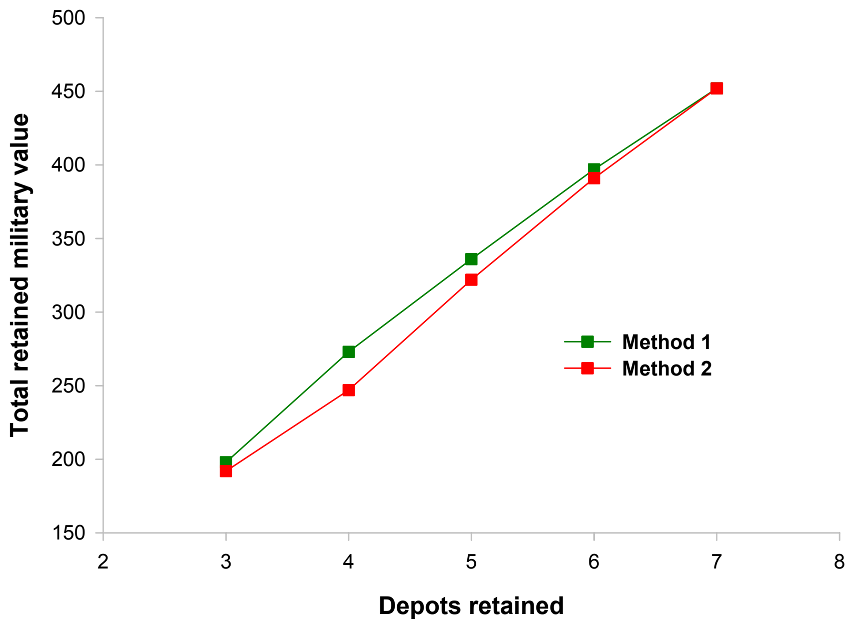

Figure 4.4: Method 2 results for total retained military value compared to method 1

Table B.4 also shows the importance we have assigned to each of the product (function) lines. Air frames and tanks are twice as important as the other two product lines. For method 3, we maximize the total retained functional military values weighted by the importance. We also normalize the functional military values by dividing all of the functional military values for a product line by the maximum functional value for that product line, while penalizing the retention of depots. For method 4, we maximize the total retained functional military values weighted by the importance while penalizing the retention of resources. We also normalize the functional military values as we did for method 3. If a depot is retained, we assume that all of its resources are retained regardless of the assignment of workload to the depot.

Figures 4.7 through 4.10 plot the results for methods 3 and 4 for air frame, tank, turbine, and electronic repair, respectively.

The resource capacity reduction obtained with methods 3 and 4 are plotted in figure 4.11.

Table 4.7 shows the depots retained by each method 3 solution while

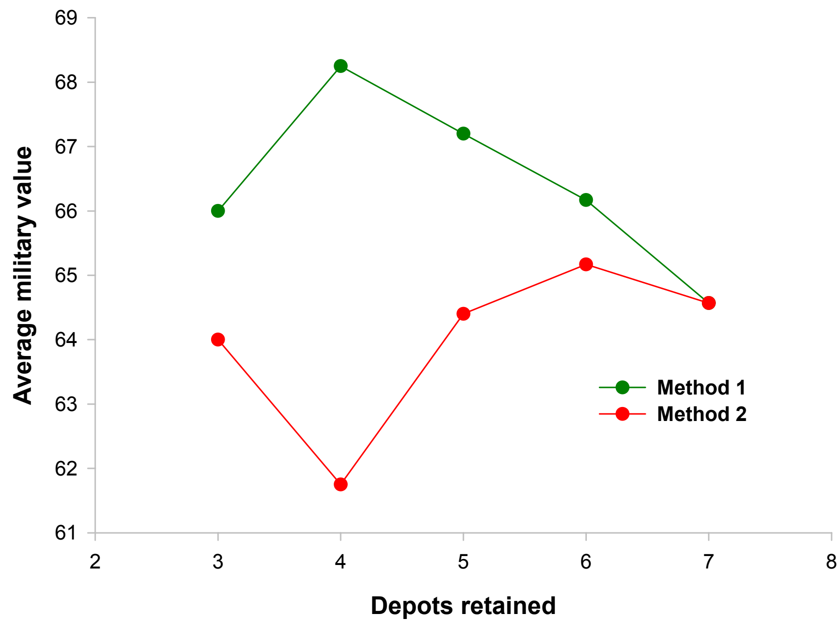

Figure 4.5: Method 2 results for average military value compared to method 1

table 4.8 shows the depots retained by each method 4 solution.

4.9 Methodology applied to the DoN decision problem

The problem of developing recommendations becomes more complex if we must simultaneously consider the entire set of functions that compete for location on installations. However, that is precisely the problem DoN faces in this current BRAC round. In previous BRAC rounds, we simplified matters by treating an installation as if it supported one primary functional category. With many installations hosting a variety of functions, this assumption is less appropriate. Indeed, with growing interest in cross-service use of facilities and security imperatives that favor relocation of administrative and support activities onto military bases, an integrated look at the use of base resources is essential.

The methods presented here extend the example presented in the previous section to consider a set of installations that can host several types

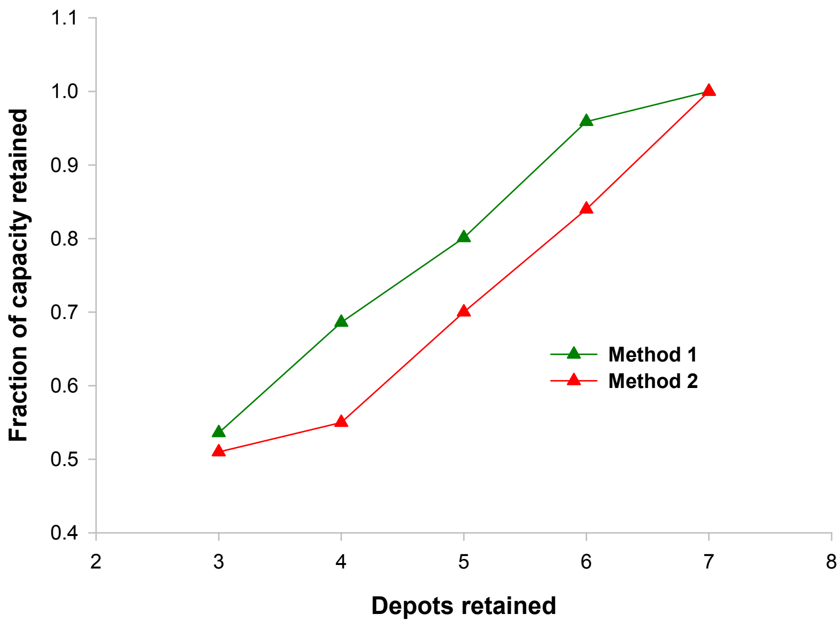

Figure 4.6: Method 2 results for retained capacity compared to method 1

of activities, including operational, administrative and support activities. We will consider alternatives for relocating the various functions performed by these activities. We discuss methods for generating and filtering such alternatives, illustrating the methods with an example that is described in greater detail in appendix C.

4.9.1 Retaining military value

Our objective function, as before, rewards retaining military value. Exactly how military value enters the optimization model is again a key decision. Here, we consider military value to be function specific, with a distinct value for each function and installation to which the function may be assigned. Overall military value is the sum of the functional values that result from locating functions at specific bases. While it is a challenge to assess distinct values for each function and site, the alternative of a common installation-wide value for all functions is no longer appealing. Now that we are simultaneously considering the assignment of all functions, the relative values across functions become a critical factor in determining how bases will be used. It

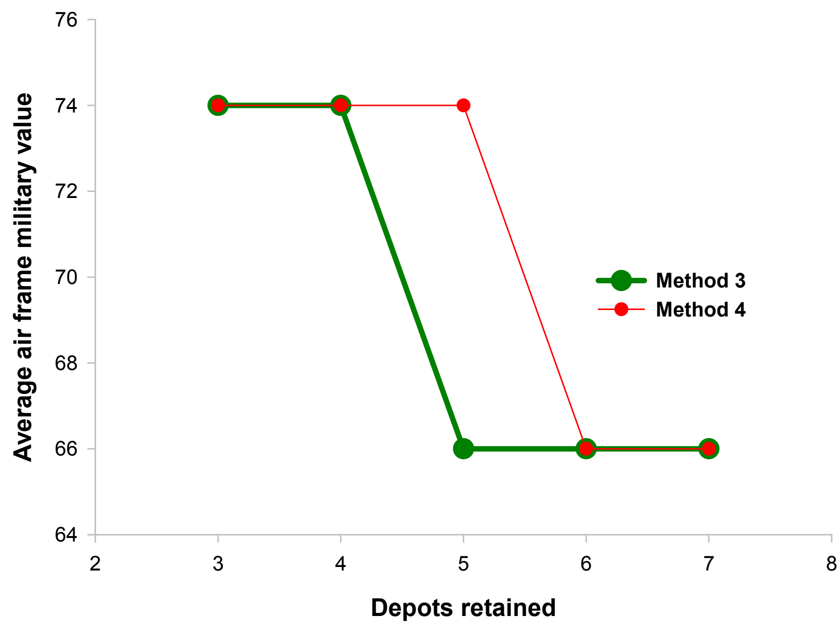

Figure 4.7: Method 3 and 4 results for air frame repair

is unlikely the exact same ranking of locations would hold for every category of operational and support function. These considerations were not much of a concern when we were looking at a single category of activities in isolation as in the previous depot example.

4.9.2 Reducing infrastructure

The goal of reducing infrastructure enters our objective function in three ways. The infrastructure goal may be to reduce the number of installations, the number of activities, or the excess capacity in resources. The goal of reducing the number of activities is actually somewhat less than convincing in the current context. There may be limited savings from closing an activity on a base that otherwise remains open. In fact, in our example, we consider the facilities freed up by closing an activity to be retained and available for other uses on the base. The primary benefit of closing activities is thus in the ability to consolidate functions on fewer bases, perhaps using less resources in total. These goals are already embodied within the other two infrastructure elements of the objective function.

Our focus then will be on reducing installations and eliminating excess

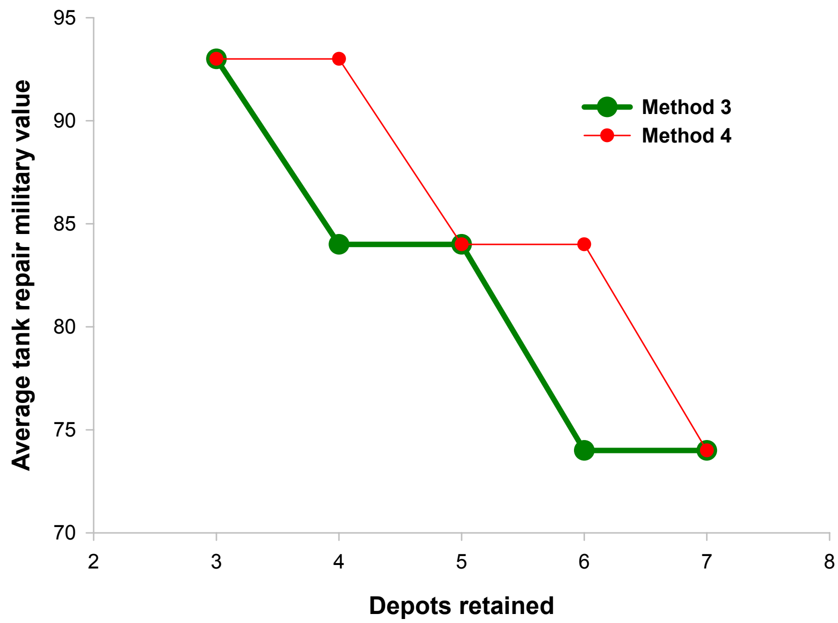

Figure 4.8: Method 3 and 4 results for tank repair

capacity. Which of these is most important depends on priorities and specific costs. Is it better to shut down three installations to reduce general overhead costs or two bases where there was a heavier concentration of excess capacity? Either may be appropriate. Varying the relative weights on these two elements within the objective function will be used to generate different scenarios for consideration.

4.9.3 Expansion of infrastructure

Our methodology allows for some expansion of resources, but penalizes expansion in the objective function. It might seem strange to allow for any expansion given the presumption of BRAC that there is current excess capacity. However, there are situations where adding key resources at one installation may allow the retention of higher-valued facilities or closure of additional facilities. By adjusting the penalty on infrastructure expansion, we can explore the extent to which it is possible to achieve overall gains at a reasonable cost.

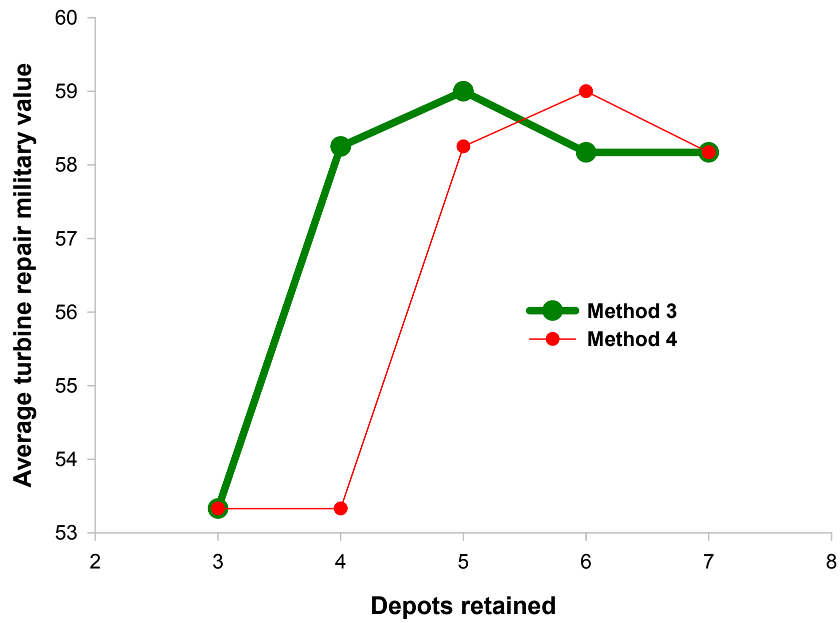

Figure 4.9: Method 3 and 4 results for turbine repair

4.9.4 Constraining solutions

All solutions are constrained to meet certain minimal requirements. These include:

- *Functional requirements*: The desired forces and functional requirements must be assigned to a site. (This can be thought of as a workload requirement.)
- *Resource capacity constraints*: The functions assigned to an installation can use no more resources than are available (including any added by expansion).
- *Feasibility constraints*: Certain conditions that may rule out assigning a particular function to certain locations (e.g., aircraft require adequate runway length and access to training ranges; ships require adequate water channels).
- *Strategic requirements and policy imperatives*: Strategic requirements and policy imperatives can be addressed by adding further constraints, e.g., geographic dispersion requirements.

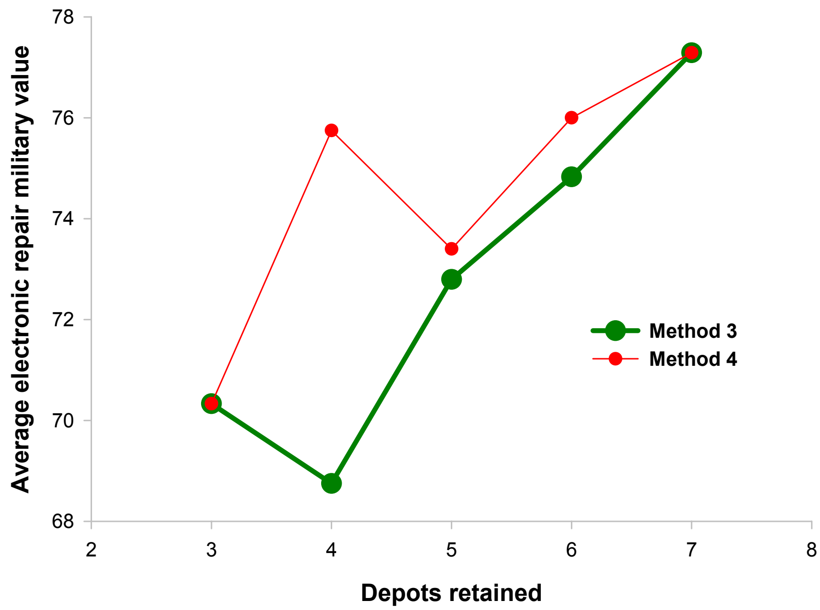

Figure 4.10: Method 3 and 4 results for electronics repair

The feasibility constraints noted above were not considered in the depot example. They become more relevant now that we are considering each site as a potential location for each function. They can be very helpful in ensuring that functions are located on appropriate sites.

4.9.5 Procedures for generating DoN alternatives

We may generate alternatives for DoN decision maker consideration by varying the emphasis on the infrastructure elements in the objective function. Increasing the emphasis on infrastructure results in a tradeoff of military value for further resource reductions. Allowing for targeted resource expansion tends to improve this tradeoff, with a greater reduction in resources possible while retaining the same military value.

We use a fictional example to illustrate the application of these methods to the DoN decision problem. The results are illustrative, intended only to highlight the issues involved.

In this example, we begin with 18 installations and 24 activities located across these installations as described in appendix C. Activities include headquarters, intelligence centers, depots, supply centers, naval stations, air

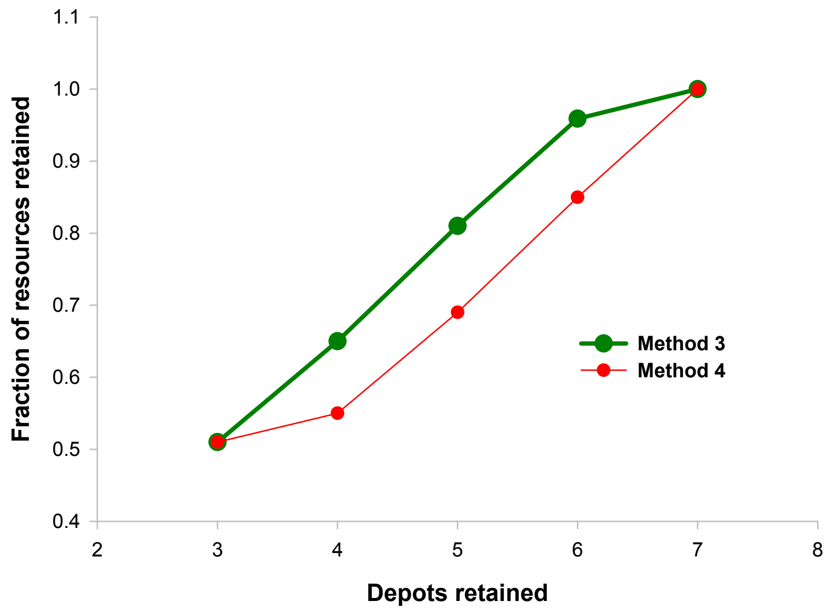

Figure 4.11: Method 3 and 4 results for capacity reduction for the depot example

stations, Marine divisions, Marine air wings, and an aircraft test center. These activities perform one or more functions. Depots, for example, repair airframes, tanks, turbines, or electronics in various combinations; air stations can host squadrons of fighter, attack, or experimental planes; and naval stations may berth carriers or cruisers. There are 15 functions in total, each with a given requirement. These functions must be assigned to locations. Each function has specific resource needs that limit where it can be located. Military values reflect preferences for where a function may best be located. The facilities used by the activities offer a variety of resources, some of which are fairly specific to the category of function performed (e.g., fabrication shops, berths, and hangars) and others that are generic (e.g., office space, storage space, and utilities). When existing functions move, the resources are considered to be freed up and available to any other function that may locate on the installation.

4.9.6 Generating alternatives: reducing installations

Figure 4.12 presents a summary of alternatives generated by reducing the number of retained installations. The various solutions shown were gen-

Number retained	A	B	C	D	E	F	G	Capacity retained
7	1	1	1	1	1	1	1	1.00
6	1	1	1	1	1	1	0	0.96
5	1	1	1	1	1	0	0	0.81
4	1	0	1	1	1	0	0	0.65
3	1	0	1	0	1	0	0	0.51
Product			Functional values					
Air frames	82	50	66					
Tanks				75	93	54		
Turbines	35	62	81	73	44	54		
Electronics	57	89	80	64	74	85	92	

Table 4.7: Solution sets from method 3 for the depot example

erated by gradually increasing the objective function penalty on retained installations. The solutions shown in green correspond to not allowing any expansion. Solutions shown in red allowed for expansion. For each solution, the figure shows the total retained functional value and the number of installations closed. For example, the solution labelled H results in the closure of nine installations and retains functional values totaling 997.

In our example, because of substantial excess capacity and less than ideal placement of existing functions, six installations can be closed with no loss of functional value. Additional closures will come at the expense of increasing loss in functional value. A maximum of 12 installation can be closed before it becomes impossible to support the desired workload and force requirements with the facilities retained. As can be seen in figure 4.12, the possibility of well-targeted expansion of resources results in greater value being retained for any given number of base closures.

Further information on each solution is provided in table 4.9. For example, the table shows that solution H results in the closure of five major installations[2] and leads to a 31-percent reduction in resource capacity.

4.9.7 Generating alternatives: reducing capacity

Figure 4.13 presents a summary of alternatives generated by reducing capacity. The solutions shown here were generated by changing the objective function penalty on retaining resource capacity. The green solutions did not

[2]For this example, a major installation is one that hosts operational forces such as air stations, naval stations, etc.

Number retained	A	B	C	D	E	F	G	Capacity retained
7	1	1	1	1	1	1	1	1.00
6	1	1	1	1	1	0	1	0.85
5	1	0	1	1	1	0	1	0.69
4	1	0	1	0	1	0	1	0.55
3	1	0	1	0	1	0	0	0.51
Product			Functional values					
Air frames	82	50	66					
Tanks				75	93	54		
Turbines	35	62	81	73	44	54		
Electronics	57	89	80	64	74	85	92	

Table 4.8: Solution sets from method 4 for the depot example

allow for expansion. The red solutions did allow for expansion. Again, we can see the tradeoffs involved. We see the tradeoff between functional value and resource capacity. Further summary measures on the solutions are presented in table 4.9. In comparison to the solutions generated by reducing the number of installations, those shown here tend to favor the closure of more substantial facilities as a means to achieve capacity reduction.

4.9.8 A comparison of solutions

In table 4.9, there are five solutions that result in the closure of nine bases. We compare three of these solutions to highlight various aspects of the optimization approach. Tables 4.10, 4.11, and 4.12 display the final status of each base and any major adjustment that took place for each of these solutions. Table 4.13 shows the total retained functional value for each solution and the percent reduction in retained resources for each of these solutions.

Review of solution I

Solution I was generated with a moderately high penalty on the number of installations retained. It allows for no expansion of resources. It results in nine closures, a 33-percent reduction in resources, and a total retained functional value of 992. There are five major base closures: three air bases, one naval station, and one Marine Corps base. These closures are accomplished by moving forces to other bases with excess capacity. Two administrative headquarters are co-located, allowing for the closure of one facility. Simi-

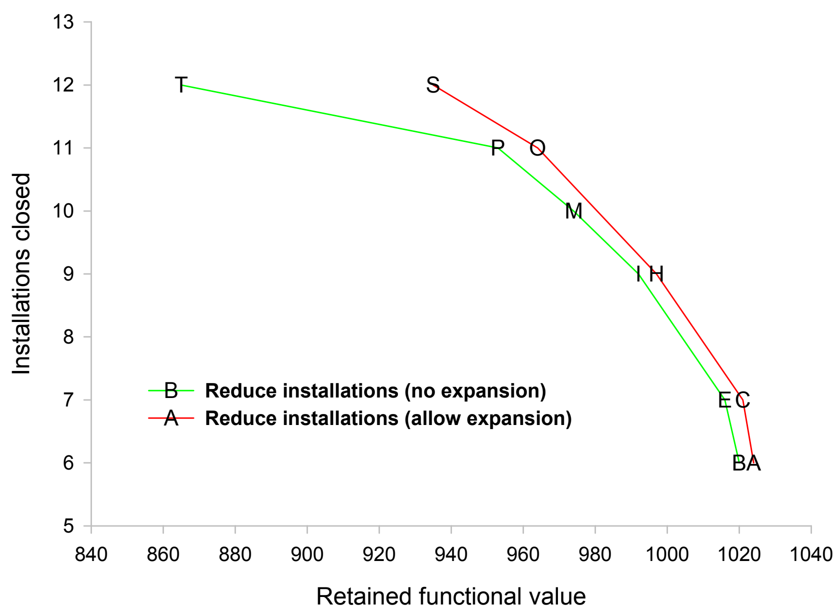

Figure 4.12: DON alternatives generated by reducing the number of installations

larly, two intelligence centers are co-located on a military base, allowing the closure of one stand-alone facility. The two supply centers are moved to available space on different bases. There are no closures among the four depots. This last item is noteworthy because our depots are the same as four used in the earlier example (Depots A, B, D, and E are taken from the depot example above). The advantage of shutting a particular depot changes once we account for their location on an installation that may otherwise remain open.

In comparison to solutions A or B (six bases closed), solution I closes an additional air station, another administrative facility, and a Marine Corps base. Less than a three-percent loss in military value results from these additional closures. Meanwhile, the reduction in capacity achieved has climbed from 20 percent to 33 percent.

Review of solution J

Solution J is generated with a high penalty on retained capacity. It allows for expansion of resources. It results in nine closures, a 40-percent reduction in resources, and total retained functional value of 989. There are now six

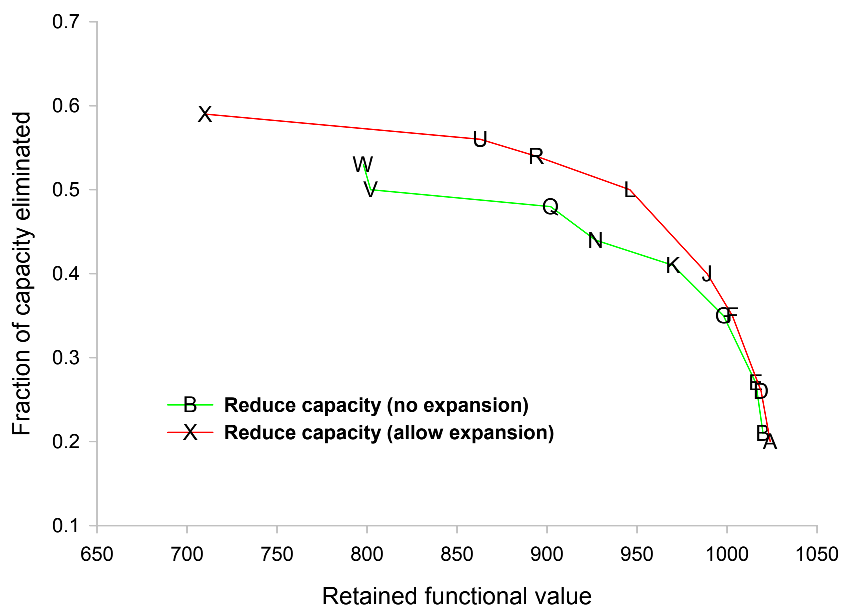

Figure 4.13: DON alternatives generated by reducing capacity

major bases closed. In comparison to the previous solution, an additional air base is closed. This is made possible by the construction of an additional hangar and office space at another base. One depot is now closed, with its workload transferred to the three remaining depots. The co-location of headquarters no longer occurs; so one administrative facility is open that was shut under solution I.

Where the previous solution emphasized closing installations with little regard to size, the reward here is on reducing excess capacity. As a result, the closures here tend to be rather more substantial. That is apparent in the 40-percent reduction in overall resource capacity (up from 33 percent). That gain is achieved with an almost trivial loss in total retained functional value, although at some cost for expansion.

Review of solution L

Solution L is generated with a very high penalty on retained capacity. Again, it allows for resource expansion. It results in nine closures, a 50-percent reduction in resources, and a total retained functional value of 946. In this solution, we have a new combination of closures among the major bases. Two of the bases closed under solution J are kept open (an air base and Marine Corps base) and two somewhat larger bases are closed. The closed bases now include Base 7 (which served as host to an air station, naval base, and depot) and Base 6 (which housed aviation development and an intelligence center). These closures were made possible by a substantial investment at other locations: several berths added at two naval stations and hangar facilities added at an air station. Two depots are now closed. The intelligence centers consolidate at a single location, but the administrative facilities are all kept open. Maintaining the small administrative installations is necessary to allow closure of a larger facility.

This solution results in a significant reduction in capacity, with overall capacity reduced by 50 percent. There is, however, a noticeable reduction in total retained functional value. Further, the required investment in berths and hangar facilities seems to be substantial.

4.9.9 DoN example summary

Simultaneous analysis of the various categories of functions has a number advantages in terms of realism over the single category approach taken in prior BRAC rounds. At the same time, this more realistic approach introduces a number of practical complications. We summarize some of the advantages and disadvantages.

Advantages:

- The simultaneous approach allows the model to retain or close a particular activity while taking into account the status of the rest of the base.
- The simultaneous approach allows for greater flexibility in the alternative uses for resources freed up when a function moves.
- The simultaneous approach allows for greater flexibility in locating activities. The single category approach must either constrain solutions to existing locations or risk inconsistencies between solutions.

Disadvantages:

- With flexibility comes complexity.
- Careful attention to the comparative levels of military values between functions becomes critical.

- The potential flexibility in locating functions requires the evaluation of a greater number of functional values and calls for explicit attention to feasibility conditions.

- Allowing for flexibility in resource use requires greater attention to modeling resource use.

4.10 More on allowing expansion

The linear programming algorithms we use to solve the optimization problems are very precise. If an activity has only 2.99 units of a resource needed for a function that requires 3.00 units, the function will not be placed at that activity. It is prudent when generating alternative scenarios to allow for the possibility of expanding resources at an activity or installation in seeking solutions with the optimization model. We have built this capability into the methodology. We use the three-depot solution from the method 4 results given in table 4.8 to demonstrate this capability.

Three is the smallest number of depots that can handle the required yearly workload if expansion is not allowed. If we allow the depots to expand their resources by the amounts shown in table B.2, we get a two-depot solution that retains depots Alpha and Echo. The resources added included additional capacity for fabrication shops and an additional hangar at Alpha and additional test facility capacity at Echo. Table 4.14 compares the average functional values and the capacity retained for the two solutions. Allowing for the expansion of capacity actually lowers the overall capacity retained.

This example shows the value of allowing some expansion of resources. With only a small increase in a few resources, it may be possible to find significantly better scenario alternatives.

4.11 Generating the second- and third-best solutions

We will use the depot example to demonstrate the generation of second- and third-best solutions. We will start with the four-depot solution for method 1. Table 4.15 shows the three solutions. The methodology for generating the second- and third-best solutions is described in appendix E.

4.12 Setting the penalty parameters

The setting of the penalty parameters will usually require some experimentation to find a useful range of values for a particular penalty parameter. We varied the value of the penalty parameter on the number of sites from a low of 50 to a high of 80 to obtain the results shown in table 4.5. Having found a working range, we then try intermediate values to obtain a range of solutions.

Because we cannot predict in advance of running a particular model what a good range for a penalty parameter will be, it is necessary to allow experimentation to determine a good working range. Once a good working range is determined, alternate solutions can be generated by trying values of the penalty parameter that lie inside the working range.

We recommend using a range of penalty parameters for the following reasons. A set of solutions can be generated using a range of penalty parameters. An activity or site that is retained in most of these solutions is probably one that should be retained. An activity or site that is not retained in most of the different solutions is a candidate for closure or realignment. Activities or sites that fall into neither of these two groups require a more detailed analysis.

Solution label	Installations closed	Major closures	Total retained functional value	Fraction of capacity eliminated	Resource expansion allowed?
A	6	3	1,024	0.20	Y
B	6	3	1,020	0.21	N
C	7	4	1,021	0.25	Y
D	7	4	1,019	0.26	Y
E	7	4	1,016	0.27	N
F	8	5	1,003	0.35	Y
G	8	5	998	0.35	N
H	9	5	997	0.31	Y
I	9	5	992	0.33	N
J	9	6	989	0.40	Y
K	9	5	970	0.41	N
L	9	6	946	0.50	Y
M	10	5	974	0.38	N
N	10	4	927	0.44	N
O	11	6	964	0.40	Y
P	11	5	953	0.39	N
Q	11	6	902	0.48	N
R	11	6	894	0.54	Y
S	12	6	935	0.46	Y
T	12	6	865	0.46	N
U	12	6	863	0.56	Y
V	12	6	802	0.50	N
W	12	6	798	0.53	N
X	12	6	710	0.59	Y

Table 4.9: Candidate solutions generated for the DON example

Site	Activities	Status	Comments
Air Station 1	NAS 1, Depot 2	Open	Receives ICP
Air Station 2	NAS 2	Close	
Air Station 3	NAS 3	Close	
Air Station 4	NAS 4	Open	
Air Station 5	NAS 5	Close	
Air Station 6	RDT&E, Intel E	Open	Receive Intel W
Dual 1	NAS 6, NS 4, NS 4, Depot 1	Open	
Station 1	NS 1	Open	
Station 2	NS 2	Close	
Station 3	NS 3	Open	
MCB 1	MCEF 1, MCAW 1	Close	MCAW 1 to Air Station 4 MCEF 1 split to Air Station 6 and Dual 1
MCB 2	MCEF 2, Depot 3	Open	
Support 1	ICP 1	Close	ICP to Air Station 1
Support 2	ICP 2	Close	ICP to MCB 2
Support 3	Depot 4	Open	
Admin 1	HQ E	Close	HQ E to Admin 2
Admin 2	HQ 2	Open	Both HQ locate here
Admin 3	Intel W	Close	Intel W to Air Station 6

Table 4.10: Detailed results for DoN Solution I

Site	Activities	Status	Comments
Air Station 1	NAS 1, Depot 2	Close	Depot 2 closes squadrons move
Air Station 2	NAS 2	Close	
Air Station 3	NAS 3	Close	
Air Station 4	NAS 4	Expand	Add air capacity
Air Station 5	NAS 5	Close	
Air Station 6	RDT&E, Intel E	Open	Receive Intel W
Dual 1	NAS 6, NS 4, NS 4, Depot 1	Open	
Station 1	NS 1	Open	
Station 2	NS 2	Close	
Station 3	NS 3	Open	
MCB 1	MCEF 1, MCAW 1	Close	MCAW 1 to Air Station 4
MCB 2	MCEF 2, Depot 3	Open	
Support 1	ICP 1	Close	ICP to Air Station 1
Support 2	ICP 2	Close	ICP to MCB 2
Support 3	Depot 4	Open	
Admin 1	HQ E	Open	
Admin 2	HQ 2	Open	
Admin 3	Intel W	Close	Intel W to Air Station 6

Table 4.11: Detailed results for DoN Solution J

Site	Activities	Status	Comments
Air Station 1	NAS 1, Depot 2	Open	
Air Station 2	NAS 2	Close	
Air Station 3	NAS 3	Close	
Air Station 4	NAS 4	Expand	Further air capacity
Air Station 5	NAS 5	Close	
Air Station 6	RDT&E, Intel E	Close	Intel E moves to Admin 3
Dual 1	NAS 6, NS 4, NS 4, Depot 1	Close	Depot 1 closed ships, and squadrons move
Station 1	NS 1	Open	Add berthing capacity
Station 2	NS 2	Close	
Station 3	NS 3	Expand	Add berthing capacity
MCB 1	MCEF 1, MCAW 1	Expand	Add air capacity
MCB 2	MCEF 2, Depot 3	Open	
Support 1	ICP 1	Close	ICP to Air Station 1
Support 2	ICP 2	Close	ICP to MCB 2
Support 3	Depot 4	Close	Depot 4 closed
Admin 1	HQ E	Open	
Admin 2	HQ 2	Open	
Admin 3	Intel W	Open	Both Intel locate here

Table 4.12: Detailed results for DoN Solution L

	Solution		
Statistic	I	J	L
Retained functional value	992	989	946
Percent capacity eliminated	33	40	50

Table 4.13: Summary statistics for selected DON solutions

	Average functional values	
Product	No expansion	Expansion allowed
Air frames	74.00	82.00
Tanks	93.00	93.00
Turbines	53.33	39.50
Electronics	70.33	65.50
Capacity retained	0.51	0.40

Table 4.14: Comparison for non-expansion (ACE) and expansion (AE) examples

Solution	A	B	C	D	E	F	G	Total MV	Average MV	Capacity retained
Best	1	0	1	1	0	1	0	273	68.25	0.69
Second-best	1	0	1	1	1	1	0	336	67.20	0.80
Third-best	1	1	1	1	0	1	0	334	66.80	0.84
MV	62	61	67	69	63	75	55			

Table 4.15: Second- and third-best solutions for the depot example

Chapter 5

Scenario analysis

Along with other sources of scenarios, the decision makers for the BRAC process could choose from the solutions generated with the optimization model those which would undergo detailed analysis with respect to the last four selection criteria: cost, economic impact, community impact, and environmental impact. The preceding steps of the process have filtered all of the many possibilities down to a select set for this final step prior to the making of recommendations.

In this chapter, we briefly discuss these criteria to provide context for the next step in the DoN BRAC process after the scenario generation step. The discussion is meant to emphasize that the outputs from the optimization model are not the final outputs in the process of making BRAC recommendations.

5.1 COBRA analysis

After using the optimization tool to generate a realignment or closure alternative, military departments and JCSGs must conduct more detailed cost analysis of a given alternative (or scenario) using a common costing tool provide by OSD. This software tool, called *Cost of Base Realignment Actions* (COBRA), contains many pre-loaded base characteristics and standard cost factors to simplify and ensure a consistent methodology behind cost calculations. Despite being very detailed in its scope, COBRA is intended to provide only macro-level estimates for comparative purposes, not budget-fidelity numbers. OSD also provides documentation for the COBRA model.

5.1.1 How COBRA works

COBRA requires a user to input certain details of the particular realignment scenario, such as how many personnel and how much equipment will be moving from one base to another and how many and what types of facilities need to be built or renovated to accommodate them. Based on these user-specified data, as well as many pre-loaded standard cost factors, COBRA calculates a net present value for the scenario.

5.1.2 What COBRA includes in costs and savings

The purpose of COBRA is to provide a uniform methodology for estimating and itemizing the costs and savings associated with each realignment scenario to be considered. COBRA considers both one-time and recurring costs and savings. Among one-time costs associated with implementation of a realignment scenario are:

- *Personnel* Severance, unemployment, additional hiring required.
- *Construction* If new facilities or renovation is required.
- *PCS costs* Travel and homeowners' assistance program.
- *Transportation* Transportation for equipment and vehicles that must be moved.
- *Miscellaneous mission-specific costs* If special equipment cannot be moved, what costs will be incurred to duplicate the capability at the receiving site?

COBRA also considers any changes in recurring costs that result from a given scenario (Typically savings from a scenario are realized in these categories.):

- *Personnel changes* Changes in payroll and housing allowances.
- *Overhead changes* Changes in costs for facility sustainment, facility recapitalization, and base operations support (BOS).
- *Medical cost changes* Changes in medical treatment costs for active duty members, their families, and retirees.

5.1.3 COBRA assumptions

COBRA requires all realignment or closure actions to be completed within six years, such that a new "steady state" is reached by year six. As a result, COBRA assumes that net recurring savings occurring in the sixth year will continue to accrue annually through year 20.

Other assumptions built into COBRA include a standard discount rate (determined in accordance with OMB Circular A-94), average salaries and housing allowances, standard construction cost factors by type of facility, and current installation-specific data such as base population and operating expenses.

5.1.4 COBRA flexibility

In some unique realignment scenarios, the standard cost factors built into COBRA may not approximate a known saving or cost with sufficient accuracy. In such cases, the COBRA user has the option to enter these unique costs or savings directly into the calculation. Whenever unique items are entered, however, the user must also enter a footnote that documents and justifies (in an auditable sense) the unique entry. These footnotes become part of the COBRA output reports.

5.1.5 COBRA output reports

There are several reports generated by COBRA. These include:

- *Realignment Summary* Positions eliminated/realigned by year, net costs by category by year, NPV and payback year.
- *Net Present Values Report* The cumulative NPV of savings through years 1-20.
- *One-time Costs Report* Itemized by base.
- *MILCON Report* Itemized by base
- *Personnel Report* Changes by year and by base
- *Input Data Report* Lists all data used for the scenario calculation, as well as footnotes.
- *Scenario Error Report* Would indicate if total population movements are inconsistent, for example.

5.2 Economic impact

In addition to calculating the cost of the scenario with COBRA, separate analysis is necessary to assess the economic impact of the proposed scenario on communities in the vicinity of the military installations under consideration. The *Economic Impact Tool* (EIT) is provided by OSD for performing the economic impact analysis.

This economic impact analysis will help decision makers evaluate alternative closure and realignment scenarios and gain a better understanding of:

- The potential impact of a decision on all aspects of a local economy
- The potential for an activity workforce to find suitable employment within the area after the closure
- The issues which might affect the ability of an area to recover from the closure decision

The local community in the vicinity of military installations is the geographic area in which most of the installation's employees live and work. The definition of a local community for a given installation can be built from existing, well-defined geographical areas, such as Metropolitan Statistical Areas (MSAs) or counties. These standard geographic areas have the advantage that a wealth of data is available to measure the relevant demographic and economic variables specific to these areas.

5.2.1 Forecasting job changes

Net job change in the local area is the primary indicator used to measure the economic effect of installation closure on the local economy. Specifically, the analysis estimates the total (sum of direct and indirect) potential job change in the area, expressed both as an absolute total and as a percentage of the local area employment. Using both an absolute and relative measure takes into account differing sizes of economic areas when considering impact.

Direct employment change includes military personnel, DOD civilian employees, military students, and on-base contractors. Indirect employment change is estimated using multipliers that take into consideration the size of the economic area, the type of installation being evaluated (e.g. an industrial vs. and operational activity), and differences in impact associated with different categories of employees. There are a number of currently available models that forecast the net economic impact of a given change in a local economy. To the extent these models can be tailored to fit the base closure process, they may be used as a method of corroborating results.

5.2.2 Economic profiles of communities

The current economic condition and recent economic trends in a community provide additional context within which to assess a given potential change in employment. As a result, an additional part of the economic impact analysis is to compile a statistical economic profile of each area, covering current conditions and historical trends. These statistics can then be compared to national averages to give a better picture of the relative economic health of local areas. Most of these data are available from other government sources, such as the U.S. Census Bureau, the Bureau of Economic Analysis, and the Bureau of Labor Statistics. Some indicators may be available only at the state level, but may nonetheless provide additional insight into the health of an area's economy.

The local area statistical profile includes indicators such as:

- Demographic data (population, age and education profiles, migration patterns)
- Employment data (unemployment rate, per capita personal income by private/public sector, employment by occupation and industry, earnings by industry)
- Other economic data (diversity of industry, diversity of residents, housing cost relative to local income, federal expenditures, small business net incorporations, duration of unemployment)

Finally, some of the data in the economic profile of a local area may have additional application. For example, as a follow-on analysis, the employment data collected by occupation and industry may allow comparison of DoD job skills with the predominant requirements in the local economy. Such a comparison can help determine whether DoD employees who could be potentially released into the local economy are involved in occupations/industries for which there is large demand.

5.3 Community impact

Criteria 7 specifies that the decision process should include an assessment of impact of any closure or realignment on the infrastructure of the communities involved. Before any recommendation is put forward, factors such as cost-of-living, educational opportunities, child care facilities, spousal employment opportunities, housing availability, transportation, and other relevant factors should be assessed in terms of the good and bad points with respect to the recommendation.

Strictly speaking, the analysis will go beyond just impacts on the community to include how attractive the community would be as a host for military activities. How attractive the community may be will depend to some degree on the type of activities that would reside there and the particular needs of the uniformed and civilian personnel associated with those activities.

Part of the assessment must also focus on the ability of the community's infrastructure to absorb additional loading that would accompany any transfer of additional activities or functional load into the community. Specifically, can the community schools absorb additional students? Is the road system adequate to support additional activity? Is there enough capacity in the various utilities to not only support additional loading on the local installation, but also to support the additional citizens in the local community.

5.4 Environmental impact

Military use of land, air, coastal, sea and air space is increasingly being challenged by non-military demand for resources. The military relies upon natural resources for its operations, (e.g., airspace for training and water for industrial processes), as well as for air to absorb emissions, water to absorb effluent discharge, and land to assure civilian safety from accidents. Constraints associated with nonmilitary resource use (limits on the nonmilitary impacts of military use, reduced availability of resources due to nonmilitary use, and delays due to compliance requirements) can force modifications to current and future military operations, and, as such, they should be considered as encroachments on operations. As a result, encroachment can be regarded as a threat to the current and future functionality of any activity or installation.

While the environmental considerations discussed in this section play a fundamental role in this stage of the process, they may also be applied to determining military value and setting constraints on the generation of scenario alternatives.

5.4.1 Encroachment issues

An encroachment restriction is only meaningful in terms of the impact it has on the current and future functionality. Therefore, analysis and data questions should focus on identifying the level of functionality possible given the encroachment. For example, whether or not live ordnance use is or could be supported is what is important, not whether or not urbanization or endangered species are present. Analysis need only concern itself with encroachment to the degree that it can be addressed to allow for increased functionality.

After identifying potential functions to be located at each base, we will also need to consider the aggregate resource needs of a lay-down configuration. For example, air quality may accommodate relocation of a Navy squadron or an Air Force squadron; however, it may not be able to accommodate both without an offsetting pollution reduction, either by the relevant installations or by others in the air shed (accomplished through a DOD purchase of emission permits).

Natural resource/environmental encroachment issues include:

1. Limits on air pollution
2. Limits on water pollution
3. Protection of habitat and listed species
4. Protection of migratory birds
5. Protection of marine mammals
6. Protection of wetlands
7. Controls on solid waste
8. Controls on hazardous waste
9. Controls on other toxic substances
10. Limits on water availability

Non-environmental encroachment issues include:

1. Restrictions (and complaints) on noise levels
2. Restrictions on civilian safety risk

   (a) De-confliction with nonmilitary air traffic
   (b) Restrictions on ordnance footprint support
   (c) Restrictions on ordnance carry and release
   (d) Restrictions on laser footprint support
   (e) Requirements for unexploded ordnance clearance

3. Protection of cultural/historic/archaeological resources

4. Limits on frequency availability and interference generation

Urbanization can create many of these encroachment issues in that it can increase noise level complaints and safety concerns, civilian air and land safety risks, the likelihood of frequency competition and interference, environmental quality risks (due to an increase in cumulative discharges and an increase in the exposed population), and the urgency of natural resource and species protection.

The functionality of an installation is a function of all the infrastructure and natural resources, on and off the installation premises, that contribute to and support operations, i.e., all resources that contribute to military value. Therefore, we are interested in the availability of all these resources. For example, on base, both the runway length and hangar space are important, while off base, the air installation compatibility use zones (AICUZ) and proximity to and characteristics of training areas are also important.

In general, we want to know:

- What resources are currently available with respect to an installation?
- What resources will be available in the future with respect to an installation?
- What resources are currently threatened by nonmilitary factors?
- What resource expansion opportunities exist?
  - Through current initiatives at the installation, or in the region, Service, DoD, local community, state, or nation?
  - Through other opportunities (e.g., expenditures)?

Encroachment is simply another category of resource restrictions, such as budgets and budgetary processes, manpower, time (personnel related, training time related), ordnance availability, equipment and parts availability, etc., and can be treated the same as any other restriction on the available physical resources. Furthermore, restrictions within and across categories do not affect operations equally; some may not affect operations at all. Therefore, we need to address encroachment only to the extent that it may impede functionality.

There are three ways that encroachment can affect the functionality of an installation:

1. Restrict current operations and training

2. Threaten the resources currently available, and therefore threaten future operations and training

3. Limit expansion and realignment opportunities

We need to identify resource requirements, e.g., ships need piers with particular specifications and, possibly, a nearby shipyard. We need to be sure to think about all the resources that are needed by operations. For instance, aircraft carrying live ordnance need flight paths over uninhabited land or over water both on their way out of and into the installation. Some may be environmental resources, such as air quality for emissions and undeveloped land for AICUZ. A comprehensive list of resource requirements for a function should include all the resources that could be encroached upon.

- Identify currently available resources of value on- and off-base, i.e. the resources required by functions.

- Identify limiting factors to resource use. Some will be encroachment issues, some will not.

- Identify current resource control. DoD does not own all the resources used by operations (e.g., airspace, AICUZ). DoD has varying levels of control over resource access and use and, therefore, varying levels of exposure to potential loss of access and use.

- Identify threats to resources, such as development patterns and new species likely to receive state or federal protection, and their likely impact on resource availability.

- Identify opportunities (and costs) for abating threats

The 1995 BRAC methodology provided no insight into what encroachment really means. The encroachment data collected in the BRAC 95 data calls impose a very rigid interpretation: not having encroachment problems is good, anything else is bad. In reality, not all resource restrictions are constraints on operations and not all constraints are significant. Many are just nuisances.

Appendix A

Data guidelines

The material in this appendix was provided in various forms to the JCSGs and to the DON BRAC analysts to help structure their thought processes on how to ask for data that would support the analytical process.

A.1 Background

BRAC analyses considers possible realignments of forces and support services among installations in order to reduce excess capacity and maintain military value while adhering resource constraints. There are three primary areas of data required:

- *Functions and functional requirements:* What are the functions we need to assign? How much of each function is to be assigned?
- *Resources and resource needs:* What resources may limit the capacity of a site to support functions? How much of each resource is used by a function?
- *Military value:* What factors determine the value of locating functions at a particular site?

These guidelines are not meant to limit your interests, nonetheless, they may help to offer some definitions and suggestions that will help ensure that data collected is useful for analysis. The key factors are:

- *Relevance to the base closure and realignment decision*: We are modeling closure and realignment decisions. Data not relevant to those decisions are not needed.

- *Quantitative:* In the end, numbers are better than essay questions. To the extent possible, find a way to ask for specific quantities or yes/no responses.

As a caveat, in what follows, we are thinking much more about closure and realignment than about any business re-engineering processes analysis that the JCSGs may choose to undertake.

A.2 Definitions

Functions, resources, capacities, requirements, and military value–all are important and can be closely related, but distinctions should be made to provide a common framework. We propose some definitions that might be useful as a starting point.

A.2.1 Functions

A function is a rational partition of an activity into identifiable groups of missions, outputs, or things that are substantial enough to be worth considering for relocation or closure. In practical terms, the functions considered should be big enough, in aggregate, that the choices to be made raise the possibility of facility closure.

There are any number of ways to partition military activities into functions. One way is to think of functions as a partition into product lines–for example, airframe repairs and engine overhauls. They can be thought of as outputs–for example, dental care and undergraduate pilot training. Another way to think of them is as a subordinate organization–for example, a technical center's laboratory and fabrication shop. The various types of ships and aircraft squadrons are an obvious partition of the primary forces. Examples of functions in the administrative and support area would be command headquarters and administrative headquarters.

There are both independent and follower functions. For example, finance and accounting services are independent–they can be located without much concern for where the activities it serves are located. Other functions, such as facility management, are follower–type functions–their locations and scale are dependent on the location of their customers. Local follower functions are least likely to be considered in the analysis.

A.2.2 Functional requirements

A defining factor for what we treat as a function is that a requirement is specified. Generally speaking, requirements are derived from the force structure. Will the force structure specifically define the supporting infrastructure required? Probably not. So, the determination of these requirements becomes part of the JCSG's analysis–determining how much of each function is needed to support the force structure. If a functional requirement is entirely derived from the local presence of other function, this function probably does not need to be separately considered. Surge requirements can be one component of overall requirements, if there is a need for flexibility to meet short-term demands.

A.2.3 Resources

Resources are the services or things at a facility that the functions need in order to operate. In practical terms, we will be concerned with resources that are not trivially expandable. Examples are buildings, utility systems, airfields, piers, and environmental permits. Considering resource use will be important because relocation may be limited by availability or may free up critical resources for other uses. It would seem to be the responsibility of the JCSGs to determine the resource needs for each function they define.

A.2.4 Capacity

Capacity may be defined in terms of potential throughput–how much of a function can be supported by the available resources. In practice, this may be a fairly simple measure derived from one key resource (for example, piers or office space). Alternatively, it may be a composite measure that reflects several resources and practical constraints on operations. Under these circumstances, measuring capacity may be very difficult. For a detailed discussion of how to address the more complicated situations, please see the discussion in chapter 2.

A.2.5 Military value

Military value should measure the suitability of the activity or location to support a function. It might include quality of life, availability of a suitable workforce, closeness to customers, and adequacy of the physical facility and local support structure. Keep in mind the long run perspective of BRAC. We are considering whether it is wise to permanently dispose of a facility.

From this perspective, the important factors are the enduring or permanent attributes of the location itself. The efficiency of the current workforce is perhaps better viewed as a short term concern–it can be fixed.

A.3 Suggestions

A.3.1 Functions

- *Choose substantial functions that are relevant to closure decisions.* Avoid partitioning into functions that are so fine as to have no likely relevance to base closure decisions. We will almost have to discard data on minor functions-those that do not use substantial, identifiable facilities-and local follower functions-those whose existence depends only on the local activities they support. Don't bother with functions that are too small to possibly influence any significant closure decision or that are sufficiently dependent that we can just consider them as closing if their associated base closes. At the same time, partition finely enough to distinguish between major functions using resources or equipment for which suitable substitutes do not exist.

- *Don't forget, we need to know the requirements?* If you cannot easily determine requirements, it may be because this is a local follower function that you do not need to consider.

A.3.2 Resources

- *Screen for relevance.* Is this resource realistically ever going to be a constraint on expanding throughput or use? Something like telephone service is probably so easily expanded that it need not be considered. On the other hand, buildings, land, runways, piers, and sewage treatment plants are big enough hurdles that they should be considered explicitly as resources.

- *Don't forget about resource needs.* For each function, someone has to determine how much of each resource is needed per unit of functional throughput, e.g., office space or water use.

- *Consider the potential expansion of resource.* Almost all resources are expandable at some cost. It is useful to get some sense of the realistic limits to expansion.

A.3.3 Capacity

- *Clear assumptions.* Make sure the assumptions about how capacity is to be measured are clearly stated. For example, if measuring throughput, should the activity assume a single shift, five days a week?

- *Clarify the constraints.* Ask capacity questions that will allow the analysts to understand the constraints on the capacity, e.g., work force limitations or other input resource constraints.

- *Expansion.* If capacity expansion opportunities are to be considered, then ask questions that allow an analyst to reasonably assess the expansion possibilities, including possibilities of trading functional capacity in one area for increased capacity in another. The collected data should identify practical limits on expansion potential.

A.4 General guidelines on preparing questions

1. *Know how the answer to the question will be used.* Do not ask a question because the answer "might" be useful. Asking a question should signify an expectation that the answers will be used somewhere in the BRAC process.

2. *Try to avoid essay questions.* All answers must be auditable. To the extent possible, an answer should be a number, a "Yes/No" response, or category membership. The essay question is best reserved for asking if there is anything important that was overlooked. We do need to be realistic enough to admit that we may not know everything and the field activities/facilities may have good ideas.

3. *Consider the likelihood the question can be answered.* Is the requested information readily available and, if not, is there a standard procedure that activities can follow to obtain it?

4. *Review the complete set of questions for balance.* Is each subject sufficiently addressed with a minimum number of questions?

5. *Ask questions that help distinguish between different activities.* There is no point in asking a question that all activities answer in approximately the same way. If a question does distinguish between activities, make sure the distinction matters.

6. *Make the assumptions clear.* Make sure all assumptions about how measurements are to be derived are clearly stated in the question.

Appendix B

Depot example

For this example, we have seven depots that repair four types of products: airframes, tanks, turbines, and electronics. The future requirement for annual production for each product type is discussed in 2.3.2.

There are four resource types needed by the depots depending on the products produced at the depot. These resource types include test ranges, fabrication shops, hangars, and test facilities. The quantities, or resource capacities, of each of these resource types available at each depot are shown in table B.1. These resource capacities may be expanded by the amounts shown in table B.2

The resources required by a depot to produce one unit of each product type are shown in table B.3.

Military value can be measured and incorporated at the activity level or at the functional level. Table B.4 specifies the military values of these depots as well as the military values we have associated with each depot/product combination. A blank in the table for a depot/product combination indi-

Activity	Test range	Fabrication shop	Hangars	Test facility
Alpha	2	1.2	12	0.9
Bravo	1	0.9	7	1.3
Charlie	1	1.6	3	2.3
Delta	2	2.1	0	1.7
Echo	1	3	0	0.7
Foxtrot	2	1.7	0	2.4
Golf	0	0	0	1.8

Table B.1: Depot resource capacities

Activity	Test range	Fabrication shop	Hangars	Test facility
Alpha	0	0.1	3	0.5
Bravo	0	0.1	2	0.6
Charlie	0	0.1	1	0.7
Delta	0	0.2	0	0.4
Echo	0	0.3	0	0.6
Foxtrot	0	0.1	0	0.6
Golf	0	0	0	0.4

Table B.2: Depot resource capacity expansion possibilities

Product	Test range	Fabrication shop	Hangars	Test facility
Air frames	0.02	0.01	0.37037	0.002326
Tanks	0.01	0.058824	0	0.004673
Turbines	0	0.006667	0	0.003003
Electronics	0	0	0	0.000222

Table B.3: Resources required per unit of production

cates that that product will not be assigned to that depot. Here we can see how the activity/functional decision may affect the results. Activity Golf has a low military value because it only does electronics, even though it has the highest electronics military value.

Table B.4 also shows the importance of each product type. Whenever the product military values are used, they will be multiplied by these *importance* factors.

Activity	Activity MV	Air frames MV	Tanks MV	Turbines MV	Electronics MV
Alpha	62	82		35	57
Bravo	61	50		62	89
Charlie	67	66		81	80
Delta	69		75	73	64
Echo	63		93	44	74
Foxtrot	75		54	54	85
Golf	55				92
Averages	64.57	67.13	74.00	62.28	79.30
Importance		2	2	1	1

Table B.4: Depot military values

Appendix C

DoN example

We have constructed a notional DoN laydown of 18 installations and 23 hosted activities that we use in chapter 4 to demonstrate our methodology as applied to a DoN example. Figure C.1 shows the notional laydown for these installations and activities. The figure also shows five notional aviation training ranges.

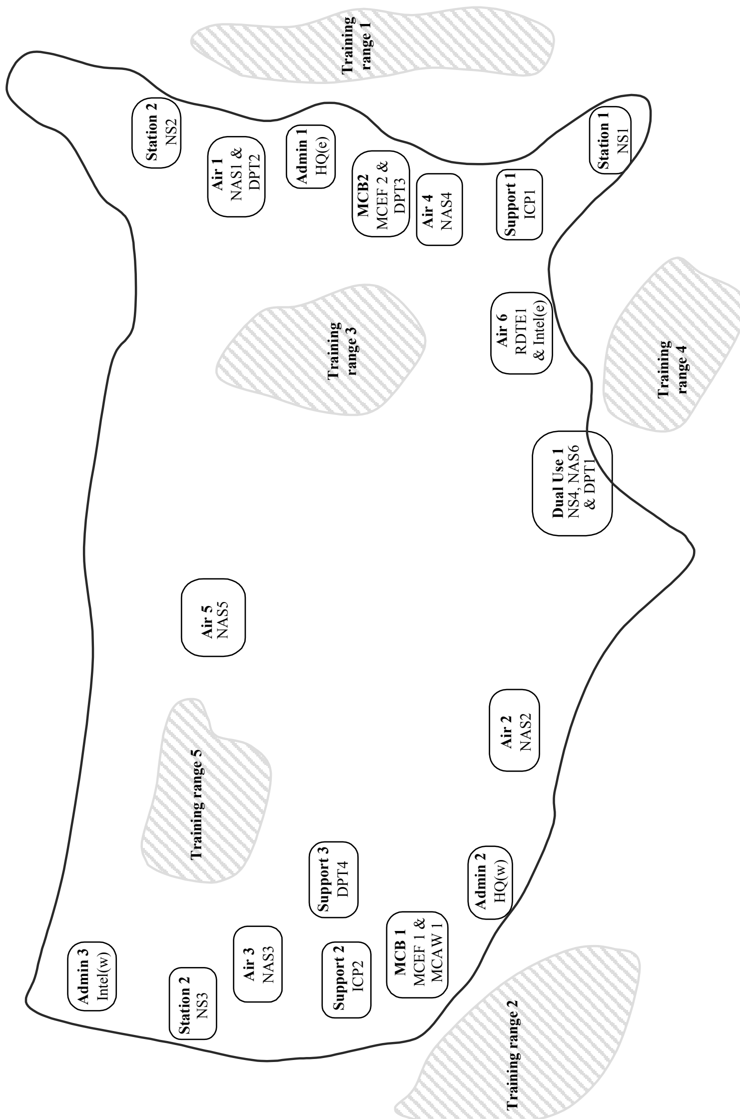

Figure C.1: Laydown of sites and activities

There are five training ranges that may be used by the activities. The types of training that may be done on each range are shown in table C.1.

	Type of training allowed			
Training range	Air-to ground	Training route	Special use	Instrumented
1		X	X	X
2		X	X	
3	X			
4		X		
5	X	X	X	X

Table C.1: Training range capabilities

We have divided the 18 installations or sites into six categories as described in table C.2.

Site types	Site names					
Air station	Air 1	Air 2	Air 3	Air 4	Air 5	Air 6
Station	Station 1	Station 2	Station 3			
Combined station	Dual 1					
Marine Corps base	MCB 1	MCB 2				
Support	Support 1	Support 2	Support 3			
Admin site	Admin 1	Admin 2	Admin 3	Admin 4		

Table C.2: Site categories and names

The activities assigned to each site are given in table C.3. We distinguish between air station site and the naval air station activity so that we may consider the naval air station as an activity with hosting operational, training, or test and evaluation squadrons as its major function. We apply the same reasoning to naval stations and the other activity types used in this example.

The force structure requirements for the operational, training, and test and evaluation unit types are given in table C.4. These are the number of units of each indicated type that must be placed at activities in any feasible solution.

The depot activities perform four types of repair work: air frames, tanks, turbines, and electronics. The total annual repair quantities for each type of repair work is shown in table C.5.

Site	Activity names		
Air Station 1	NAS 1	Depot 2	
Air Station 2	NAS 2		
Air Station 3	NAS 3		
Air Station 4	NAS 4		
Air Station 5	NAS 5		
Air Station 6	RDT&E 1	Intel East	
Dual 1	NAS 6	NS 4	Depot 1
Station 1	NS 1		
Station 2	NS 2		
Station 3	NS 3		
Support 1	ICP 1		
Support 2	ICP 2		
Support 3	Depot 4		
Admin 1	HQ East		
Admin 2	HQ West		
Admin 3	Intel West		
MCB 1	MCEF 1	MCAW 1	
MCB 2	MCEF 2	Depot 3	

Table C.3: Site activities

There are four headquarters/administrative units that must be accommodated in any feasible solution. These units are summarized in table C.6.

C.1 Resources

Before an activity can perform a function, it must have the resources required to perform that function. Table C.7 shows the resource types we use for this example. Table C.7 also shows the shorthand names that we employ for each resource type. Some of the resource types are not very realistic, but they are useful for demonstrating the capabilities of the model.

The number of annual flight evolutions and the number of hangars available at each naval air station activity are shown in table C.8.

Resources available at each naval station activity for hosting ships are shown in table C.9. We have not specified units for the Ship Intermediate Maintenance Activity (SIMA) resource, but it can be thought of as an annual workload limit.

Operational entities	Force Structure requirement
CVN	5
CG	15
VF	10
VX	2
VT	4
BTLN	5
MCWING	1

Table C.4: Operational, training, and test and evaluation force structure requirements

Repair products	Annual requirement
Air frames	15
Tanks	12
Turbines	100
Electronics	1,600

Table C.5: Depot-level repair requirements

Table C.10 shows some of the resources needed for performing depot functions available at each of the four depots. Other general resources available at the depots are given in table C.11.

Table C.11 shows more resources available at the activities. P10 refers to the amount of particulate emissions the activity is licensed to emit.

Support units	Requirement
ICP	2
SEA HQ	1
C3 HQ	1

Table C.6: Support requirements

Resource types	Short name
Number of CVN berth	CVN Berth
Number of CG berths	CG berths
Maximum annual SIMA workload	SIMA
Maximum annual flight evolutions	Flight evolutions
Number of hangars	Hangars
Annual test facility capacity	Test facility
Outdoor storage space	Storage
Office space	Space
Total water available	Water
P10	Emissions allowed
Maximum annual range utilization	Range
Maximum annual fabrication capacity	Fabrication

Table C.7: Types of resources

Activity	Flight evolutions	Hangars
NAS 1	15,000	5
NAS 2	25,000	6
NAS 3	25,000	6
NAS 4	25,000	7
NAS 5	15,000	4
NAS 6	15,000	4
MCAW 1	10,000	4
RDT&E 1	20,000	6

Table C.8: Available aviation activity resources

C.2 Resource expansion

Table C.12 shows the additional resources that could be made available for a reasonable investment at each of the aviation activities.

Table C.13 shows the additional resources that could be made available at each of the naval stations.

The additional resources that could be made available at each of the repair or supply activities is shown in table C.14.

The additional general resources that could be made available at each of the activities for a reasonable investment are shown in table C.15.

Activity	CVN berths	CG berths	SIMA
NS 1	0	8	350
NS 2	2	4	275
NS 3	3	3	500
NS 4	2	6	400

Table C.9: Available ship station activity resources

Activity	Fabrication shop	Hangars
Depot 1	1.2	4
Depot 2	0.9	3
Depot 3	2.1	.
Depot 4	3.0	.
ICP 1	0.5	.
ICP 2	0.3	.

Table C.10: Available repair activity resources

Activity	Test facility	Storage	Space	Water	p10	Range
NAS 1	0.2	60	22	3.0	2.5	0.2
NAS 2	.	40	25	2.5	2.0 0.2	
NAS 3	0.3	80	40	3.5	2.5 0.3	
NAS 4	.	30	20	1.5	2.0	0.3
NAS 5	.	15	85	2.0	0.7	1.5
NAS 6	0.5	35	30	1.5	0.8	0.8
RDT&E 1	0.5	50	35	2.3	3.0	1.5
MCAW 1	.	20	15	1.5	1.5	0.2
MCEF 1	.	20	25	1.2	0.8	3.0
MCEF 2	.	35	25	2.5	1.0	3.0
NS 1	.	40	25	4.0	0.7	0.2
NS 2	.	50	50	4.0	0.7	.
NS 3	.	40	25	5.5	0.7	0.2
NS 4	.	35	70	3.5	0.5	0.2
Depot 1	1.0	50	6	3.5	1.2	0.4
Depot 2	1.5	50	8	3.0	1.6	0.2
Depot 3	1.7	50	7	3.0	1.5	0.4
Depot 4	0.7	50	7	3.0	1.5	0.2
ICP 1	0.8	90	10	.	0.8	.
ICP 2	0.9	90	9	.	0.5	.
HQ West	.	.	50	0.2	.	.
HQ East	.	.	60	0.2	.	.
Intel East	.	.	30	0.2	.	.
Intel West	.	.	50	0.2	.	.

Table C.11: Available general activity resources

Activity	Flight evolutions	Hangars	Fabrication shop
NAS 1	2,500	2	0.1
NAS 2	3,500	2	0.1
NAS 3	3,500	2	0.1
NAS 4	3,500	2	0.1
NAS 5	2,500	2	0.1
NAS 6	2,500	2	0.1
MCAW 1	5,000	2	0.1
RDT&E 1	5,000	2	0.1

Table C.12: Maximum additional aviation activity resources

Activity	CVN berths	CG berths	SIMA	Fabrication shop
NS 1	1	2	25	0.1
NS 2	1	2	25	0.1
NS 3	1	2	25	0.1
NS 4	1	2	25	0.1

Table C.13: Maximum additional ship station activity resources

Activity	Fabrication shop	Hangars
Depot 1	0.2	1
Depot 2	0.2	1
Depot 3	0.4	.
Depot 4	0.5	.
ICP 1	0.1	.
ICP 2	0.1	.

Table C.14: Maximum additional repair activity resources

Activity	Test facility	Storage	Space	Water	p10
NAS 1	0.2	30	25	0.25	0.2
NAS 2	0.2	30	25	0.25	0.2
NAS 3	0.2	30	25	0.25	0.2
NAS 4	0.2	30	25	0.50	0.2
NAS 5	0.2	30	25	0.50	0.2
NAS 6	0.2	30	25	0.20	0.2
RDT&E 1	0.2	30	25	0.25	0.2
MCAW 1	0.1	15	20	0.20	0.2
MCEF 1	0.1	20	24	0.20	0.2
MCEF 2	0.1	25	24	0.20	0.2
NS 1	0.2	30	25	0.25	0.2
NS 2	0.2	30	25	0.25	0.2
NS 3	0.2	30	25	0.25	0.2
NS 4	0.2	30	25	0.20	0.1
Depot 1	0.2	10	5	0.25	0.2
Depot 2	0.3	10	5	0.25	0.2
Depot 3	0.2	10	5	0.25	0.2
Depot 4	0.1	10	5	0.20	0.2
ICP 1	0.1	15	5	.	0.2
ICP 2	0.1	15	5	.	0.2
HQ West	.	.	10	.	.
HQ East	.	.	10	.	.
Intel East	.	.	5	.	.
Intel West	.	.	5	.	.

Table C.15: Maximum additional general activity resources

Ship class	CVN berths	CG berths	SIMA	Space	Water
CVN	1	0	15	3	0.30
CG	0	1	10	2	0.15

Table C.16: Annual resource consumption by ships

Aviation entity	Flight evolutions	Hangars	Space	Water	P10
VF	3,000	1	2	0.2	0.12
VX	1,500	1	5	0.2	0.12
VT	4,000	1	5	0.2	0.16
Wing	9,000	3	7	0.5	0.30

Table C.17: Annual resource consumption by aviation units

C.3 Resource consumption by function

In this section, we describe the annual amount of each resource type required per unit of functional output. Table C.16 shows the resources required to host each CVN and each CG at a naval station.

The annual consumption of resources by each aviation unit is shown in table C.17. P10 indicates the number of particulate emission licenses needed per hosted squadron.

Table C.18 shows the resources required for each hosted Marine Corps ground unit.

The annual resource consumption for each unit of depot or supply production is shown in table C.19.

Annual resource consumption for each of the support functions is shown in table C.20.

MC ground entity	Space	Water	Range
BTLN	7	0.1	1.0

Table C.18: Annual resource consumption by MC ground units

Function	Test facility	Fabrication shops	Storage	Range	Space	Water	P10
Air frame	0.0005	0.1500	0.220	0.0040	0.0200	0.00500	0.002000
Tank	.	0.0080	0.3000	0.001	0.0240	0.00600	0.002500
Turbines	0.0010	.	0.0300	.	0.0025	0.00060	0.000250
Electronics	0.0001	.	0.0015	.	0.0002	0.00005	0.000015
ICP	0.2000	.	60.0000	.	10.0000	0.05000	.

Table C.19: Annual resource consumption for depot and supply functions

Function	Space	Water
ONI	15	0.05
SEA HQ	24	0.05
C3 HQ	10	0.05

Table C.20: Annual resource consumption for support functions

C.4 Minimum and maximum constraints

Table C.21 shows the minimum requirement for runway length for each type of aviation squadron. The table also indicates that the ONI function has a minimum requirement for 6 gigaflops of computational capability.

Corresponding to the minimum requirements for runway length and computing power given in table C.21, table C.22 shows the availability of these resources at each activity.

The maximum allowed distance to ranges with each required capability for each aviation unit type are shown in table C.23.

	Resource	
Function	Runway length	GFlops
VF	8,000	.
VX	9,000	.
VT	7,000	.
Wing	8,000	.
ONI	.	6

Table C.21: Minimal functional resource requirements: runway length and GFlops

The distances to the closest range with each required capability for each activity is shown in table C.24.

C.5 Military values

The military values for each activity/function combination are presented in this section. We begin with the naval air stations and the RDT&E site as shown in table C.25. We have assigned military values for all functions at each activity regardless of whether of not the activity is currently performing that function.

Next, table C.26 shows the military values for each naval station for all functions.

Table C.27 shows the military values for performing all of the functions at each of the Marine Corps activities in the example.

The military values for performing all of the functions at each repair and supply activity are shown in table C.28.

The remaining military values for performing all of the functions at each of the support activities are shown in table C.29.

	Resource	
Function	Runway length	GFlops
NAS 1	10,000	8
NAS 2	12,000	10
NAS 3	12,000	3
NAS 4	12,000	5
NAS 5	10,000	11
NAS 6	12,000	5
RDT&E 1	12,000	15
MCAW 1	8,500	3
MCEF 1	.	3
MCEF 2	.	4
NS 1	.	7
NS 2	.	3
NS 3	.	6
NS 4	.	4
Depot 1	.	5
Depot 2	.	2
Depot 3	.	2
Depot 4	.	2
ICP 1	.	7
ICP 2	.	6
HQ West	.	3
HQ East	.	1
Intel East	.	12
Intel West	.	15

Table C.22: Activity runway length and GFlops availability

	Range type			
Function	Special use	Air-to-ground	Training route	Instrumented
VF	1,110	1,100	.	.
VX	1,000	1,000	1,000	.
VT	1,100	1,100	.	1,100
Wing	1,100	1,100	.	.

Table C.23: Maximum allowed distance to ranges by type

	Range type			
Function	Special use	Air-to-ground	Training route	Instrumented
NAS 1	534	468	534	534
NAS 2	622	1,090	595	1,090
NAS 3	394	508	394	
NAS 4	414	521	414	414
NAS 5	421	421	421	421
NAS 6	1,036	729	321	1163
RDT&E 1	919	321	414	919
MCAW 1	400	1050	900	1050

Table C.24: Activity distance to ranges by type

	Naval air station						
Function	NAS 1	NAS 2	NAS 3	NAS 4	NAS 5	NAS 6	RDT&E 1
CG	48.67	52.82	53.55	51.08	34.99	58.28	21.98
CVN	52.05	62.38	62.99	48.49	38.93	57.99	24.26
VF	59.73	51.78	68.00	70.01	61.44	62.36	69.03
VX	45.67	50.96	58.74	65.65	46.88	53.32	55.99
VT	56.83	42.33	61.78	69.36	59.99	57.16	68.86
Air frame	40.72	20.76	14.31	8.99	6.46	48.06	17.16
Tank	20.72	20.76	14.31	8.99	6.46	15.06	17.16
Turbine	52.72	20.76	14.31	8.99	6.46	33.06	17.16
Electronics	58.72	20.76	14.31	8.99	6.46	37.06	17.16
ICP	18.22	25.26	22.81	16.68	15.06	35.02	20.24
ONI	39.94	42.09	11.18	25.66	46.98	40.62	59.40
SEA HQ	13.31	26.21	28.07	26.24	45.14	37.90	30.04
C3 HQ	13.31	26.21	28.07	26.24	45.14	37.90	30.04
BTLN	39.00	36.00	32.00	31.00	28.00	30.00	33.00
Wing	45.00	43.00	49.00	51.00	47.00	41.00	48.00

Table C.25: Functional military values for naval air stations

	Naval station			
Function	NS 1	NS 2	NS 3	NS 4
CG	74.96	62.19	80.03	73.28
CVN	67.84	63.23	82.62	72.99
VF	18.02	37.96	39.82	50.36
VX	22.47	41.36	34.47	41.32
VT	25.01	41.90	41.90	45.16
Air frame	21.80	25.31	16.97	48.06
Tank	21.80	25.31	16.97	15.06
Turbine	21.80	25.31	16.97	33.06
Electronics	21.80	25.31	16.97	37.06
ICP	20.29	22.51	15.98	35.02
ONI	28.03	20.88	17.12	40.62
SEA HQ	11.73	39.87	5.90	37.90
C3 HQ	11.73	39.87	5.90	37.90
BTLN	35.00	26.00	22.00	20.00
Wing	20.00	28.00	30.00	32.00

Table C.26: Functional military values for naval stations

	Marine Corps activity		
Function	MCEF1	MCEF2	MCAW1
CG	62.19	23.86	62.19
CVN	63.23	24.34	63.23
VF	37.96	25.94	37.96
VX	41.36	25.46	41.36
VT	41.90	29.59	41.90
Air frame	25.31	30.72	25.31
Tank	25.31	58.72	25.31
Turbine	25.31	32.72	25.31
Electronics	25.31	52.72	25.31
ICP	22.51	45.73	22.51
ONI	20.88	19.81	20.88
SEA HQ	39.87	28.76	39.87
C3 HQ	39.87	28.76	39.87
BTLN	64.00	62.00	42.00
Wing	58.00	52.00	45.00

Table C.27: Functional military values for Marine Corps activities

Function	Repair or supply activity					
	Depot 1	Depot 2	Depot 3	Depot4	ICP 1	ICP 2
CG	23.85	22.86	23.86	23.86	12.72	10.87
CVN	19.31	20.34	24.34	24.34	7.09	6.80
VF	13.50	12.94	25.94	25.94	6.31	6.24
VX	21.38	24.46	25.46	25.46	15.57	19.95
VT	10.03	9.59	29.59	29.59	4.81	4.75
Air frame	82.00	50.00	10.00	5.00	37.81	43.40
Tank	5.00	5.00	75.00	93.00	37.81	43.40
Turbine	35.00	62.00	73.00	44.00	37.81	43.40
Electronics	57.00	89.00	64.00	74.00	37.81	43.40
ICP	43.36	45.73	45.73	45.73	78.28	79.47
ONI	13.59	16.81	19.81	19.81	31.34	19.91
SEA HQ	17.41	25.76	28.76	28.76	20.60	19.48
C3 HQ	17.41	25.76	28.76	28.76	20.60	19.48
BTLN	15.00	15.00	15.00	15.00	10.00	10.00
Wing	15.00	15.00	15.00	15.00	7.00	8.00

Table C.28: Functional military values for repair or supply activities

Function	Support activity			
	HQ East	HQ West	Intel East	Intel West
CG	10.63	13.34	8.72	8.65
CVN	9.41	12.02	7.63	6.37
VF	3.09	3.63	2.05	1.69
VX	11.43	13.02	6.35	3.48
VT	2.37	2.81	1.59	1.30
Air frame	14.86	3.93	13.33	8.62
Tank	14.86	3.93	13.33	8.62
Turbine	14.86	3.93	13.33	8.62
Electronics	14.86	3.93	13.33	8.62
ICP	17.84	11.08	14.83	11.85
ONI	34.89	11.21	51.18	47.76
SEA HQ	53.82	42.99	41.67	27.62
C3 HQ	42.99	53.99	27.62	41.67
BTLN	10.00	10.00	15.00	10.00
Wing	10.00	10.00	7.00	10.00

Table C.29: Functional military values for support activities

Appendix D

Military value calculations

A fundamental part of assessing military value is determining the weight to be given to each question or characteristic measured. This appendix presents a formal description of how these weights may be determined.

For each class or group of activities decision makers should weight the military value criteria as follows. Let there be a set, I, of military value criteria. For BRAC 95 these roughly corresponded to readiness, facilities, mobilization, and costs. Let w_i be the weight assigned to criteria $i \in I$. The following must hold: $w_i \geq 0$ and $\sum_{i \in I} w_i = 100$.

In addition, the decision makers should specify a set of attributes and weights for each criteria. If the class is air stations, then an attribute for the readiness criteria might be runways. Another attribute might be hangars. LetJ_i be the set of attributes associated with criteria i and let a_{ij} be the weight assigned by the decision makers to attribute $j \in J_i$. We must have $a_{ij} \geq 0$ for all $i \in I$ and $j \in J_i$. In addition, we must again have $\sum_{j \in J_i} a_{ij} = w_i$ for each $i \in I$ so that the weight for a criteria is spread across the attributes for that criteria. With this scheme there will be a total of 100 points available for each activity or activity/function combination that is assessed for military value.

Now, let us introduce the questions or characteristics into the calculation. Let K be the set of questions or characteristics used to assess military value for this class or group of activities. For question $k \in K$, the decision makers will give the question or characteristic a *score*, s_k, from one to ten indicating the relative strength of the question to other questions. The decision makers will also decide to what subset of the criteria/attribute combinations the question is relevant. We let ϕ_{ijk} be an indicator variable that equals 1 if question k is relevant to attribute j of criteria i and 0 otherwise.

Let V_k be the total weight associated with question k and let v_{ijk} be the weight that question k derives from attribute ij so that $V_k = \sum_{ij} v_{ijk}$. The weight that question or characteristic k derives from attribute ij is given by

$$v_{ijk} = a_{ij} s_k \phi_{ijk} / \sum_{l \in K} s_l \phi_{ijl}.$$

Why does this work? First, note that

$$\begin{aligned} \sum_{k \in K} v_{ijk} &= \sum_{k \in K} a_{ij} s_k \phi_{ijk} / \sum_{l \in K} s_l \phi_{ijl} \\ &= a_{ij} \sum_{k \in K} s_k \phi_{ijk} / \sum_{l \in K} s_l \phi_{ijl} \\ &= a_{ij} \end{aligned}$$

so that all of the questions or characteristics associated with attribute ij contribute exactly a_{ij} weight to the total as desired.

These relationships are fairly straightforward to implement within a spreadsheet. Variations of this methodology were used by the DON IAT and the JCSGs during BRAC 2005.

Appendix E

Mathematical formulations for generating alternatives

We present the basic formulation of the optimization methodology described in this handbook in this appendix. We also show how we have added the capability to model expansion possibilities and the modifications needed to generate the second- and third-best solutions.

E.1 Basic mathematical formulation for the optimization methodology

Our model is a mixed integer programming (MIP) problem. A MIP is a specialized form of a linear program. We begin by defining the basic parameters of the model. We then discuss the objective functions that we maximize. Finally, we describe the constraints that any solution to the MIP must satisfy.

E.1.1 Model parameters

The basic entities for this model are activities or installations, the functions performed at those activities or installations, and the resources available at the activities or installations that are needed to perform the functions. Each functional area has a requirement across all activities or installations that must be met.

$$
\begin{aligned}
A &= \text{The set of activities or installations} \\
F &= \text{The set of functional entities that must be assigned} \\
R &= \text{The set of resource types that functions require} \\
d_i &= \text{The total requirement for function } i \in F \\
k_{jm} &= \text{The amount of resource } m \in R \text{ available at activity } j \in A \\
z_{jim} &= \text{Quantity of } m \in R \text{ required to produce a unit of } i \in F \\
&\quad \text{at activity } j \in A \\
V_j &= \text{The military value of activity } j \in A \\
v_{ji} &= \text{The military value of performing function } i \in F \\
&\quad \text{at activity } j \in A
\end{aligned}
$$

The parameter, z_{jim}, specifies how much of resource type m is consumed at activity j to produce one unit of output for function i. The military values are determined external to the model. For any given run of the model, only one set of military values will be used. The choice will depend on the focus of the model as discussed in chapter 4.

We use three sets of decision variables–one corresponding to retaining or eliminating an activity or installation, one corresponding to retaining or eliminating a function at an activity or installation, and one for the allocation of functional requirements to activities or installations:

$$
\begin{aligned}
O_j &= 1 \text{ if activity } j \in A \text{ is open and 0 otherwise} \\
T_{ij} &= 1 \text{ if function } i \in F \text{ is performed at activity } j \in A \text{ else } 0 \\
X_{ij} &= \text{The amount of function } i \in F \text{ to be performed at } j \in A
\end{aligned}
$$

E.1.2 Objective functions

In general, if cost were not an issue, DoD and the Military Departments would keep all activities and installations since they all add value. Some activities or installations have more value as indicated by the military values given to each one. Reducing infrastructure while maintaining high military value should be the goal of the optimization model. This is a classic case of a trade-off analysis. In this case, DoD or the Military Departments must trade military value for infrastructure reduction. We incorporate this trade-off in our model by using an objective function that includes total retained military value minus a penalty for retained infrastructure. For example, if we let ρ_s be the penalty parameter for the number of activities retained, then we would maximize the following objective function:

$$f_1(O,T) = \sum_{j \in A} V_j O_j - \rho_s \sum_{l \in A} O_l$$

If ρ_s is set to 0, then all activities are retained. If ρ_s is set to a very large value, then the minimum number of activities required to perform the functions and having the highest total military value would be selected. Intermediate values for ρ_s will give solutions that lie between these two extremes. Note that this objective function corresponds to method 1 of chapter 4. The efficient frontier discussed in chapter 4 is derived by varying ρ_s.

Instead of, or in addition to, penalizing the number of activities retained, we may wish to penalize the resources retained. If we let ρ_r be the penalty parameter for retaining resources, we have the following objective function:

$$f_2(O,T) = \sum_{j \in A} V_j O_j - \rho_r \sum_{m \in R} \sum_{l \in A} k_{lm} O_l$$

If certain resource types are more precious, requiring a higher penalty on their retention, then we can use individual penalty parameters for each resource type. Note that this objective function corresponds to method 2 of chapter 4.

If we focus on the military value of performing a function at an activity or installation, instead of activity military value, the the objective function that penalizes retained sites would be:

$$f_3(O,T) = \sum_{j \in A,\, i \in F} v_{ji} T_{ji} - \rho_s \sum_{l \in A} O_l$$

Note that if $O_l = 0$, a constraint that is discussed below will guarantee that $T_{il} = 0$ for all i, i.e., if an activity is closed, no functions are performed there. This objective function corresponds to method 3 of chapter 4.

Finally, if we wish to focus on the military value of performing functions at activities or installations while penalizing the retention of resources, we have the following objective function corresponding to method 4 of chapter 4:

$$f_4(O,T) = \sum_{j \in A,\, i \in F} v_{ji} T_{ji} - \rho_r \sum_{m \in R} \sum_{l \in A} k_{lm} O_l$$

Next we address the minimum set of constraint functions that each model must have.

E.1.3 Constraint functions

The first constraint we must have makes sure that the functional requirements are met across the retained activities:

$$\sum_{j \in A} X_{ji} \geq d_i \text{ for all } i \in F$$

Now, let M be a very large number. We want to make sure that if an activity or installation is not retained, that no functions may be performed there:

$$\sum_{i \in F} T_{ji} \leq MO_j \text{ for all } j \in A$$

We want to make sure that no activities or installations are retained if no functions are performed there:

$$O_j \leq \sum_{i \in F} T_{ji} \text{ for all } j \in A$$

We cannot assign functional workload to an activity or installation that is not permitted to perform that function:

$$X_{ji} \leq MT_{ji} \text{ for all } j \in A \text{ and } i \in F$$

The resources needed to perform the functions assigned to an installation or activity cannot exceed the resources available at an installation or an activity:

$$\sum_{i \in F} X_{ij} z_{jim} \leq O_j k_{jm} \ \forall\, j \in A \text{ and } \forall\, m \in R \tag{E.1}$$

The four objective functions and these constraints constitute the basic optimization model formulation.

E.2 Modeling expansion possibilities

We let $\hat{k}_{jm}$ be an allowable expansion of resource type m at activity j and we create a new variable U_{jm} to be the amount of resource type m that will actually be added to activity j. We must modify constraint (E.1) to be

$$\sum_{i \in F} X_{ij} z_{jim} \leq O_j k_{jm} + U_{jm} \ \forall\, j \in A \text{ and } \forall\, m \in R.$$

We must constrain the added capacity to be less than maximum expansion allowed:

$$U_{jm} \leq \hat{k}_{jm} \; \forall j \in A \text{ and } \forall m \in R.$$

We need constraints to make sure that the added capacities are not allowed at an activity that is not retained:

$$U_{jm} \leq MO_j \; \forall \, m \in R \text{ and } \forall j \in A$$

where, as before, M is a large value.

Since we do not want the added capacities to occur automatically in every run, we penalize the use of added capacity with another term added to each of the objective functions. We define a penalty parameter ρ_u and add the following to the objective function:

$$-\rho_u \sum_{j \in A, m \in R} U_{jm}.$$

If ρ_u is set to a large value, no additional capacity will be allowed.

Finally, we must modify the penalty terms for retained resources in objective functions $f_2(.,.)$ and $f_4(.,.)$ to include the new U_{jm} variables.

E.3 Generating the second- and third-best solutions

We use figure E.1 to describe our method for generating the second- and third-best solutions for a single instance of an optimization problem. There are eight activities, labelled A through H, in this illustration that may either be open or closed in a solution. The optimal solution retains activities A, B, and C. We let S_1 be the set of retained activities in the best solution. We will let S_2 represent the set of activities retained in the second-best solution for this optimization problem. In figure E.1, S_2 would include activities C, D, E, and F. Activities G and H are retained in neither the best or second-best solution. The second-best solution is obtained by solving the original optimization problem with additional constraints, that we will describe shortly, that will exclude the best solution. In a similar fashion, we find the third-best solution by solving the original optimization problem with additional constraints that will exclude the best and second-best solutions from consideration.

In addition to S_1, the set of retained activities in the optimal solution, and S_2, the set of retained activities in the second-best solution, we will have need of the following sets:

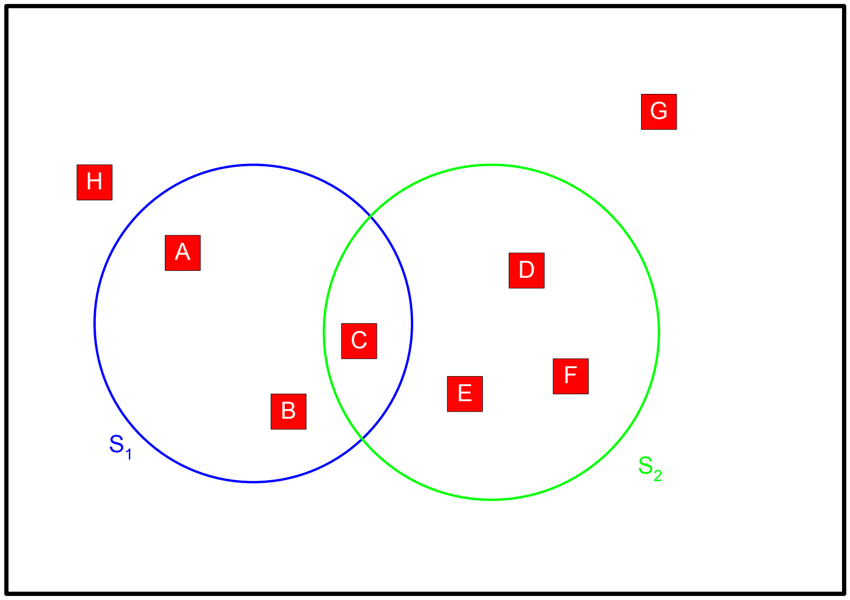

Figure E.1: Solution sets

- $S_1 \cap S_2$–the set of activities that are retained in both the best and the second-best solutions. Note that if we have not yet solved for the second-best solution, then $S_2 = \emptyset$, the empty set.

- S_I–if we have found the second-best solution, $S_I = S_1 \cap S_2$, the intersection of the two sets of retained activities, i.e., the activities retained in the best and second-best solutions. If we have not yet solved for the second-best solution, then $S_I = S_1$.

- $S_1 - S_2$–the set of activities that are retained in the best solution, but not in the second-best solution. If we have not yet solved for the second-best solution, then $S_2 = \emptyset$ and $S_1 - S_2 = S_1$.

- $S_2 - S_1$–the set of activities retained in the second-best solution, but not retained in the best solution. If we have not yet solved for the second-best solution, then $S_2 = S_2 - S_1 = \emptyset$.

- $\widehat{S_1 \cup S_2}$–the complement of the set of activities that are retained in either the best or second-best solutions. This is the set of activities not retained in either the best or second-best solution. If we have not yet solved for the second-best solution, then $\widehat{S_1 \cup S_2} = \widehat{S_1}$, the set of activities not retained in the best solution.

We will also have need of a parameter, $\eta = \max(0, |S_I| - 1)$, where $|S_I|$ is the number of activities in the set S_I.

We add the following five constraints to the basic model in order to generate the second- and third-best solutions. In these equations, O_s is binary variable that will be equal to one if installation or activity s is retained and zero otherwise.

$$\sum_{s \in S_I} O_s \leq \eta + 1 - \alpha \tag{E.2}$$

$$\sum_{s \in \widehat{S_1 \cup S_2}} O_s \geq \beta \tag{E.3}$$

$$\sum_{s \in S_1 - S_2} O_s \geq \gamma \tag{E.4}$$

$$\sum_{s \in S_2 - S_1} O_s \geq \gamma \tag{E.5}$$

$$\alpha + \beta + \gamma \geq 1 \tag{E.6}$$

where α, β, and γ are binary variables.

In the following discussion, we will assume that our problem is not degenerate, i.e., $S_1 \neq \emptyset$. If we have solved for the second-best solution, then $S_2 \neq \emptyset$.

If we have found the best solution and are solving for the second-best solution, then S_2 is the empty set and the following hold:

$$S_I = S_1 \tag{E.7}$$

$$\widehat{S_1 \cup S_2} = \widehat{S_1} \tag{E.8}$$

$$S_1 - S_2 = S_1 \tag{E.9}$$

$$S_2 - S_1 = \emptyset. \tag{E.10}$$

In this case, equations (E.6) and (E.10) imply that $\gamma = 0$. Equation (E.10) implies that equation (E.5) holds. Since $\gamma = 0$, we must have $\alpha + \beta \geq 1$ from equation (E.6). If $\alpha = 1$ then

$$\sum_{s \in S_1} O_s \leq \eta = |S_1| - 1$$

from equations (E.3) and (E.8) which implies that at least one of the installations or activities retained in the best solution is not included in the second-best solution. If $\beta = 1$, then from equations (E.4) and (E.9)

$$\sum_{s \in \widehat{S_1}} \geq 1$$

which implies that at least one installation or activity not retained in the best solution is retained in the second-best solution. In either case, the second-best solution must differ from the first solution and, since it is the solution to the original optimization problem with only these added constraints and variables, it must be the second-best solutions.

If we are solving for the third-best solution, then at least one of α, β, or γ must be equal to one. If $\alpha = 1$, then

$$\sum_{s \in S_I} \leq \eta = |S_1| - 1$$

so at least one installation or activity that was retained in both the best and second-best solutions is not retained in the third-best solution, i.e., the third-best solution differs from the first two.

If $\beta = 1$, then

$$\sum_{s \in \widehat{S_1 \cup S_2}} O_s \geq 1$$

so at least one installation or activity that was not retained in either the best or second-best solution is retained in the third-best solution.

If $\gamma = 1$, then

$$\sum_{s \in S_1 - S_2} O_s \geq 1 \text{ and } \sum_{s \in S_2 - S_1} O_s \geq 1$$

so that the third-best solution has at least one installation or activity that was retained in the best solution and not in the second *and* at least one installation or activity that was retained in the second-best solution and not in the best. This implies that the third-best solution must differ from both the best and the second-best solutions. This completes the proof that the addition of these five constraints and the three binary variables can be used to generate the second- and third-best solutions.

E.4 The BRAC 95 optimization methodology

For BRAC 95, the DoN minimized excess capacity subject to the retained installations having an average military value that was at least as great as the average for all of the installations under consideration. The basic formulation of this optimization problem is as follows:

$$
\begin{aligned}
\text{minimize} \quad & \sum_i O_i c_i \\
\text{subject to:} \quad & \sum_i O_i (V_i - AVG) \geq 0 \\
& \sum_i O_i c_i \geq K \\
& O_i \quad \text{binary for} \quad i = 1, \ldots, n
\end{aligned}
\tag{E.11}
$$

where $O_i = 1$ if installation or activity i is retained and $O_i = 0$ if it is not retained. The military value of installation i is V_i and it has capacity equal to c_i. K is the required capacity. AVG is the average military value across the n installations under consideration:

$$AVG = \frac{1}{n} \sum_i V_i \ .$$

Appendix F

AMPL code and data for the simple example

The AMPL (**A** **M**athematical **P**rogramming **L**anguage) code for the examples used in this handbook is included in this and the following two appendices. Reference [11] describes the AMPL language and reference [12] is a user's guide for the CPLEX solver.

F.1 Simple example model file

The AMPL files used to model the simple example from section 4.4 are included in this section.

```
######################################################################
# Optimization Methodology                                           #
# Simple Example                                                     #
#                                                                    #
# Ron Nickel                                                         #
# 12 January 2004                                                    #
######################################################################

set   ACTIVITIES;

param closed { ACTIVITIES } binary, default 1;
      # Use to force closures.

param MV { ACTIVITIES };    # Overall military value for each
                            # activity
```

```
param total_MV := sum { j in ACTIVITIES } MV[j];
      # Total military value across all activities.

param Avg_MV := total_MV/card( ACTIVITIES );

param capacity { ACTIVITIES } >= 0;
      # Capacity for each resource type at each activity

param total_capacity := sum { j in ACTIVITIES } capacity[j];
      # Total capacity across all activities.

param requirement >= 0;
      # Total requirement.

param constrain_avg_MV binary, default 0;
      # Set to 1 if you want to find the BRAC 95 solution.

param AVG := if constrain_avg_MV then Avg_MV else 0;


######################################################################
#  Sets used to find alternative solutions.  May be applied to either #
#  activities or functions or both.  Version for activities included  #
#  here.                                                              #
######################################################################

set   EXCLD1 within ACTIVITIES default {};
         # Exclude the best solution.

set   EXCLD2 within ACTIVITIES default {};
         # Exclude the next best solution.

set   EXCLD_INTER := if card (EXCLD2) > 0 then ( EXCLD1 inter EXCLD2 )
                           else EXCLD1;

set   EXCLD_1DIFF2 := EXCLD1 diff EXCLD2;
         # Activities in EXCLD1 but not in EXCLD2

set   EXCLD_2DIFF1 := EXCLD2 diff EXCLD1;
         # Activities in EXCLD2 but not in EXCLD1

set   EXCLD_COMPLEMENT := ACTIVITIES diff ( EXCLD1 union EXCLD2 );
         # Set of activities not in EXCLD1 or EXCLD2

param excld_num := max( 0, card( EXCLD_INTER ) - 1 );
```

```
######################################################################
# Parameters used to control the optimization                        #
######################################################################

param rho_number  default 0.01;  # Penalty parameter for number of
                                 # activities

param rho_excess  default 0.01;  # Penalty parameter on retained
                                 # capacity

param capacity_lb default requirement;
      # Used to constrain solutions

check capacity_lb >= requirement;

param capacity_ub default total_capacity;
      # Used to constrain solutions

check capacity_ub <= total_capacity;

check capacity_lb < capacity_ub;

param total_MV_ub default total_MV;
      #Used to constrain solutions

check total_MV_ub <= total_MV;

param total_MV_lb default 0;
      #Used to constrain solutions

check total_MV_lb >= 0;

check total_MV_lb < total_MV_ub;

param bignum := 1000000;


#######################################################################
# Index dictionary                                                    #
#                                                                     #
```

```
# j, j1, j2, j3, j4   Activity                                            #
###########################################################################


###########################################################################
# Variables                                                               #
###########################################################################

var   Open { j in ACTIVITIES } binary, <= closed[j];
      # Open or close variables for the activities

###########################################################################
# Variables, ALPHA, BETA, and GAMMA, are used to generate alternative  #
# solutions.                                                              #
###########################################################################

var   Alpha binary;  # At least one activity from the intersection is
                     # excluded from the solution.

var   Beta binary;   # At least one activity from the complement of the
                     # union is included in the solution.

var   Gamma binary;  # At least one activity from
                     #        EXCLD1 - (EXCLD1 intersect EXCLD2)
                     # and at least one site from
                     #        EXCLD2 - (EXCLD1 intersect EXCLD2)
                     # are included in the solution.



###########################################################################
# Objective functions                                                     #
###########################################################################

# Maximize the total retained activity military value while penalizing
# retained excess capacity and/or the number of open activities.

maximize max_retained_MV:

      sum { j in ACTIVITIES } Open[j] * ( MV[j] - rho_number )

         - rho_excess * sum { j2 in ACTIVITIES }
                          Open[j2] * capacity[j2];
```

```
# Minimize retained capacity subject to maintaining a minimal average
# military value.

minimize retained_capacity:

      sum { j in ACTIVITIES } Open[j] * capacity[j];


########################################################################
# Constraints                                                          #
########################################################################

# Meet the total requirement.

subject to meet_requirement:

      sum { j in ACTIVITIES } Open[j] * capacity[j] >= requirement;

# Solution must have a military value that is at least as good as
# the original average military value.

subject to avg_mv:

      sum { j in ACTIVITIES } Open[j] * ( MV[j] - AVG ) >= 0;


# Define box for allowable solutions in terms of total capacity
# and total retained military value.

subject to capacity_upper_bound:
      sum { j in ACTIVITIES } Open[j] * capacity[j] <= capacity_ub;

subject to capacity_lower_bound:
      sum { j in ACTIVITIES } Open[j] * capacity[j] >= capacity_lb;

subject to total_MV_upper_bound:
      sum { j in ACTIVITIES } Open[j] * MV[j] <= total_MV_ub;

subject to total_MV_lower_bound:
      sum { j in ACTIVITIES } Open[j] * MV[j]>= total_MV_lb;


########################################################################
# Constraints used to generate alternative solutions                   #
# Exclude solutions defined by the sets EXCLD1 and EXCLD2.             #
```

```
########################################################################

subject to alt_opt_cond_1:

      sum { s in EXCLD_INTER } Open[s] <= excld_num + 1 - Alpha;


subject to alt_opt_cond_2:

      sum { s in EXCLD_COMPLEMENT } Open[s] >= Beta;


subject to alt_opt_cond3a:

      sum { s in EXCLD_1DIFF2 } Open [s] >= Gamma;


subject to alt_opt_cond_3b:

      sum { s in EXCLD_2DIFF1 } Open [s] >= Gamma;

subject to alt_opt_cond_123:

      Alpha + Beta + Gamma >= 1;
```

F.1.1 Simple example data file

```
#######################################################################
# Data file for the simple case                                       #
#######################################################################

param: ACTIVITIES:   MV  capacity :=    # defines ACTIVITIES and MV
                                        # and capacity

      Alpha    65  4
      Bravo    68  7
      Charlie  70 10
      Delta    75 13
      Echo     85 15
;

set EXCLD1 := Charlie Delta Echo;

set EXCLD2 := Delta Echo;
```

```
param requirement := 23;
```

F.1.2 Simple example run file

```
######################################################################
# Run file for the simple case                                       #
######################################################################

model c:\brac\simple2\simple.mod;
data c:\brac\simple2\simple.dat;

# For each type of model, you must remove the ##'s from the appropriate
# lines below to run the selected model.  Only one line beginning
# with "objective" should have the "##" removed.  Also, make sure
# the appropriate "let" line, if required, has the "##" removed.

# Objective function for maximizing retained MV:

objective max_retained_MV;
let rho_number := 69; # Penalty parameter for number of retained sites.
let rho_excess := 0; # Penalty parameter for retained capacity.
##let capacity_ub := 30;
##let capacity_lb := 24;
##let total_MV_ub := 220;
##let total_MV_lb := 160;
##let constrain_avg_MV := 0;
printf "objective = max_retained_MV \n\n"  > c:\brac\simple2\simple.out;


# Objective needed to find BRAC 95 solution:

##objective retained_capacity;
##let constrain_avg_MV := 1;
##printf "objective = retained_capacity \n\n"  > c:\brac\simple2\simple.out;


display closed > c:\brac\simple2\simple.out;

display rho_number > c:\brac\simple2\simple.out;

display rho_excess > c:\brac\simple2\simple.out;

display EXCLD1 > c:\brac\simple2\simple.out;
```

```
display EXCLD2 > c:\brac\simple2\simple.out;

solve > c:\brac\simple2\simple.out;

display Open > c:\brac\simple2\simple.out;


printf "Average MV = %12.2f\n",sum { j in ACTIVITIES } ( Open[j] * MV[j])/
               sum { jj in ACTIVITIES } Open[jj]
                  > c:\brac\simple2\simple.out;

printf "Total MV = %12.2f\n",sum { j in ACTIVITIES } ( Open[j] * MV[j])
                  > c:\brac\simple2\simple.out;

printf "Total capacity = %12.2f\n",sum { j in ACTIVITIES }
         ( Open[j] * capacity[j]) > c:\brac\simple2\simple.out;
```

Appendix G

Modeling the depot example

The AMPL files used to model the depot example are included in this section.

G.1 Depot example with expansion code

This model file for the depot example includes the code that implements the model in chapter 4 to generate the scenario alternatives. The code for finding the maximum capacity of the seven depots, as discussed in chapter 2 is included in this model file (see the *max_production* objective function).

```
########################################################################
# Optimization Methodology                                             #
# JCSG Depot Example                                                   #
#                                                                      #
# Ron Nickel                                                           #
# 16 October 2003                                                      #
########################################################################

set   ACTIVITIES;

param closed { ACTIVITIES } binary, default 1;
      # Use to force closures.

set   FUNCTIONS;

param func_Importance { FUNCTIONS } default 1.0;
      # Importance to be associated with each type of functions.

set   RESOURCES;
```

```
param MV { ACTIVITIES };   # Overall military value for each
                           # activity

param funcval { j in ACTIVITIES, FUNCTIONS } >= 0, <= 100,
                     default MV[j];
      # Functional value for performing a function at an
      # activity (If functional values are not provided,
      # use the military value of the activity.

param max_funcval { i in FUNCTIONS }
         := max { j in ACTIVITIES } funcval[j,i];

check { i in FUNCTIONS } : max_funcval[i] > 0;

param capacity { ACTIVITIES, RESOURCES } >= 0;
      # Capacity for each resource type at each activity

param total_capacity { m in RESOURCES } :=
               sum { j in ACTIVITIES } capacity[j,m];
      # Total capacity across all activities for this resource type.

param rate { FUNCTIONS, RESOURCES } >= 0, default 0;
      # The amount of each resource required to produce one unit
      # for a function at an activity.

set OK_ASSIGNMENTS := { j in ACTIVITIES, i in FUNCTIONS:
                                    funcval[j,i] > 0 };
      # Need to identify the allowable assignment of functions to
      # activities.

param requirement { FUNCTIONS } >= 0;
      # Total requirement for the function.

param p { i in FUNCTIONS } := if requirement[i] > 0 then
               ( max { i1 in FUNCTIONS } requirement[i1] )/
                  requirement[i]
              else 0;
      # Scaled value for a product function based on the overall
      # requirement.


#######################################################################
# Expansion sets and parameters                                       #
#######################################################################
```

```
param maxResExp { ACTIVITIES, RESOURCES } default 0;
      # The increase in resource available for an activity.

param addAllowed { j in ACTIVITIES, m in RESOURCES } :=
         if maxResExp[j,m] > 0 then 1 else 0;
      # Allows capacity expansion in this resource type for this
      # activity if equal to 1; otherwise, no expansion allowed

set   EXP_ALLOWED :=
      { j in ACTIVITIES, m in RESOURCES: addAllowed[j,m] == 1 };
         # Subset of activity/resource combinations where expansion
         # would be allowed

#######################################################################
#  Sets used to find alternative solutions.  May be applied to either #
#  activities or functions or both.  Version for activities included  #
#  here.                                                              #
#######################################################################

set   EXCLD1 within ACTIVITIES default {};
         # Exclude the best solution.

set   EXCLD2 within ACTIVITIES default {};
         # Exclude the next best solution.

set   EXCLD_INTER := if card (EXCLD2) > 0 then ( EXCLD1 inter EXCLD2 )
                             else EXCLD1;

set   EXCLD_1DIFF2 := EXCLD1 diff EXCLD2;
         # Activities in EXCLD1 but not in EXCLD2

set   EXCLD_2DIFF1 := EXCLD2 diff EXCLD1;
         # Activities in EXCLD2 but not in EXCLD1

set   EXCLD_COMPLEMENT := ACTIVITIES diff ( EXCLD1 union EXCLD2 );
         # Set of activities not in EXCLD1 or EXCLD2

param excld_num := max( 0, card( EXCLD_INTER ) - 1 );




#######################################################################
```

```
# Parameters used to control the optimization                        #
######################################################################


param norm_Func_Values binary, default 0;
      # If set to 1, then normalize functional values.

param normed_FV { j in ACTIVITIES, i in FUNCTIONS }
      := if norm_Func_Values = 1 then 100 * funcval[j,i]/max_funcval[i]
         else funcval[j,i];
      # This will normalize the functional values.

param rho_resource default 0.01; # Penalty parameter for retaining
                                 # resources

param rho_number  default 0.01;  # Penalty parameter for number of
                                 # activities

param rho_elastic default 10000000; # Penalty parameter for control
                                    # of elastic resource capacity
                                    # expansion variables

param rho_val default 0.001;  # Term that forces unique assignments
                              # of requirements to activity/function
                              # combinations.

param minAssign default 1; # Non-zero assignments of workload have to
                           # be at least this big.

param bignum := 1000000;


#######################################################################
# Index dictionary                                                    #
#                                                                     #
# i, i1, i2, i3, i4   Function type                                   #
# j, j1, j2, j3, j4   Activity                                        #
# k                   Expansion resource                              #
# m                   Resource type                                   #
#######################################################################


#######################################################################
# Variables                                                           #
#######################################################################
```

```
var   Open { j in ACTIVITIES } binary, <= closed[j];
      # Open or close variables for the activities

var   FuncOpen { OK_ASSIGNMENTS } binary;
      # Variable used to count the number of open functional activities

var   Assign { OK_ASSIGNMENTS } >=0;
      # Amount of each functional requirement to assign to each
      # activity (constrained by resource capacities and allowable
      # assignments)


#######################################################################
# Variables for expansion                                             #
#######################################################################

var   AddRes { j in ACTIVITIES, m in RESOURCES } >= 0,
                                          <= maxResExp[j,m];
      # An elastic variable used to expand or add a resource
      # at an activity if expansion is allowed for this
      # activity/resource combination.

########################################################################
# Variables, ALPHA, BETA, and GAMMA, are used to generate alternative  #
# solutions.                                                           #
########################################################################

var   Alpha binary;  # At least one activity from the intersection is
                     # excluded from the solution.

var   Beta binary;   # At least one activity from the complement of the
                     # union is included in the solution.

var   Gamma binary;  # At least one activity from
                     #        EXCLD1 - (EXCLD1 intersect EXCLD2)
                     # and at least one site from
                     #        EXCLD2 - (EXCLD1 intersect EXCLD2)
                     # are included in the solution.


########################################################################
# Objective functions                                                  #
```

```
########################################################################

# Maximize the total retained activity military value while penalizing
# retained excess resource capacity, the number of open activities,
# and/or expansion allowed

maximize max_retained_MV:

      sum { j in ACTIVITIES } Open[j] * ( MV[j] - rho_number )

         - rho_resource * sum { m2 in RESOURCES } ( sum { j2 in ACTIVITIES }
               ( Open[j2] * capacity[j2,m2] + if (j2,m2) in EXP_ALLOWED
                     then AddRes[j2,m2] else 0 ) )/total_capacity[m2]

         - rho_elastic * sum { m4 in RESOURCES }
               ( sum { (j4,m4) in EXP_ALLOWED }
                     AddRes[j4,m4]
                          / total_capacity[m4] );


# Maximize the total value of assigned of functional requirements while
# penalizing the retention of excess resource capacity, the number of
# open activities, and/or expansion allowed

maximize max_retained_funcval:

      sum { (j,i) in OK_ASSIGNMENTS }
            func_Importance[i] * normed_FV[j,i] * FuncOpen[j,i]

         - rho_val * sum { (j1,i1) in OK_ASSIGNMENTS }
               ( Assign[j1,i1] * (100 - normed_FV[j1,i1])/requirement[i1] )

         - rho_number * sum { j2 in ACTIVITIES } Open[j2]

         - rho_resource * sum { m2 in RESOURCES } ( sum { j3 in ACTIVITIES }
               ( Open[j3] * capacity[j3,m2] + if (j3,m2) in EXP_ALLOWED
                     then AddRes[j3,m2] else 0 ) )/total_capacity[m2]

         - rho_elastic * sum { m4 in RESOURCES }
               ( sum { (j5,m4) in EXP_ALLOWED }
                     AddRes[j5,m4]
                          / total_capacity[m4] );


# Calculate the maximum production possible using the pseudo values
```

```
# for the product lines.

maximize max_production:

      sum { i in FUNCTIONS } p[i] *
            sum { (j,i) in OK_ASSIGNMENTS } Assign[j,i]

         - rho_elastic * sum { m in RESOURCES }
               ( sum { (j2,m) in EXP_ALLOWED }
                     AddRes[j2,m]
                          / total_capacity[m] );


########################################################################
# Constraints                                                          #
########################################################################

# Assign all of the functional requirements.

subject to meet_requirements { i in FUNCTIONS }:

      sum { (j,i) in OK_ASSIGNMENTS } Assign[j,i] >= requirement[i];


# Functions cannot be available at a closed activity.

subject to func_at_open_site { j in ACTIVITIES }:

      sum { (j,i) in OK_ASSIGNMENTS } FuncOpen[j,i] <=
              card( FUNCTIONS ) * Open[j];


# An activity cannot be open if no functions are assigned.

subject to active_func_at_open_site { j in ACTIVITIES }:

      Open[j] <= sum { (j,i) in OK_ASSIGNMENTS } FuncOpen[j,i];


# Assignments cannot be made to closed functions at an activity.

subject to function_open { (j,i) in OK_ASSIGNMENTS }:

      Assign[j,i] <= bignum * FuncOpen[j,i];
```

```
# Require a minimum assignment.

subject to min_assign { (j,i) in OK_ASSIGNMENTS }:

      Assign[j,i] >= minAssign * FuncOpen[j,i];



#######################################################################
# Expansion constraints                                               #
#######################################################################

# AddRes[j,i] must be set to 0 if Open[j] = 0.

subject to add_res_restrict { (j,i) in EXP_ALLOWED }:
        AddRes[j,i] <= maxResExp[j,i] * Open[j];


# Resources needed for assigned functional load cannot exceed the sum
# of available resources plus added resource capacity for each resource
# type.

subject to resources_available { j in ACTIVITIES, m in RESOURCES }:

      sum { (j,i) in OK_ASSIGNMENTS } Assign[j,i] * rate[i,m] <=
            capacity[j,m] + addAllowed[j,m] * AddRes[j,m];


########################################################################
# Constraints used to generate alternative solutions                   #
# Exclude solutions defined by the sets EXCLD1 and EXCLD2.             #
########################################################################

subject to alt_opt_cond_1:

      sum { s in EXCLD_INTER } Open[s] <= excld_num + 1 - Alpha;


subject to alt_opt_cond_2:

      sum { s in EXCLD_COMPLEMENT } Open[s] >= Beta;


subject to alt_opt_cond3a:
```

```
      sum { s in EXCLD_1DIFF2 } Open [s] >= Gamma;


subject to alt_opt_cond_3b:

      sum { s in EXCLD_2DIFF1 } Open [s] >= Gamma;

subject to alt_opt_cond_123:

      Alpha + Beta + Gamma >= 1;
```

G.2 Depot example with expansion data file

```
########################################################################
# Data file for the JCSG Depot Expansion Model                         #
########################################################################

param: ACTIVITIES:   MV :=    # defines ACTIVITIES and MV

      Alpha    62
      Bravo    61
      Charlie  67
      Delta    69
      Echo     63
      Foxtrot  75
      Golf     55
;

##set EXCLD1 := Alpha Charlie Delta Foxtrot;

##set EXCLD2 := Alpha Charlie Delta Echo Foxtrot;

set FUNCTIONS := AF Tanks Turbines  Elec;

param func_Importance :=
   AF    2
   Tanks 2
;

param funcval:
               AF Tanks Turbines Elec :=
      Alpha    82 0     35       57
      Bravo    50 0     62       89
```

```
      Charlie  66 0     81       80
      Delta    0  75    73       64
      Echo     0  93    44       74
      Foxtrot  0  54    54       85
      Golf     0  0     0        92
;

param requirement :=
      AF       40
      Tanks    32
      Turbines 300
      Elec     4500
;

set RESOURCES :=  Range Fab Hgrs Test;

param capacity:
               Range Fab   Hgrs  Test :=
      Alpha    2     1.2   12    0.9
      Bravo    1     0.9   7     1.3
      Charlie  1     1.6   3     2.3
      Delta    2     2.1   0     1.7
      Echo     1     3     0     0.7
      Foxtrot  2     1.7   0     2.4
      Golf     0     0     0     1.8
;

param maxResExp:
               Range Fab   Hgrs  Test :=
      Alpha    0     0.1   3     0.5
      Bravo    0     0.1   2     0.6
      Charlie  0     0.1   1     0.7
      Delta    0     0.2   0     0.4
      Echo     0     0.3   0     0.6
      Foxtrot  0     0.1   0     0.6
      Golf     0     0     0     0.4
;


param rate :      Range    Fab      Hgrs     Test :=
         AF       0.02     0.01     0.37037  0.002326
         Tanks    0.01     0.058824 0        0.004673
         Turbines 0        0.006667 0        0.003003
         Elec     0        0        0        0.000222
;
```

G.2.1 Depot example with expansion run file

```
######################################################################
# Run file for the JCSG Depot Model with Expansion                   #
######################################################################

model c:\brac\demo\depot.mod;
data c:\brac\demo\depot.dat;

# For each type of model, you must remove the ##'s from the appropriate
# lines below to run the selected model.  Only one line beginning
# with "objective" should have the "##" removed.  Also, make sure
# the appropriate "let" line, if required, has the "##" removed.

# Objective function for maximizing retained MV:

#objective max_retained_MV;
#let rho_resource := 0;    # Penalty parameter for excess capacity.
#let rho_number := 65;  # Penalty parameter for number of retained sites.
#let rho_elastic := 10000; # Penalty parameter for allowing expansion.
#printf "objective = max_retained_MV \n\n"  > c:\brac\demo\depot.out;

# Objective for maximizing assigned functional values:

##objective max_retained_funcval;
#let norm_Func_Values := 1;
##let rho_number := 0;  # Penalty parameter for number of retained sites.
##let rho_resource := 10000;    # Penalty parameter for excess capacity.
##let rho_elastic := 0.1; # Penalty parameter for allowing expansion.
##printf "objective = max_retained_funcval \n\n"  > c:\brac\demo\depot.out;

# Objective for maximizing production:

objective max_production;
let rho_resource := 0.0;    # Penalty parameter for excess capacity.
let rho_number := 0.0;  # Penalty parameter for number of retained sites.
let rho_elastic := 10000000; # Penalty parameter for allowing expansion.
display p > c:\brac\demo\depot.out;
```

```
##let closed["Alpha"]   := 0;
##let closed["Bravo"]   := 0;
##let closed["Charlie"] := 0;
##let closed["Delta"]   := 0;
##let closed["Foxtrot"] := 0;
##let closed["Golf"]    := 0;

display closed > c:\brac\demo\depot.out;

printf "objective = max_production \n\n"  > c:\brac\demo\depot.out;


display rho_resource > c:\brac\demo\depot.out;

display rho_number > c:\brac\demo\depot.out;

display rho_elastic > c:\brac\demo\depot.out;

display EXCLD1 > c:\brac\demo\depot.out;

display EXCLD2 > c:\brac\demo\depot.out;

solve > c:\brac\demo\depot.out;

display Open > c:\brac\demo\depot.out;

display FuncOpen > c:\brac\demo\depot.out;

display AddRes > c:\brac\demo\depot.out;

display Assign > c:\brac\demo\depot.out;

printf "\nTotal MV = %12.2f \n", sum { j in ACTIVITIES } ( Open[j] * MV[j])
            > c:\brac\demo\depot.out;

printf "\nAverage MV = %12.2f\n",sum { j in ACTIVITIES } ( Open[j] * MV[j])/
               sum { jj in ACTIVITIES } Open[jj]
                  > c:\brac\demo\depot.out;

printf "\nTotal normed and weighted functional value = %12.2f \n",
      sum { (j,i) in OK_ASSIGNMENTS }
         ( FuncOpen[j,i] * normed_FV[j,i] ) > c:\brac\demo\depot.out;

printf "\nFunction        Max v        Min v        Avg v\n"
                   > c:\brac\demo\depot.out;
```

```
printf { i in FUNCTIONS }: "%8s %12.2f %12.2f %12.2f \n", i,
   max { j in ACTIVITIES: Open[j] == 1} funcval[j,i],
   min { j in ACTIVITIES: Open[j] == 1 and funcval[j,i] > 0 } funcval[j,i],
   sum { (j,i) in OK_ASSIGNMENTS }
         ( FuncOpen[j,i] * funcval[j,i] )/ (sum { (j1,i) in OK_ASSIGNMENTS }
             FuncOpen[j1,i] ) > c:\brac\demo\depot.out;

printf "\nAverage fraction of resources retained = %12.2f \n",
   ( sum { m in RESOURCES } ( sum { j in ACTIVITIES }
                 ( Open[j] * capacity[j,m] + if (j,m) in EXP_ALLOWED
                    then AddRes[j,m] else 0 ) )/total_capacity[m] ) /
                          card( RESOURCES )
                          > c:\brac\demo\depot.out;

printf "\nResources consumed by each activity:\n" > c:\brac\demo\depot.out;

printf "Activity     " > c:\brac\demo\depot.out;
for { m in RESOURCES }
printf : "%12s", m > c:\brac\demo\depot.out;
printf "\n" > c:\brac\demo\depot.out;

for { j in ACTIVITIES } {
   if Open[j] == 1 then {
      printf "%8s   ", j > c:\brac\demo\depot.out;
      for { m in RESOURCES } {
         if capacity[j,m] > 0  then printf : "%12.3f ",
            sum { (j,i) in OK_ASSIGNMENTS } Assign[j,i] * rate[i,m]
               > c:\brac\demo\depot.out;
         if capacity[j,m] == 0 then printf "      na     "
                     > c:\brac\demo\depot.out;
      }
      printf "\n"  > c:\brac\demo\depot.out ;
   }
};

printf "\nUtilization of resources by each activity:\n"
                                 > c:\brac\demo\depot.out;

printf "Activity     " > c:\brac\demo\depot.out;
for { m in RESOURCES }
printf : "%12s", m > c:\brac\demo\depot.out;
printf "\n" > c:\brac\demo\depot.out;

for { j in ACTIVITIES } {
```

```
   if Open[j] == 1 then {
      printf "%8s   ", j > c:\brac\demo\depot.out;
      for { m in RESOURCES } {
         if capacity[j,m] > 0  then printf : "%12.3f ",
            sum { (j,i) in OK_ASSIGNMENTS } 100 * Assign[j,i] *
                  rate[i,m]/capacity[j,m]
                     > c:\brac\demo\depot.out;
         if capacity[j,m] == 0 then printf "      na     "
                  > c:\brac\demo\depot.out;
      }
      printf "\n"  > c:\brac\demo\depot.out ;
   }
};
```

Appendix H

DoN example code

```
###################################################################
# INDEXING CONVENTIONS                                            #
# s - is used to index Sites (e.g. installations/bases)           #
# a - is used to index Activities                                 #
# f - is used to index Functions (e.g. ship types, squadron types)#
# r - is used to index Resources (e.g. berths,hangars)            #
################################################################


###################################################################
# Sites, Activities, Functions, Resources, Use, Requirements
###################################################################
set SITES;
       #List of installations/bases
set ACTIVITY{SITES};
       #Lists of the activities at each site
set ACTIVITIES  := union {s in SITES} ACTIVITY[s];
       #List of all activities
set FUNCTIONS;
set RESOURCES;

param Capacity{ACTIVITIES, RESOURCES} >=0;
      #Available resources, by activity

param TotalCapacity{r in RESOURCES } :=
      sum{a in ACTIVITIES } Capacity[a,r];
      # Total capacity across all activities, by resource type.

param Rate{FUNCTIONS, RESOURCES} >=0;
      #Resource usage per unit of function
```

```
param Requirement{FUNCTIONS} >=0;
      #Total functional units to be assigned

param minAssign{FUNCTIONS} >=0;
      # Non-zero assignments of work must be at least this big.

#######################################################################
# Expansion
#######################################################################
param MaxResExp{ACTIVITIES,RESOURCES} >= 0;
    # Limit on potential expansion of each resource by activity


###################################################################
# Define objective function values and related penalties
###################################################################

param FuncVal{FUNCTIONS, ACTIVITIES}; #military value
      # Functional value for performing a function at an activity


param SitePenalty default 0.1;
       #objective function penalty on open sites
param ActivityPenalty default 0.1;
       #objective function penalty on open acitivities
param ExpPenalty default 1.0;
       #objective function penalty on adding resources -
param ScaleFactor{RESOURCES};
       #scales the expansion penalty to adjust for order of magnitude
param ResourcePenalty default 1.0;
       #objective function penalty on percent of original resources
       #now retained in open activities



###################################################################
# Feasibility Conditions: These will be used to check feasibility
# This is for "resources" that are not consumed, but may be required
###################################################################
set ResourcesMax; # Things like distance to training range where
                  # too HIGH a value is unacceptable
set ResourcesMin; # Things like runway length where too LOW
                  # a value is unacceptable
```

```
param AcceptableMax{FUNCTIONS, ResourcesMax};
            #The maximum acceptable value, by function
param AvailMax{ACTIVITIES, ResourcesMax};
            #The observed value at the activity
param AcceptableMin{FUNCTIONS, ResourcesMin};
            #The minimum acceptable value, by function
param AvailMin{ACTIVITIES, ResourcesMin};
            #The observed value at the activity



###################################################################
# VARIABLES:   Assign's track allocation of functions to activity #
#              OpenSite tracks if a base is open or closed     #
#              OpenActivity tracks if a base is open or closed    #
#              (closed is implied if no functions are assigned)   #
###################################################################
var OpenSite{SITES} binary >= 0;
             #1 if a site is used; 0 if closed
var OpenActivity{ACTIVITIES} binary >= 0;
             #1 if a activity is used; 0 if closed

var Assign{f in FUNCTIONS, a in ACTIVITIES} integer >=0,
        <= if( (exists {r in ResourcesMin} AvailMin[a,r]
         < AcceptableMin[f,r]) or
        ( exists {r in ResourcesMax} AvailMax[a,r] > AcceptableMax[f,r]) )
        then 0 else Requirement[f];
             # How much of the function is assigned to the activity

  # Note: The "if" statement checks feasibility. Assign is set to 0 if
  # an assignment is infeasible, otherwise it is constrained to be
  # between 0 and the total requirement for that function.

###################################################################
#  EXPANSION
###################################################################

var AddRes {a in ACTIVITIES, r in RESOURCES} >=0;
            #Variable to track resource expansion


###################################################################
# OBJECTIVE FUNCTION
###################################################################
```

```
maximize Military_Value :
   sum {f in FUNCTIONS, a in ACTIVITIES} (FuncVal[f,a]*Assign[f,a]
                /(Requirement[f]+.0001))
   - ExpPenalty * sum {a in ACTIVITIES, r in RESOURCES}
         ScaleFactor[r]*AddRes[a,r]
   - SitePenalty * sum {s in SITES} OpenSite[s]
   - ActivityPenalty * sum {a in ACTIVITIES} OpenActivity[a]
   - ResourcePenalty * sum {s in SITES, a in ACTIVITY[s], r in RESOURCES}
       (Capacity[a,r]*OpenSite[s] + AddRes[a,r])
                /(TotalCapacity[r]*card(RESOURCES))

#   - ResourcePenalty * sum {a in ACTIVITIES, r in RESOURCES}
#       (Capacity[a,r]*OpenActivity[a] + AddRes[a,r])
#                /(TotalCapacity[r]*card(RESOURCES))


   ;

# Maximize the military value from the laydown of functions to activities
# Note the penalty terms on keeping sites or activities open, on keeping
# resource and on expanding resources.
# The ScaleFactor scales resources to adjust for order of magnitude

###################################################################
# CONSTRAINTS
###################################################################

subject to MeetRequirements {f in FUNCTIONS}:
        sum {a in ACTIVITIES} Assign[f,a] = Requirement[f];
        # Any reason to have ">=" here?


#*********************************************************************
###  Use this if resources are fungible across all activities at the base

#subject to ResourceAvailability {s in SITES, r in RESOURCES} :
#        sum {f in FUNCTIONS, a in ACTIVITY[s]} (Rate[f,r]*Assign[f,a]) <=
#                sum {a in ACTIVITY[s]} (Capacity[a,r]*OpenActivity[a]
#                                     + AddRes[a,r]);
## THIS TAKES RESOURCE OUT IF YOU CLOSE THE ACTIVITY. NO CHANCE TO MOVE TO
## OTHER USE

subject to ResourceAvailability {s in SITES, r in RESOURCES} :
        sum {f in FUNCTIONS, a in ACTIVITY[s]} (Rate[f,r]*Assign[f,a]) <=
```

```
                sum {a in ACTIVITY[s]} (Capacity[a,r]*OpenSite[s]
                           + AddRes[a,r]);
## THIS TAKES RESOURCES OUT ONLY IF YOU CLOSE THE BASE


###  Use this if resources are not fungible between activities at the base

#subject to ResourceAvailability {a in ACTIVITIES, r in RESOURCES} :
#      sum {f in FUNCTIONS} (Rate[f,r]*Assign[f,a]) <=
#               Capacity[a,r]*OpenActivity[a] + AddRes[a,r];
#************************************************************************


subject to No_function_closed_activity {f in FUNCTIONS, a in ACTIVITIES}:
               Assign[f,a] <= Requirement[f]*OpenActivity[a];
      # Make sure no functions assigned to closed activity.
      # This is overkill if the second resource availability contraint
      # is used, but no harm is done

subject to ExpansionLimits {a in ACTIVITIES,r in RESOURCES}:  AddRes[a,r]
                                <= MaxResExp[a,r]*OpenActivity[a];
      # Make sure there is no resource expansion at closed activities.

subject to Close_Unused_Sites {s in SITES} : OpenSite[s]
      <= sum {a in ACTIVITY[s]} OpenActivity[a];
    # OpenSite is forced to zero when no functions are assigned to
    # activities on the base

subject to No_Activity_Closed_Site {s in SITES} : sum {a in ACTIVITY[s]}
                  OpenActivity[a] <= OpenSite[s]*card(ACTIVITY[s]);
    # No activities open at a base that is closed.
    # "card" counts the number of current activities at the base. This allow
    # up to that number to remain open.

#subject to Min_assign {f in FUNCTIONS, a in ACTIVITIES}:
#                     Assign[f,a] >= minAssign[f]*OpenFunc[f,a];
    # Require a mimimum level of assignment.Need to define OpenFunc as
    # having any of a function assigned.  See Ron's version
```

H.1 DoN example data

```
#############################
#         DATA file         #
```

```
#############################


######################################################
# Define the SITES
######################################################
set SITES := A1 A2 A3 A4 A5 A6 S1 S2 S3 D1 Spt1 Spt2
Spt3 Adm1 Adm2 Adm3 MCB1 MCB2;


######################################################
# Define the ACTIVITIES
######################################################
# Note: Sites may have multiple activities

 set ACTIVITY[A1]:=    NAS1 DPT2;
 set ACTIVITY[A2]:=    NAS2;
 set ACTIVITY[A3]:=    NAS3;
 set ACTIVITY[A4]:=    NAS4;
 set ACTIVITY[A5]:=    NAS5;
 set ACTIVITY[A6]:=    RDTE1 INTELe;
 set ACTIVITY[D1]:=    NAS6 NS4 DPT1;
 set ACTIVITY[S1]:=    NS1;
 set ACTIVITY[S2]:=    NS2;
 set ACTIVITY[S3]:=    NS3;
 set ACTIVITY[Spt1]:=  ICP1;
 set ACTIVITY[Spt2]:=  ICP2;
 set ACTIVITY[Spt3]:=  DPT4;
 set ACTIVITY[Adm1]:=  HQe;
 set ACTIVITY[Adm2]:=  HQw;
 set ACTIVITY[Adm3]:=  INTELw;
 set ACTIVITY[MCB1]:=  MCEF1 MCAW1;
 set ACTIVITY[MCB2]:=  MCEF2 DPT3;

######################################################
# DEFINE FUNCTIONS, REQUIREMENTS & MIN ASSIGNMENTS

######################################################
param:   FUNCTIONS:      Requirement   minAssign :=
          CVN             5             1
          CG              15            1
          VF              10            1
          VX              2             1
          VT              4             1
          Frame           15            3
```

```
        Tank           12           3
        Turb           100          20
        Elec           1600         100
        ICP            2            1
        ONI            2            1
        SEAHQ          1            1
        C3HQ           1            1
        BTLN           5            1
        WING           1            1
    ;


######################################################
# RESOURCES - enter the lists of resources
######################################################
set RESOURCES := CVNBERTH CGBERTH SIMA FLTEVOL HANGAR
TESTFAC OUTSTOR SPACE WATER P10 RANGE FAB;


###################################################
# RESOURCE AVAILABILITY/CAPACITY by activity
# (RHS of constraint)
###################################################
param Capacity default 0
:        CVNBERTH  CGBERTH   SIMA   FLTEVOL   HANGAR :=
NAS1       .         .         .    15000        5
NAS2       .         .         .    25000        6
NAS3       .         .         .    25000        6
NAS4       .         .         .    25000        7
NAS5       .         .         .    15000        4
RDTE1      .         .         .    20000        6
NS4       2         6         400      .          .
NAS6       .         .         .    15000        4
NS1       0         8         350      .          .
NS2       2         4         275      .          .
NS3       3         3         500      .          .
DPT1       .         .         .        .        4
DPT2       .         .         .        .        3
DPT3       .         .         .        .        .
DPT4       .         .         .        .        .
ICP1       .         .         .        .        .
ICP2       .         .         .        .        .
HQw     .         .         .         .         .
HQe     .         .         .         .         .
INTELe     .         .         .        .        .
```

```
INTELw    .         .         .         .         .
MCAW1      .         .         .     10000        4
MCEF1      .         .         .         .         .
MCEF2      .         .         .         .         .

# RESOURCE AVAILABILITY/CAPACITY (continued)
:          TESTFAC OUTSTOR   SPACE  WATER     P10 :=
NAS1      .2        60        22        3.0       2.5
NAS2      0         40        25        2.5       2.0
NAS3      .3        80        40        3.5       2.5
NAS4      0         30        20        1.5       2.0
NAS5      0         15        85        2.0       0.7
RDTE1     .5        50        35        2.3       3.0
NS4       0         35        70        3.5       0.5
NAS6      .5        35        30        1.5       0.8
NS1       0         40        25        4.0       0.7
NS2       0         50        50        4.0       0.7
NS3       0         40        25        5.5       0.7
DPT1      1         50        6         3.5       1.2
DPT2      1.5       50        8         3.0       1.6
DPT3      1.7       50        7         3.0       1.5
DPT4      0.7       50        7         3.0       1.5
ICP1      .8        90        10        .         0.8
ICP2      .9        90        9         .         0.5
HQw       .         .         50        .2         .
HQe       .         .         60        .2         .
INTELe    .         .         30        .2         .
INTELw    .         .         50        .2         .
MCAW1      .        20        15        1.5        1.5
MCEF1     .         20        25        1.2         .8
MCEF2     .         35        25        2.5        1.0

# RESOURCE AVAILABILITY/CAPACITY (continued)
:         RANGE    FAB :=
NAS1      .2         .
NAS2      .2         .
NAS3      .3         .
NAS4      .3         .
NAS5      1.5        .
RDTE1     1.5        .
NS4       .2         .
NAS6      .8         .
NS1       .2         .
NS2       .          .
NS3       .2         .
```

```
DPT1      .4         1.2
DPT2      .2          .9
DPT3      .4         2.1
DPT4      .2         3.0
ICP1      .         .5
ICP2      .         .3
HQw      .         .
HQe      .         .
INTELe    .         .
INTELw    .         .
MCAW1     0.2        .
MCEF1     3         .
MCEF2     3         .
;


###################################################
#EXPANDABILITY - limit to adding resources by activity
###################################################
param MaxResExp default 0
:        CVNBERTH  CGBERTH   SIMA   FLTEVOL   HANGAR :=
NAS1      .         .         .      2500        2
NAS2      .         .         .      3500        2
NAS3      .         .         .      3500        2
NAS4      .         .         .      3500        2
NAS5      .         .         .      2500        2
RDTE1     .         .         .      5000        2
NS4      1         1         25       .          .
NAS6      .         .         .      2500        2
NS1      1         2         25       .         .
NS2      1         2         25       .         .
NS3      1         2         25       .         .
DPT1      .         .         .         .        1
DPT2      .         .         .         .        1
DPT3      .         .         .         .        .
DPT4      .         .         .         .        .
ICP1      .         .         .         .        .
ICP2      .         .         .         .        .
HQw       .         .         .         .        .
HQe       .         .         .         .        .
INTELe    .         .         .         .        .
INTELw    .         .         .         .        .
MCAW1      .         .         .      5000       2
MCEF1      .         .         .         .        .
MCEF2      .         .         .         .        .
```

```
#EXPANDABILITY - upper bound on adding (continued)
:          TESTFAC OUTSTOR   SPACE  WATER     P10 :=
NAS1      .2         30       25        .25        .2
NAS2      .2         30       25        .25        .2
NAS3      .2         30       25        .25        .2
NAS4      .2         30       25        .5         .2
NAS5      .2         30       25        .5         .2
RDTE1     .2         30       25        .25        .2
NS4       .2         30       25        .20        .1
NAS6      .2         30       25        .20        .2
NS1       .2         30       25        .25        .2
NS2       .2         30       25        .25        .2
NS3       .2         30       25        .25        .2
DPT1      .2         10        5        .25        .2
DPT2      .3         10        5        .25        .2
DPT3      .2         10       5         .25        .2
DPT4      .1         10       5         .2         .2
ICP1      .1         15       5         .          .2
ICP2      .1         15       5         .          .2
HQw       .         .         10        .          .
HQe       .         .         10        .          .
INTELe    .         .         5         .          .
INTELw    .         .         5         .          .
MCAW1      .1        15        20       .20        .2
MCEF1      .1        20        24       .20        .2
MCEF2      .1        25        24       .20        .2

#EXPANDABILITY - upper bound on adding (continued)
:         RANGE    FAB :=
NAS1      .        .1
NAS2      .        .1
NAS3      .        .1
NAS4      .        .1
NAS5      .        .1
RDTE1     .        .1
NS4       .        .1
NAS6      .        .1
NS1       .        .1
NS2       .        .1
NS3       .        .1
DPT1      .        .2
DPT2      .        .2
DPT3      .        .4
DPT4      .        .5
```

```
ICP1      .        .1
ICP2      .        .1
HQw       .        .
HQe       .        .
INTELe    .        .
INTELw    .        .
MCAW1     .        .1
MCEF1     .        .1
MCEF2     .        .1
;

##################################################
#EXPANDABILITY - factor to scale resources
##################################################
param     ScaleFactor:=
CVNBERTH  1
CGBERTH   .5
SIMA      .005
FLTEVOL   .0001
HANGAR    .5
TESTFAC   1
OUTSTOR   .01
SPACE     .01
WATER     1
P10       1
RANGE     1
FAB       1
;


##################################################
# RESOURCE USE - the use of resources per
# unit of function. (LHS constraint coefficents)
##################################################
param Rate default 0

:          CVNBERTH  CGBERTH  SIMA   FLTEVOL   HANGAR :=
CVN        1        0         15      .         .
CG         0        1         10      .         .
VF         .        .         .      3000      1
VX         .        .         .      1500      1
VT         .        .         .      4000      1
Frame     .         .         .         .      .08
Tank      .         .         .         .      .
Turb      .         .         .         .      .
```

```
Elec     .         .          .         .      .
ICP       .         .         .         .      .
ONI       .         .         .         .      .
SEAHQ     .         .         .         .      .
C3HQ      .         .         .         .      .
BTLN      .         .         .         .      .
WING      .         .         .       9000     3

# RESOURCE USE (continued)
:        TESTFAC   OUTSTOR     SPACE    WATER     P10 :=
CVN       .         .           3         .3         0
CG        .         .           2         .15        0
VF        .         .           2         .2        0.12
VX        .         .           5         .2        0.12
VT        .         .           5         .2        0.06
Frame    .0005     .22         .02       .005      .002
Tank     .        .3           .024      .006      .0025
Turb     .001     .03          .0025     .0006     .00025
Elec     .0001    .0015        .0002     .00005    .000015
ICP       .2        60          10         0.05      0
ONI       .         .           15         0.05      0
SEAHQ     .         .           24         0.05      0
C3HQ      .         .           10         0.05      0
BTLN      .         .           7           .1      .
WING      .         .           7           .5      .3

# RESOURCE USE (continued)
:         RANGE    FAB :=
CVN       .         .
CG        .         .
VF        .         .
VX        .         .
VT        .         .
Frame     .004     .015
Tank      .001    .008
Turb      .         .
Elec      .         .
ICP       .         .
ONI       .         .
SEAHQ     .         .
C3HQ      .         .
BTLN      1.0       .
WING      .         .
;
```

```
####################################################
# OBJECTIVE FUNCTION COEFFICIENTS - the value of
# assigning one unit of function to a site
# Note: This does not account for infeasibility or
# value of resources used in capacity constraints
####################################################

param FuncVal default 0
:         NAS1     NAS2     NAS3     NAS4     NAS5     NS1       :=
CG        48.67    52.82    53.55    51.08    34.99    74.96
CVN       52.05    62.38    62.99    48.49    38.93    67.84
VF        59.73    51.78    68.00    70.01    61.44    18.02
VX        45.67    50.96    58.74    65.65    46.88    22.47
VT        56.83    42.33    61.78    69.36    59.99    25.01
Frame     40.72    20.76    14.31    8.99     6.46     21.80
Tank      20.72    20.76    14.31    8.99     6.46     21.80
Turb      52.72    20.76    14.31    8.99     6.46     21.80
Elec      58.72    20.76    14.31    8.99     6.46     21.80
ICP       18.22    25.26    22.81    16.68    15.06    20.29
ONI       39.94    42.09    11.18    25.66    46.98    28.03
SEAHQ     13.31    26.21    28.07    26.24    45.14    11.73
C3HQ      13.31    26.21    28.07    26.24    45.14    11.73
BTLN      39       36       32       31       28       35
WING      45       43       49       51       47       20


# OBJECTIVE FUNCTION COEFFICIENTS - (continued)
:         NS2      NS3      NS4      NAS6     RDTE1     DPT1      :=
CG        62.19    80.03    73.28    58.28    21.98    23.85
CVN       63.23    82.62    72.99    57.99    24.26    19.31
VF        37.96    39.82    50.36    62.36    69.03    13.50
VX        41.36    34.47    41.32    53.32    55.99    21.38
VT        41.90    41.90    45.16    57.16    68.86    10.03
Frame     25.31    16.97    48.06    48.06    17.16    82
Tank      25.31    16.97    15.06    15.06    17.16    5
Turb      25.31    16.97    33.06    33.06    17.16    35
Elec      25.31    16.97    37.06    37.06    17.16    57
ICP       22.51    15.98    35.02    35.02    20.24    43.36
ONI       20.88    17.12    40.62    40.62    59.40    13.59
SEAHQ     39.87    5.90     37.90    37.90    30.04    17.41
C3HQ      39.87    5.90     37.90    37.90    30.04    17.41
BTLN         26       22       20    30       33          15
WING         28       30       32    41       48          15
```

```
# OBJECTIVE FUNCTION COEFFICIENTS - (continued)
:          DPT2      DPT3      ICP1      ICP2      HQe    INTELe      :=
CG         22.86     23.86     12.72     10.87     10.63    8.72
CVN        20.34     24.34     7.09      6.80      9.41     7.63
VF         12.94     25.94     6.31      6.24      3.09     2.05
VX         24.46     25.46     15.57     19.95     11.43    6.35
VT         9.59      29.59      4.81      4.75      2.37     1.59
Frame      50        10        37.81     43.40     14.86    13.33
Tank       5         75        37.81     43.40     14.86    13.33
Turb       62        73        37.81     43.40     14.86    13.33
Elec       89        64       37.81     43.40     14.86    13.33
ICP        45.73     45.73     78.28     79.47     17.84    14.83
ONI        16.81     19.81     31.34     19.91     34.89    51.18
SEAHQ      25.76     28.76     20.60     19.48     53.82    41.67
C3HQ       25.76     28.76     20.60     19.48     42.99    27.62
BTLN       15        15        10        10        10       15
WING       15        15        7         8         10       7


# OBJECTIVE FUNCTION COEFFICIENTS - (continued)
:           HQw   INTELw    MCEF1     MCEF2     MCAW1    DPT4     :=
CG          13.34    8.65      62.19     23.86     62.19    23.86
CVN         12.02    6.37      63.23     24.34     63.23    24.34
VF          3.63     1.69      37.96     25.94     37.96    25.94
VX          13.02    3.48      41.36     25.46     41.36    25.46
VT          2.81     1.30      41.90     29.59     41.90    29.59
Frame       3.93     8.62      25.31     30.72     25.31    5
Tank        3.93     8.62      25.31     58.72     25.31    93
Turb        3.93     8.62      25.31     32.72     25.31    44
Elec        3.93     8.62      25.31     52.72     25.31    74
ICP         11.08    11.85     22.51     45.73     22.51    45.73
ONI         11.21    47.76     20.88     19.81     20.88    19.81
SEAHQ       42.99    27.62     39.87     28.76     39.87    28.76
C3HQ        53.99    41.67     39.87     28.76     39.87    28.76
BTLN        10       10        64        62        42       15
WING        10       10        58        52        45       15

;


###################################################
# MINIMUM FEASIBILITY CONDITIONS
###################################################
```

```
set ResourcesMin:= RunWay     Gflops;

# Minimum Acceptable level of resource/feature
###################################################
param AcceptableMin default 0:
          RunWay        Gflops :=
VF        8000          .
VX        9000          .
VT        7000          .
WING      8000          .
ONI       .             6
;

# Actual Activity Levels of the resource/feature
###################################################
param AvailMin default 0:

          RunWay        Gflops  :=
NAS1      10000         8
NAS2      12000         10
NAS3      12000         3
NAS4      12000         5
NAS5      10000         11
NS1       .             7
NS2       .             3
NS3       .             6
NS4       .             4
NAS6      12000         5
RDTE1     12000         15
DPT1      .             5
DPT2      .             2
DPT3      .             2
DPT4      .             2
ICP1      .             7
ICP2      .             6
HQw .                3
HQe    .             1
INTELe    .             12
INTELw    .             15
MCAW1     8500          3
MCEF1     .             3
MCEF2     .             4
;
```

```
####################################################
# MAXIMUM FEASIBILITY CONDITIONS
####################################################

set ResourcesMax:=  SpecUse   AGRange TrainRte InstRng;

# Maximum Acceptable value of the resource/feature
####################################################
param AcceptableMax default 99999:

          SpecUse   AGRange   TrainRte  InstRng :=
VF        1100      1100      .         .
VX        1000      1000      1000      .
VT        1100      1100      .         1100
WING      1100      1100      .         .
;


# Actual Levels of the resource/feature
####################################################
param AvailMax default 99999:
          SpecUse   AGRange TrainRte InstRng :=
NAS1      534       468       534       534
NAS2      622       1090      595       1090
NAS3      394       394       508       394
NAS4      414       521       414       414
NAS5      421       421       421       421
NS1       775       1304      421       775
NS2       541       481       541       541
NS3       401       401       642       401
NS4       1036      729       321      1163
NAS6      1036      729       321      1163
RDTE1     919       321       414       919
MCAW1     400      1050       950      1050
;
```

H.2 DoN example run file

```
########################################################################
# Run file for the MILDEP example                                      #
########################################################################
```

```
model c:\brac\MILDEP\mildep.mod;
data c:\brac\MILDEP\mildep.dat;

# Set the penalty parameters

let SitePenalty := 0.1;
let ActivityPenalty := 0.1;
let ResourcePenalty := 1.0;
let ExpPenalty := 1.0;

printf "MILDEP Example \n\n" > c:\brac\MILDEP\mildep.out;

solve > c:\brac\MILDEP\mildep.out;

printf "\n\n*** Results ***\n\n" > c:\brac\MILDEP\mildep.out;

printf "Total sites open = %12.0f\n\n",
      sum {s in SITES} OpenSite[s] > c:\brac\MILDEP\mildep.out;

printf "Total activities open = %12.0f\n\n",
   sum {a in ACTIVITIES} OpenActivity[a] > c:\brac\MILDEP\mildep.out;

printf "Total functional value retained = %12.4f\n\n",
   sum {f in FUNCTIONS,a in ACTIVITIES} FuncVal[f,a]*Assign[f,a]
         /(Requirement[f]+.00001)
               > c:\brac\MILDEP\mildep.out;

printf "Percent resources open = %12.4f\n\n",
   100 * sum {s in SITES, a in ACTIVITY[s], r in RESOURCES}
    (Capacity[a,r]*OpenSite[s] + AddRes[a,r])
            /(TotalCapacity[r]*card(RESOURCES))
                  > c:\brac\MILDEP\mildep.out;

printf "Average MV per site = %12.4f\n\n",
   (sum {f in FUNCTIONS,a in ACTIVITIES} FuncVal[f,a]*Assign[f,a]/
      (Requirement[f]+.00001))/(sum {s in SITES} OpenSite[s])
               > c:\brac\MILDEP\mildep.out;

printf "Average functional MV per activity = %12.4f\n\n",
   (sum {f in FUNCTIONS,a in ACTIVITIES} FuncVal[f,a]*Assign[f,a]/
      (Requirement[f]+.00001))/(sum {a in ACTIVITIES} OpenActivity[a])
                  > c:\brac\MILDEP\mildep.out;

printf "\n\nPenalty parameters\n\n" > c:\brac\MILDEP\mildep.out;
```

```
printf "Penalty on leaving sites open = %10.4f\n", SitePenalty
               > c:\brac\MILDEP\mildep.out;
printf "Penalty on activities open = %10.4f\n", ActivityPenalty
               > c:\brac\MILDEP\mildep.out;
printf "Penalty on resources left open = %10.4f\n", ResourcePenalty
               > c:\brac\MILDEP\mildep.out;
printf "Expansion penalty factor = %10.4f\n", ExpPenalty
               > c:\brac\MILDEP\mildep.out;


option display_transpose -2;
option display_round '';


#display "Feasibility exclusions" > c:\brac\MILDEP\mildep.out;
#display Assign.astatus > c:\brac\MILDEP\mildep.out;

printf "\n\nBases open (1) or closed (0)\n\n"
               > c:\brac\MILDEP\mildep.out;
display OpenSite > c:\brac\MILDEP\mildep.out;

printf "\n\nActivities open (1) or closed (0)\n\n"
               > c:\brac\MILDEP\mildep.out;
display OpenActivity > c:\brac\MILDEP\mildep.out;

printf "\n\nFunctional Assigments\n\n"
               > c:\brac\MILDEP\mildep.out;
display Assign > c:\brac\MILDEP\mildep.out;



option display_round '1';


#display "Resource Excess Capacity & dual values"
#                 > c:\brac\MILDEP\mildep.out;
#display ResourceAvailability.uslack, ResourceAvailability.dual
#                 > c:\brac\MILDEP\mildep.out;

printf "\n\nExcess Capacity by activity\n\n"
                  > c:\brac\MILDEP\mildep.out;
display {a in ACTIVITIES, r in RESOURCES}
  (Capacity[a,r]*OpenActivity[a] + AddRes[a,r]
         - sum{f in FUNCTIONS} Rate[f,r]*Assign[f,a])
                  > c:\brac\MILDEP\mildep.out;
```

```
printf "\n\nExcess Capacity by site\n\n"
                  > c:\brac\MILDEP\mildep.out;
display {s in SITES,r in RESOURCES } sum{a in ACTIVITY[s]}
   (Capacity[a,r]*OpenSite[s] + AddRes[a,r]
            - sum{f in FUNCTIONS}(Rate[f,r]*Assign[f,a]))
                  > c:\brac\MILDEP\mildep.out;

printf "\n\nActivity functional value\n\n"
                  > c:\brac\MILDEP\mildep.out;
display {a in ACTIVITIES}
   sum {f in FUNCTIONS} FuncVal[f,a]*Assign[f,a]
            /(Requirement[f]+.0001)
                  > c:\brac\MILDEP\mildep.out;

printf '\n\nExpansion of resources\n\n'
                  > c:\brac\MILDEP\mildep.out;
display {a in ACTIVITIES, r in RESOURCES} AddRes[a,r]
                  > c:\brac\MILDEP\mildep.out;


printf "\n\nTotal functional value = %12.4f\n\n",
   sum {f in FUNCTIONS,a in ACTIVITIES} FuncVal[f,a]*Assign[f,a]
         /(Requirement[f]+.0001)
                  > c:\brac\MILDEP\mildep.out;

printf "\n\nExpansion penalty value = %12.4f\n\n",
      sum {a in ACTIVITIES, r in RESOURCES}
               ExpPenalty*ScaleFactor[r]*AddRes[a,r]
                  > c:\brac\MILDEP\mildep.out;

printf "\n\nTotal open site penalties = %12.4f\n\n",
   sum {s in SITES} SitePenalty*OpenSite[s]
                  > c:\brac\MILDEP\mildep.out;

printf "\n\nOpen activity penalties = %12.4f\n\n",
   sum {a in ACTIVITIES} ActivityPenalty*OpenActivity[a]
                  > c:\brac\MILDEP\mildep.out;

printf "\n\nExcess resource penalties = %12.4f\n\n",
   sum {s in SITES, a in ACTIVITY[s], r in RESOURCES}
      ResourcePenalty*(Capacity[a,r]*OpenSite[s] + AddRes[a,r])/
                     (TotalCapacity[r]*card(RESOURCES))
                           > c:\brac\MILDEP\mildep.out;
```

```
close c:\brac\MILDEP\mildep.out;
```

Appendix I

Computational environment

The computational environment built to solve the optimization problems created in the BRAC process is described in this appendix. We first describe the physical layout of the network. We then describe the repository structure for archive the optimization models used by the the DoN and JC-SGs. Finally, we describe the process followed to run optimization models and archive the results.

I.1 The CNA-BRAC Local Area Network (LAN)

A local-area-network (LAN) was created inside the DoN BRAC office to support the optimization modelling described in this document. The network is formally named `cna-brac`. The network includes two servers, each equipped with two Intel Xeon 2.0 GHz processors and 2 GB of memory. The two servers are connected to the network through a router as shown in figure I.1. We use network cabling throughout the DoN BRAC spaces to connect the analysts' laptops to the two servers, also depicted in figure I.1. For security reasons, this LAN has no external connections to the outside world, i.e., there is no connection to the internet.

Each of the servers is running the Linux[1] operating system. The laptops are either running the Linux operating system or the Windows 2000 operating system. The two servers are identified as `server1.cna-brac` and `server2.cna-brac`.

All of the communications across the network are performed using the secure shell (SSH) protocol.[2] The laptops using the Windows 2000 operating

[1]See [17].
[2]See [8].

system use the open-source software package PuTTY[3] to remotely access the servers. The the open-source Samba software[4] is used to access the directory structures on the servers and to move files between the laptops and the servers.

I.2 Repository Structure

We have used the open-source Concurrent Versions System (CVS) to build and maintain our repositories (see [16]) on server 1 (`server1.cna-brac`). These repositories allow us to maintain complete histories of all model changes and runs made with these models. We have created a repository for each JCSG and another one for the operational installations as shown in figure I.2. The repositories are all maintained in `/home/cvs` on server 1 (`server1.cna-brac/home/cvs`). Each repository has modules that correspond to the different models used in the analysis. For example, the `ops` repository, as shown in figure I.2 (`server1.cna-brac/home/cvs/ops`), has three modules, a module for modelling the air installations, a module for modelling installations that host ships and submarines, and a module for installations hosting Navy and USMC ground units.

Analysts use the TortioseCVS[5] software installed on their Windows 2000 laptops to interact with the CVS repository as described in the next section.

I.3 Managing optimization models

For this discussion of managing the optimization process, we will refer to figure I.3. The first step in managing a module is to create the module. The analyst must first create the model description file that is written in AMPL language, an example of which is given in appendix F. The user must also create a data file and a script file that tells the AMPL interpreter what files to process and what output should be created. These three files should be placed in an initial directory such that the lowest-level directory name corresponds to the desired module name. In the case of figure I.3, this directory is `C:\InitialModels\Ground`. The repository module name will be called *Ground*.

To test the model, data, and run files, the analyst will place these files in a directory on one of the servers in his home directory that he will use

[3]This software is available at `www.chiark.greenend.org.uk/~sgtatham/putty/`.

[4]See [15].

[5]This software and documentation is available at `www.tortoisecvs.org`.

to actually execute his AMPL jobs. In the case shown in figure I.3, the analyst with user name *nickelr* places the files in a directory called *Run*. After executing the AMPL command and running the script file, a new file will appear that contains the outputs from the job. In this case, the output file is called *Ground.out*. Note that there may be more than one output file created. All of these files, including the output files, should be copied back to the original directory. This cycle should be repeated until the analyst is satisfied with these files.

The analyst is now ready to create the repository module for this model. from inside this initial directory, he will use Tortoise to create a module with the name of the lowest-level directory containing his model and data files in the repository on the server.

Once the repository is created, the analyst will create a permanent working version of the repository on his laptop or local computer using the Tortoise checkout command. We call this directory the sandbox. In the case of the example shown in figure I.3, this directory is called
`C:\CVS Working\BRACModels\Ground`.

If modifications to the model are required or additional runs with modified data are required. The analyst will make needed changes to the files in the sandbox and then copy the necessary files to his run directory on the server to perform the optimization runs. After the optimization runs are finished, the analyst will copy the output files back to his sandbox on his laptop or other local computer and then commit this new version to the repository. Each time a new version is committed to the repository, the analyst provides a note that explains what has changed and why.

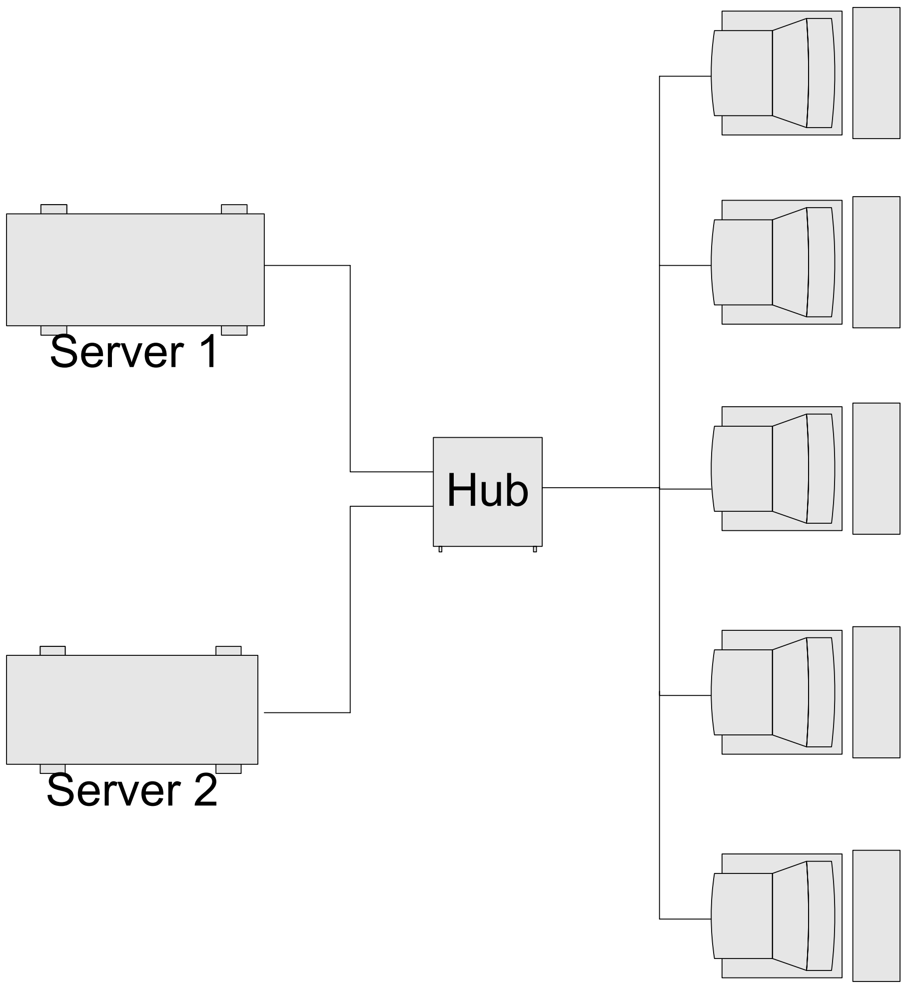

Figure I.1: Computational LAN

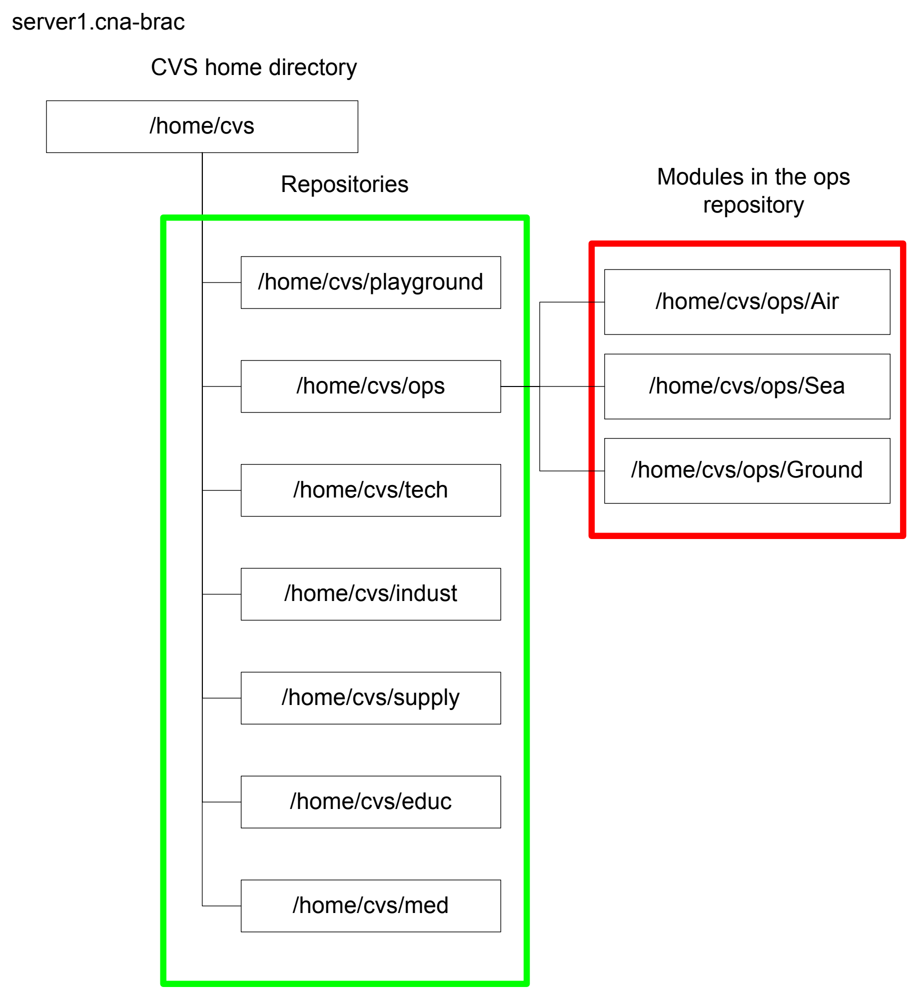

Figure I.2: Repository structure

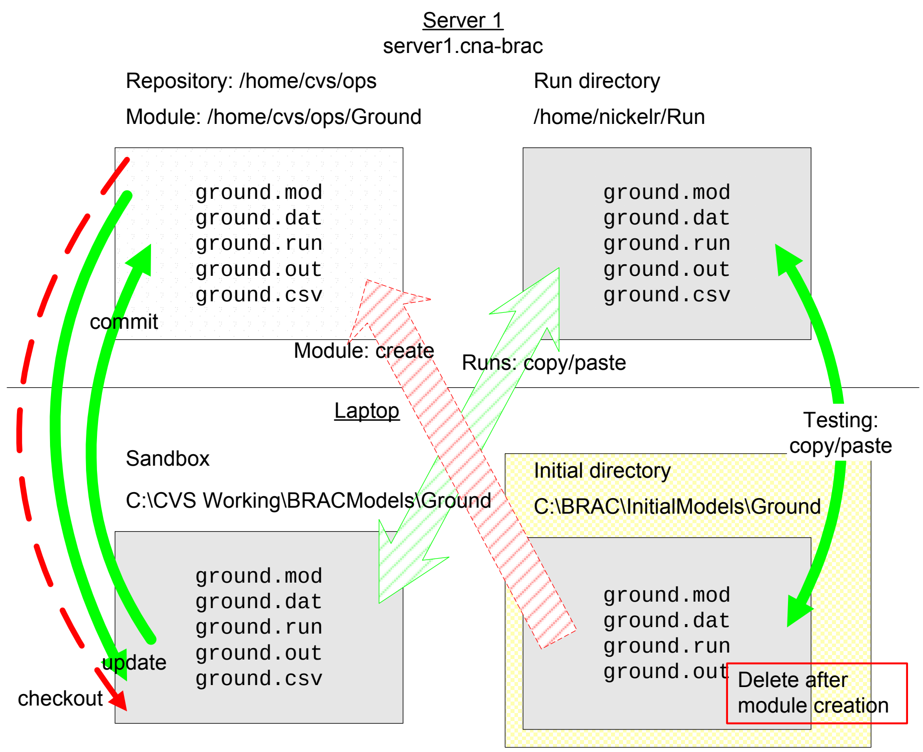

Figure I.3: Computational procedures

Appendix J

Fuzzy mathematics

Most decisions are based on imprecise or *fuzzy* inputs—perceptions, generalizations in the form of fuzzy truths or fuzzy inferences. In our minds, we generally then proceed to engage in *fuzzy* processing of those inputs and come up with some sort of averaged, summarized, or normalized decision outputs—often expressed as a single number that we can act on. Fuzzy logic can be thought of simply as an attempt to do at least some of that internalized information processing in a more conscious, visible, consistent way.

J.1 Fuzzy Subsets

Boolean logic describes the operations in classical set theory, i.e., the operations associated with conventional sets. Fuzzy logic,[1] on the other hand, describes the operations in fuzzy sets. A fuzzy set can be a group of things that cannot be precisely defined. Consider for example, a fuzzy set of large military bases.[2] How large is a large base? Where would the dividing line between, say, small bases and large bases be? Would it be at 10,000 acres, at 40,000 acres. If it were somehow determined to be at 40,000 acres, would a 39,999 acre base be considered a small base?[3] The answers of course depend

[1] Fuzzy logic is an extension of Boolean logic, developed by Dr. Lotfi A. Zadeh of U.C. Berkeley in his seminal works on fuzzy sets (See [18] and [19].), to deal with the ambiguities the characterize most human decision making.

[2] Examples of fuzzy sets are all around us - adequate housing, good schools, warm days, dense population, large communities, secure perimeters, low crime rates, etc.

[3] Fuzzy logic is an attempt to apply use of human common sense rather than rigid, expedient but somewhat arbitrary rules in answering such questions.

upon whom you ask.[4]

A conventional subset S of a set U can be defined as a set of ordered pairs with a first elements, x, in the set U and the associated second, *membership elements*, $m_S(x)$, in the binary set $\{0, 1\}$. We define one such ordered pair for each element of U where the set U containing all possible values of x, is referred to as the *universe of discourse*. A value 0 for a membership element indicates non-membership in S and a value 1 indicates membership. The truth or falsity of the statement, x is in S for any element in U, then, can be determined simply by finding the ordered pair whose first element is x. The statement above is true if the membership element of that ordered pair is 1 and the statement is false if that membership element is 0. This sort of binary logic is actually very useful in digital applications, i.e., where the reality of interest is either "black or white." These conventional sets having only $\{0, 1\}$ membership values are often called *crisp* sets in contrast to *fuzzy* sets to be described below.

In engineering, where most decision-making is done in a reality that is neither "black" nor "white," but one that is "gray" in a continuum between black and white, we may need a broader concept of set membership to deal with the gray areas. Fuzzy set theory and logic provide the needed extensions.

A fuzzy subset F of a set U can also be thought of as a set of ordered pairs with its first elements, x, in the set U but the associated second, membership elements, $m_F(x)$, in the interval $[0, 1]$. Again, we can define one such ordered pair for each element of U. A membership value of 0 for an element indicates non-membership, and a value of 1 indicates full membership. Membership values in between indicate intermediate degrees of membership. The set U is the universe of discourse for the fuzzy subset F. The $[0, 1]$ interval mapping that produces the second elements is called the *membership function* of F.[5] The values assigned by membership functions are intuitive assessments rather than precise, empirically measured facts. The degree to which the statement, x is in F, for any element in U is true, then, can be estimated simply by finding the ordered pair whose first element is x. The *degree of truth* of the statement is indicated by the value of the associated membership function, $m_F(x)$.

To illustrate this, consider how we might describe bases with good access to repair resources. In this case the universe of discourse , U, would be the set of all military bases being considered. We can define a fuzzy subset of

[4] Other references on fuzzy sets and fuzzy logic include [20], [14], [10], and [9].

[5] Fuzzy sets constitute a superset of conventional binary sets.

bases with good access to repair resources, which will answer the question to what degree is base x close to a repair depot? Let the fuzzy subset, C, be bases that are close to a repair depot. For each base, x, in U, then, a membership function will associate its degree of membership in C based on the distance from the base to the nearest depot. Suppose, it was felt that, in this context, what close means can be described:

$$m_C(x) = \begin{cases} 1 & : \quad d(x, depot) \leq 10 \\ 1 - \frac{d(x,depot)-10}{40} & : \quad 10 < d(x, depot) \leq 50 \\ 0 & : \quad d(x, depot) > 50 \end{cases}$$

where $d(x, depot)$ is the distance from a base x to the depot. Given this definition, here are some example values:

Base x	$d(x, depot)$	$m_C(x)$
A	30	0.500
B	3	1.000
C	40	0.250
D	55	0.000
E	25	0.625
F	12	0.950

This says that the degree of truth of the statement "Base E is close to a depot" is $m_C(E) = 0.625$. Membership functions can have any suitable shape. They are not limited to the simple linear ramp shape shown for $m_C(x)$ above.

Multiple criteria are common in our actual decision-making. Membership functions can also consider more than a single criterion. For example, beyond base-to-depot distance, the membership function for a fuzzy set of bases with good access to repair resources might further depend on some measure of the size of that nearest depot. In that case, the membership function would require a more complex calculation.

J.2 Fuzzy logic operators

We illustrated above how a simple statement like *Base x is close to a repair depot.* could be expressed using fuzzy logic. How might we represent more complex statements like *Base x is close to a repair depot and is far from a training range.* or perhaps *x is not close to a wetland area.*?

J.2.1 Elemental fuzzy logic operators

Using fuzzy logic, we can develop various related fuzzy sets that might be of particular interest simply by constructing new membership functions from the ones we already have. These constructions are based upon the complement of a set, the intersection of two sets, and the union of two sets:

complement The membership for x not in the set C, denoted $m_{\hat{C}}(x) = 1.0 - m_C(x)$.

intersection The membership for x in set C and in set D ($C \cap D$), denoted $m_{C \cap D}(x) = \min[m_C(x), m_D(x)]$.

union The membership for x belonging to C or D ($C \cup D$), denoted $m_{C \cup D}(x) = \max[m_C(x), m_D(x)]$.

Notice that if one used just the values 0 and 1 in these fuzzy operator definitions, the result would be the familiar truth tables found in conventional set theory and Boolean logic. Boolean logic then is just a special case of Fuzzy logic.

For example, assume the same definition of C as used in the example above and a second fuzzy subset, D, military bases with a full capability depot support with an associated membership function based on depot size:

$$m_D(x) = \begin{cases} 0 & : \quad s(x, depot) \leq 2 \\ \frac{s(x, depot) - 2}{38} & : \quad 2 < s(x, depot) \leq 40 \\ 1 & : \quad s(x, depot) > 40 \end{cases}$$

where $s(x, depot)$ is the size of the depot serving base x in terms of thousands of square feet (Kft^2). Example values follow:

Base x	$s(x, depot)$	$m_D(x)$
Alpha	50	1.000
Beta	5	0.079
Charlie	20	0.474
Delta	25	0.605
Echo	8	0.158
Foxtrot	21	0.500

One might be particularly interested in characterizing bases having a number of different base repair access situations — possibly in terms of these two fuzzy sets:

- Fuzzy set Q = base x is close to a depot *and* that depot has a full range of capabilities, $(C \cap D)$.

- Fuzzy set R = base x is either close to a depot *or* the nearest depot has a full range of capabilities, $(C \cup D)$.

- Fuzzy set T = base x is *not* close to any repair depot, $(\hat{C})$

The table below shows the computed membership values of the bases, or their degrees of inclusion, in fuzzy sets Q, R, and T based on the values of their memberships in fuzzy sets C and D, $m_C(x)$ and $m_D(x)$, respectively.

Base x	$m_Q(x)$	$m_R(x)$	$m_T(x)$
Alpha	0.500	1.000	0.500
Bravo	0.079	1.000	0.000
Charlie	0.250	0.474	0.750
Delta	0.000	0.605	1.000
Echo	0.158	0.625	0.375
Foxtrot	0.500	0.950	0.050

The traditional Boolean logic interpretations of the *AND* ($\cap$) and the *OR* ($\cup$) operators for conventional crisp sets are sometimes also used in fuzzy sets. In conventional crisp sets, intersections and unions (*AND* and *OR* operators) have a probabilistic rather than membership connotation and are associated with multiplication and addition operations rather than with finding minimums and maximums. In some fuzzy set applications, the probabilistic interpretations may[6] be appropriate also. To illustrate the differences between the notions of membership and probability, consider the two independent fuzzy subsets: Naval Air Stations (NAS) with expandable main runways, E, and NASs near over-water training ranges, W. Assume a particular base x, has membership values $m_E(x) = 0.80$ and $m_W(x) = 0.90$. Imposing a probabilistic interpretation, we infer,[7] $p_E(x) = m_E(x) = 0.80$ and $p_W(x) = m_W(x) = 0.90$ where $p_S(x)$ is the probability that base x is in set S.

[6]Some might argue that a probabilistic approach is inappropriate since base x's membership in a particular subset of bases is not a random event governed by the laws of probability. There is also a non stochastic interpretation. When an arithmetic product is used for the *AND* operator it is sometimes described as a soft *AND*, suggesting that, in some situations, there are possible trade-offs between the operands.

[7]It is more commonly the case that membership values are inferred from empirical probabilities.

The *AND* operator for generating the intersection of two fuzzy sets

What would this base's membership value be in a subset of all bases that are both runway expandable *and* have over-water training ranges? A conventional, crisp, set theory interpretation would use a probabilistic, multiplicative approach to estimating a joint membership value: $m_{E \cap W}(x) = p_E(x)p_W(x) = 0.72$. As the number of factors considered in the joint membership increases, the combined membership value computed in this fashion would get generally smaller and smaller — approaching zero in the limit. Notice that the probabilistic result is always lower than both the input membership values which, when interpreted as the chance of a base being both expandable and having access to an over-water training range, would be expected.

A fuzzy set theory interpretation, on the other hand, would use minimums in estimating joint membership values:

$$m_{E \cap W}(x) = \min[m_E(x), m_W(x)] = 0.80$$

As the number of factors considered in the joint membership increases, the joint membership value cannot increase. Dictated by base x's weakest required qualification, the joint membership value would generally decrease as the number of requirements increases. Note, however, that this would still generally be larger than the corresponding probabilistic value.

The *OR* operator for generating the union of two fuzzy sets

What would this base's membership value be in a subset of all bases that are either runway expandable *or* have over-water training ranges, or both? A conventional, crisp, set theory interpretation would again use probabilities, but this time in an essentially additive computation to estimate a combined membership value: $m_{E \cap W}(x) = p_E(x) + p_W(x) - p_E(x)p_W(x) = 0.98$. As the number of factors considered increases, the combined membership value computed in this fashion would get generally larger - approaching one in the limit. Notice that the probabilistic result is always greater than both the input membership values which again, would be expected when interpreted as the chance of a base being either expandable or having access to an over-water training range.

A fuzzy set theory interpretation would use maximums in assigning union membership values: $m_{E \cup W}(x) = \max[m_E(x), m_W(x)] = 0.90$. As the number of factors considered increases, the combined membership value cannot decrease. Dictated by base x's strongest eligibility criterion, the combined membership value would generally increase. Note, however, that this would still generally be smaller than the corresponding probabilistic value.

The intersection of two fuzzy sets is a fuzzy set and the union of two fuzzy sets is a fuzzy set with membership functions $m_{A\cap B}(x) = \min[m_A(x), mB(x)]$ and $m_{A\cup B}(x) = \max[m_A(x), m_B(x)]$, respectively. We can define other fuzzy set operators with membership functions that are intermediate between the two minimum and maximum extremes of intersections and unions.

Mean value operators for generating the harmonic fuzzy set (HFS), geometric fuzzy set (GFS), and arithmetic fuzzy set (AFS)

The arithmetic mean of n numbers is what we usually call their average value:

$$A(x_1, x_2, \ldots, x_n) \equiv \frac{1}{n}\sum_{j=1}^{n} x_j$$

For example, consider a training facility. Assume every year it trains an average of 48 new pilots. In year 1, it budgeted 40 flight hours per trainee. In year 2, it budgeted 55 flight hours per trainee. And in year 3, it budgeted 60 flight hours per trainee. What is the average number of flight hours budgeted for each trainees in the last three years? Trainees, on average, each received $A(40, 55, 60) = (40 + 55 + 60)/3 = 51.67$ flight hours of training.

There are other means, however, that may also be useful. The geometric mean of n numbers is defined:

$$G(x_1, x_2, \ldots, x_n) \equiv \Big(\prod_{j=1}^{n} x_j\Big)^{\frac{1}{n}}$$

A computationally more convenient form using logarithms is given by:

$$\log G(x_1, x_2, \ldots, x_n) \equiv \frac{1}{n}\sum_{j=1}^{n} \log x_j$$

$\log G(x_1, x_2, \ldots, x_n)$ is the arithmetic mean of the logarithms of the data.

The harmonic mean of n numbers is defined by

$$H(x_1, x_2, \ldots, x_n) \equiv n/\sum_{j=1}^{n} \frac{1}{x_j}$$

The arithmetic mean is used in situations in which the underlying quantities add together to produce totals that are of interest. The arithmetic mean can be thought of as the equivalent single value of all addends that would produce the same total. The geometric mean, on the other hand, is used in situations in which underlying quantities multiply together to produce products that are of interest. The geometric mean can be thought of as the equivalent single value of the factors that would produce the same products.

For example, say an investment earns 0 percent the first year, 20 percent the second year, and 30 percent the third year of its life. What is the Return On Investment percentage (ROI)? It is not the simple 16.67 percent obtained from the arithmetic mean of the three yearly percentages. Instead, the ROI is found by using the geometric mean of those three factors:

$$G(1.00, 1.20, 1.30) = (1.00 \times 1.20 \times 1.30)^{\frac{1}{3}} \simeq 1.1598$$

So the ROI is just under 16 percent — less than the 16.67 percent obtained from the arithmetic mean. It can be shown that the geometric mean is always less than or equal to the arithmetic mean. To find the equivalent average factor indicated in a set of data, we use the geometric mean.

The harmonic mean is often useful when averaging rates or speed. For example, the qualifying speed for a race car attempting to qualify for the Indy 500 race run at Speedway, Indiana, is the speed the car averages over four laps around the $2\frac{1}{2}$-mile oval track. If we let d be the distance around the track and t_i be the time the car takes to circle the track in hours on lap i, then

$$\text{average speed} = \frac{d + d + d + d}{t_1 + t_2 + t_3 + t_4}$$

Now, let $r_i = \frac{d}{t_i}$ be the speed of the car on the i-th lap. The average of the four r_is is not the average speed for the four laps. Since we can write $t_i = \frac{d}{r_i}$, the average speed over the four laps can be rewritten as:

$$\text{average speed} = \frac{d + d + d + d}{d/r_1 + d/r_2 + d/r_3 + d/r_4}$$

which can be rewritten as

$$\text{average speed} = \frac{4}{1/r_1 + 1/r_2 + 1/r_3 + 1/r_4}$$

This is the harmonic mean of the four individual lap speeds.[8]

Consider another example of two training air stations that each have 2,400 flight hours available each year to train pilots. The table below shows the number of pilots trained each year for three years at each facility along with the average number of hours required each year to train a single pilot. If the average number of flight hours required to train a pilot is the measure used to compare the two training air stations, then training air station B is the superior training air station since 130 pilots were trained using 7,200 flight hours for an average of 55.38 flight hours per trained pilot. The average for training air station A is 57.60 flight hours per trained pilot. If an analyst had averaged the averages in the table, the comparison would have been reversed since the average of the averages is 57.78 for training air station A and 59.74 for training air station B.

If we had only the averages shown in the table, we could still derive the correct averages by computing the harmonic averages for each set of averages.

	Training Air Station A		Training Air Station B	
Year	Pilots trained	Avg. flight hours per pilot trained	Pilots trained	Avg. flight hours per pilot trained
1	45	53.33	59	40.68
2	40	60.00	30	80.00
3	40	60.00	41	58.54

One final note on these means, if we let $X = \{x_1, x_2, \ldots, x_n\}$, it has been shown[9] that

$$\min\{X\} \leq H(X) \leq G(X) \leq A(X) \leq \max\{X\}$$

We have previously presented the intersection and union operators for combining elements from two fuzzy sets, FS_1 and FS_2:

$$m_{FS_1 \cap FS_2}(x) = \min[m_{FS_1}(x), m_{FS_2}(x)]$$

and

$$m_{FS_1 \cup FS_2}(x) = \max[m_{FS_1}(x), m_{FS_2}(x)]$$

[8] This is a modification of the discussion of harmonic means from [13].

[9] See [13]

In addition, we can use the three different kinds of means, arithmetic (A), geometric (G), and harmonic (H), to augment these fuzzy set operators:

- The arithmetic composite fuzzy set has a membership function given by the arithmetic mean:

$$m_{A(FS_1,FS_2)}(x) \equiv A(m_{FS_1}(x), m_{FS_2}(x))$$

- The geometric composite fuzzy set has a membership function given by the geometric mean:

$$m_{G(FS_1,FS_2)}(x) \equiv G(m_{FS_1}(x), m_{FS_2}(x))$$

- The arithmetic composite fuzzy set has a membership function given by the arithmetic mean:

$$m_{H(FS_1,FS_2)}(x) \equiv H(m_{FS_1}(x), m_{FS_2}(x))$$

These definitions are easily extended to combine any number of membership functions.

J.2.2 Structural properties fuzzy sets

The *support* of any fuzzy or crisp set A, $s(X)$, is another set, a conventional set of all $x \in U$ that have positive membership in A. Formally, $s(A) \equiv x \in U | m_A(x) > 0$.

The *height* of any fuzzy or crisp set A, $h(A)$, is the largest membership value, $m_A(x)$, over all $x \in U$. Formally, $h(A) \equiv \max\{m_A(x) | x \in U\}$.

Any fuzzy or crisp set, A, is a *normal* set if its height, $h(A)$, is 1. Any non-normal fuzzy set can be normalized simply by dividing its membership function, $m_A(x)$, by its height, $h(A)$.

Any fuzzy or crisp set, A, is a *subset* of another set B if the membership value for all x in A is less than or equal to the membership value of x in B. That is, $A \subseteq B \Leftrightarrow m_A(x) \leq m_B(x)\, \forall x \in U$.

Any fuzzy or crisp set, A, is *equal* to another set B if the membership value for all x in A, $m_A(x)$, equals the membership value of x in B. That is, $A = B \Leftrightarrow m_A(x) = m_B(x)\, \forall x \in U$.

Any fuzzy or crisp set, A, is *not equal* to another set B if for at least one x in U, the membership values $m_A(x)$ and $m_B(x)$ are not equal. That is, $A \neq B$ if and only if there exists an $x \in U | m_A(x) \neq m_B(x)$.

Any fuzzy or crisp set, A, is a *proper* subset of another set B if the membership value for all x in A, $m_A(x)$, is less than or equal to the membership value of x in B and for at least one x, $m_A(x) < m_B(x)$. That is, $A \subset B$ if and only if $m_A(x) \leq m_B(x) \, \forall x \in U$ and $m_A(x) < m_B(x)$ for some $x \in U$.

An *alpha cut* of any fuzzy or crisp set A, denoted A_α, is the set of all x such that $m_A(x)$ is greater than or equal to α. A *strong alpha-cut*, $A_{\alpha+}$ is the set of all x in U such that $m_A(x)$ is strictly greater than α. Mathematically, alpha cut $A_\alpha \equiv \{x \in U | m_A(x) \geq \alpha\}$ and the strong alpha cut $A_{\alpha+} \equiv \{x \in U | m_A(x) > \alpha\}$

Since every x in U is either in or not in any given alpha cut or strong alpha cut of all fuzzy sets and crisp sets, alpha cuts and strong alpha cuts[10] are always crisp sets with $\{0, 1\}$ membership functions.

The strong 0 cut, $A_{\alpha+}$ i.e., $\{x \in U | m_A(x) > 0\}$, would be the support of A, $s(A)$, i.e., $s(A) = A_{\alpha+}$. A_1 is called the *core* set of A.

That we can generate two families of nested crisp sets simply by steadily decreasing α from 1 to 0 is a useful feature of alpha-cuts and strong alpha-cuts of any fuzzy set.

The level set, $L(A)$, is the set of all α values that correspond to $m_A(x)$ for at least one $x \in U$. Formally

$$L(A) \equiv \{\alpha \quad | \quad \exists \, x \in U \, \ni: m_A(x) = \alpha\}$$

The *scalar cardinality* of a set A, $|A|$, is commonly used as the measure of A's aggregate membership. For crisp sets, this is the number of members in the set. For fuzzy sets, it is a proxy measure of the total degree or strength or pervasiveness of its membership. In either case,

$$|A| \equiv \sum_{x \in U} m_A(x)$$

J.2.3 Operational properties of crisp sets that also apply to fuzzy sets

Given the definitions of the fuzzy logic operators $\bigcap$ and $\bigcup$ above, the following important operational properties from conventional set theory also apply in fuzzy set theory. Let A, B, and C be either fuzzy or crisp sets with membership functions $m_A(x)$, $m_B(x)$, and $m_C(x)$, respectively. Then the following will be true.

DeMorgans Theorem: $\widehat{A \cap B} = \hat{A} \cup \hat{B}$ and $\widehat{A \cup B} = \hat{A} \cap \hat{B}$

[10] Similarly, every x in U is either in or not in any given support set, so the support set, s(F), is also always a crisp set.

Associativity: $(A \cap B) \cap C = A \cap (B \cap C)$ and $(A \cup B) \cup C = A \cup (B \cup C)$

Commutativity: $A \cap B = B \cap A$ and $A \cup B = B \cup A$

Distributivity: $A \cap (B \cup C) = (A \cap B) \cup (A \cap C)$ and $A \cup (B \cap C) = (A \cup B) \cap (A \cup C)$

J.2.4 The linear combination of two fuzzy sets

A linear combination of two fuzzy sets A and B is the fuzzy set $w_1A + w_2B$ where w_1 and w_2 can be any scalar weights. The membership function of the linear combination, $m_{w_1A+w_2B}(.)$ is composed of the similarly weighted sums of the elemental membership functions: $m_{w_1A+w_2B}(x) = w_1'm_A(x) + w_2'm_B(x)$ for all x contained in A or B and w_1' and w_2' are the corresponding weights normalized to sum to one. For example, an ideal naval air station might be one that is conveniently close to an over-water training range, $C_T(x)$ and also reasonably close to an ICP, $C_I(x)$. Suppose proximity to an over-water training range is considered to be twice as important as closeness to an ICP. To model the relative importance of factors, the weights should sum to one. The membership function defining the fuzzy set of ideal naval air stations would be the linear combination: $NAS_{IDEAL} = 0.67C_T + 0.33C_I$

Assume we have the closeness membership values $m_{C_T}(x)$ and $m_{C_I}(x)$ for six Naval Air Stations shown in the table below. The computations for the composite NAS_{IDEAL} membership values are shown in the table as well.

NAS x	$m_{C_T}(x)$	$m_{C_I}(x)$	$m_{NAS_{IDEAL}}(x)$	Ranking
1	0.40	1.00	0.598	4
2	1.00	0.25	0.753	1
3	0.60	0.70	0.633	3
4	0.80	0.35	0.652	2
5	0.00	0.90	0.297	6
6	0.50	0.50	0.500	5

Had the relative importance of being close to an over-water training range and to an ICP been different, the rankings of the $m_{NAS_{IDEAL}}(x)$ values could have been entirely different as shown in the table below when the relative weights are reversed ($NAS_{IDEAL} = 0.33C_T + 0.67C_I$). Note NAS's 2, 4, and 6, though they have quite different proximities to training ranges and ICPs have essentially the same composite ranking under this weighting scheme.

NAS x	$m_{C_T}(x)$	$m_{C_I}(x)$	$m_{NAS_{IDEAL}}(x)$	Ranking
1	0.40	1.00	0.802	1
2	1.00	0.25	0.498	6
3	0.60	0.70	0.667	2
4	0.80	0.35	0.499	5
5	0.00	0.90	0.603	3
6	0.50	0.50	0.500	4

J.2.5 The product of two fuzzy sets

The product of two fuzzy sets A and B is the fuzzy set AB. The membership function of the product, m_{AB} is composed of the products of the two elemental membership functions:

$m_{AB}(x) = m_A(x)m_B(x)$ for all x contained in A or B.

For example, suppose our system downtime is such that we make tradeoffs between getting special-order repair parts quickly and getting them from the lowest-cost vendor. In this case, the weaker vendor qualification, delivery timeliness or selling price, does not adequately capture the decision tradeoff that must be made. The more appropriate membership function for vendors in the fuzzy set AB, where A is the set of vendors that provide timely delivery and B is the set of low-cost vendors, would be $m_{AB}(x)$ for all x in A or B.

J.2.6 Powers of a fuzzy set

Powers of fuzzy set A are the fuzzy set A^y where y can be any positive number. The membership function of A^y is composed of the elements of the membership function for A each raised to the power y.

$m_{A^y}(x) = [m_A(x)]^y$ for all x contained in A.

As an example, suppose we want to apply different repair depot proximity requirements depending upon current availability of equipment in good operating condition at different bases. Let the fuzzy subset, C, be bases that are close to a repair depot. For each base x, let $m_C(x)$ be its membership function. When there is a surplus of equipment available for base operations, fast access to repair facilities may become less important. In this case, we can define a fuzzy set of bases that are "somewhat close" to a repair depot with membership function $m_{SC}(x) = [m_C(x)]^{0.5}$. Exponents less than 1 will give the needed effect.

On the other hand, if there was a shortage of equipment in good operating condition at bases, fast access to repair facilities might be more important. In this case, we can define a fuzzy set of bases that are "very close" to a repair depot with membership function $m_{VC}(x) = [m_C(x)]^2$. Exponents greater than 1 will give the correct effect.

Appendix K

Measuring value with fuzzy functions

In this appendix, we define a number of general types of fuzzy functions that can be used to measure value.

K.1 Small values are desirable

The fuzzy function defined in this section are useful for cases for which small values tend to be desirable and large values are not desirable.

K.1.1 Binary measures

This set membership function, FS1, a downward step function as shown in figure K.1, is a special kind of fuzzy set - one that has a binary membership function. FS1, then, is simply a Boolean set, sometimes called a crisp set. FS1 describes groups of objects with well-defined, single-valued cut-offs for being included in the set. The membership function for FS1 is given by:

$$m_{(d)}(x) = \begin{cases} 1 & : \quad x \leq d \\ 0 & : \quad x > d \end{cases}$$

For example, let FS1 be the set of military bases at sites that are free of environmental contaminants.[1] There should be no fuzziness associated with

[1] Alternatively, FS1 could be the set of military bases at sites that are in the safe range for PCB soil contamination as defined by the Environmental Protection Agency, i.e., bases with less than 1 part per million (ppm) of PCB contamination. There should be no fuzziness associated with which bases are included in this set and which are not. Base membership decreases as we go from those with more than 1 ppm of contamination.

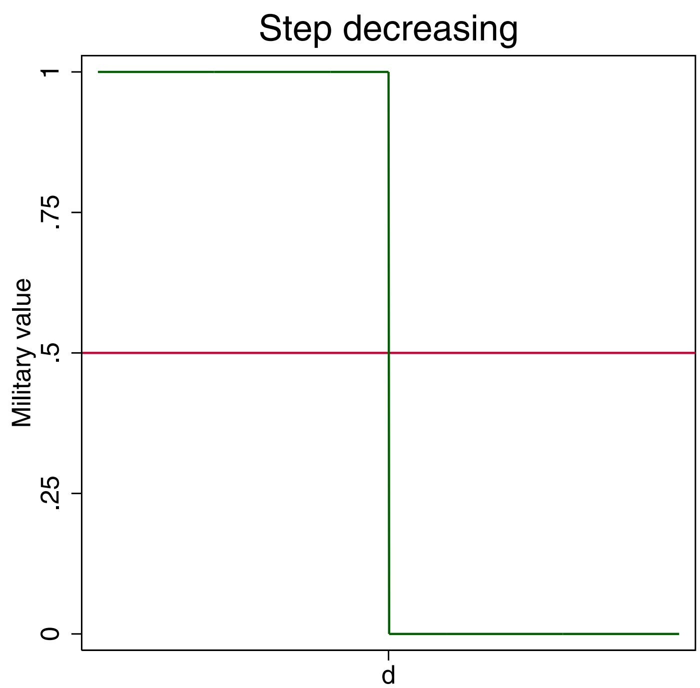

Figure K.1: Fuzzy function 1

which bases are included in this set and which are not. Base membership decreases as we go from no contaminants to one or more so we would choose FS1 over the upward step FS6 described below. Assume that bases have one or more kinds of contaminants that require cleaning up under the law and x is the number of contaminants at a base.

- Bases with no contaminants are by definition in the contaminant free set, with membership function $m_{(d)}(x) = 1$ for $x \leq 1$.

- Bases with one or more contaminants are in the set of bases requiring cleanup. The membership function for these bases is $m_{(d)}(x) = 0$ for $x > 1$.

The parameter d in this case would be $d = 0$ though, in general, this need

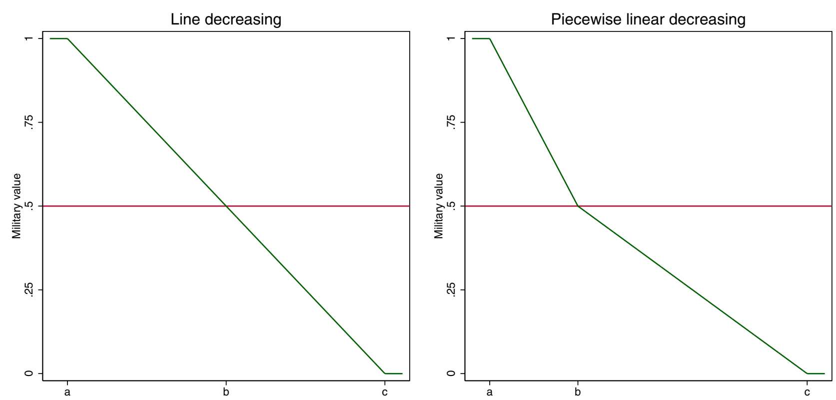

Figure K.2: Fuzzy function 2

not be the case. For example, had the criterion for inclusion in the set been being habitats for two or fewer endangered species, d would have been 2.

K.1.2 Measures that worsen at a constant rate

This fuzzy set membership function, FS2, a downward sloping linear function, may have its median, b, mid way between its minimum and its maximum parameters, a and c, as shown in the figure K.2 on the left. Or, FS2's median may be some other point between the two extremes, as shown in the right in figure K.2. FS2 describes groups of objects with membership values that decrease at an approximately constant rate. The membership function for FS2 is given by:

$$m_{(a,b,c)}(x) = \begin{cases} 1 & : \quad x < a \\ 1 - \frac{1}{2}\left(\frac{x-a}{b-a}\right) & : \quad a \leq x < b \\ \frac{1}{2}\left(\frac{c-x}{c-b}\right) & : \quad b \leq x < c \\ 0 & : \quad x \geq c \end{cases}$$

As an example, let FS2 be the set of military bases in communities where the local crime rates are low. The fuzziness arises in trying to quantify the meaning of the word low. Assume that

- Bases in communities where the annual crime rates were below 1,500 crimes per 100,000 population would have rates much below recent national averages and definitely would be described as having low crime rates, $m_{(a,b,c)}(x) = 1$ for $x < 1,500$.

- Bases in communities where the annual crime rates were 4,500 crimes per 100,000 population would have rates much above recent national averages and definitely would be described as having high crime rates, $m_{(a,b,c)}(x) = 0$ for $x \geq 4,500$.
- Bases in communities where the annual crime rates were around 3,000 crimes per 100,000 population would have rates close to recent national median rates so we might use that also as the local median rate with fuzzy set membership $m_{(a,b,c)}(3,000) = 0.5$.

These would be estimates of the minimum, median, and maximum parameters: $a = 1,500$, $b = 3,000$, and $c = 4,500$ crimes per 100,000 population, respectively.

Say, from one particular military base of interest, the crime rate in the local community is $x = 4,100$ crimes per 100,000 population. Then $m_{(1,500,3000,4,500)}(4,100) = 0.133$

K.1.3 Measures that worsen at faster and faster rates

FS3 is a downward sloping, concave fuzzy set membership function with minimum and maximum parameters a and c, respectively, and an exponent parameter n that is greater than or equal to 1. FS3 describes groups of objects with membership values that decrease at an accelerating rate, i.e., decrease very slowly at first then gradually decrease more rapidly until they reach zero at c as shown in figure K.3. The membership function for FS3 is

$$m_{(a,c,n)}(x) = \begin{cases} 1 & : \quad x < a \\ 1 - \left(\frac{x-a}{c-a}\right)^n & : \quad a \leq x < c; n \geq 1 \\ 0 & : \quad x \geq c \end{cases}$$

Let FS3 be the set of acceptable waiting times for base housing. The fuzziness arises in trying to quantify the meaning of the word *acceptable* which varies from applicant to applicant and from decision-maker to decision-maker. Assume that

- Times less than 2 weeks are considered by almost everyone as acceptable, $m_{(a,c,n)}(x) = 1$ for $x < 2$.

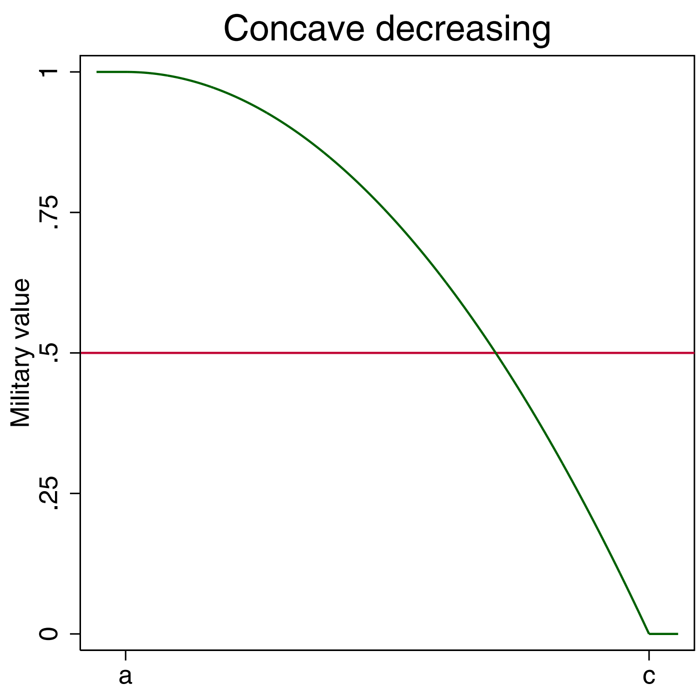

Figure K.3: Fuzzy function 3

- Times greater than or equal to 8 weeks are considered by most to be undesirable so $m_{(a,c,n)}(x) = 0$ for $x \geq 8$.

These are estimates of the minimum and maximum parameters: $a = 2$ weeks, $c = 8$ weeks. Unless there is reason to choose some other value, as general practice for simplicity, we set the exponent $n = 2$.

Say, for one particular military base of interest, the average wait time for base housing in recent years has been $x = 3.9$ weeks, then $m_{(2,8,2)}(3.9) =$ 0.900

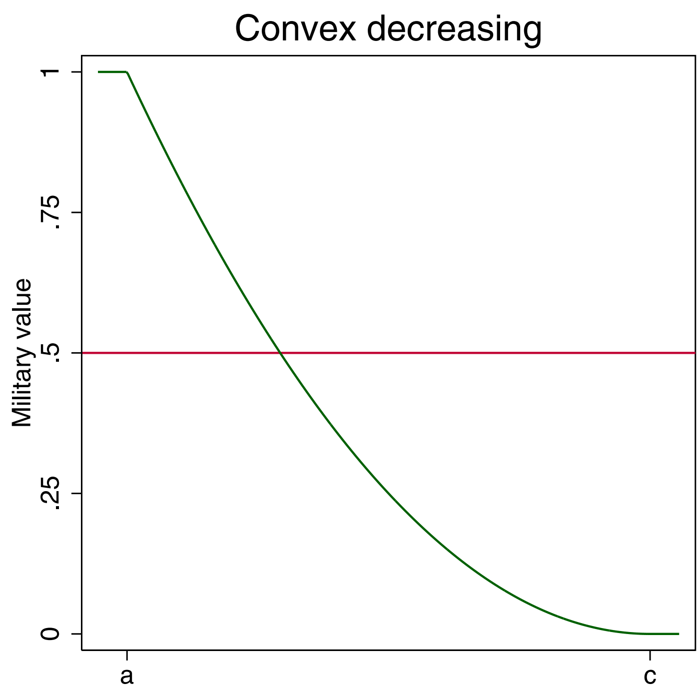

Figure K.4: Fuzzy function 4

K.1.4 Measures that worsen at slower and slower rates

FS4 is a downward sloping convex fuzzy set membership function with minimum and maximum parameters a and c, respectively, and an exponent parameter n that is greater than or equal to 1. FS4 describes groups of objects with membership values that decrease at a diminishing rate, i.e., decrease rapidly at first then decrease more and more slowly till they reach 0 at c as shown in figure K.4. The membership function for FS14 is

$$m_{(a,c,n)}(x) = \begin{cases} 1 & : \quad x < a \\ \left(\frac{c-x}{c-a}\right)^n & : \quad a \leq x < c; n \geq 1 \\ 0 & : \quad x \geq c \end{cases}$$

For example, let FS4 be the set of acceptable monthly off-base family housing costs. The fuzziness again here arises in trying to quantify the meaning of the word *acceptable* which varies from service member to service member and from decision-maker to decision-maker. Assume that

- Rents less than 100-percent of an service member's housing allowance are considered quite acceptable, $m_{(a,c,n)}(x) = 1$ for $x < 100$.
- Rents greater than or equal to 200-percent of an service member's housing allowance are considered undesirable so $m_{(a,c,n)}(x) = 0$ for $x \geq 200$.

These are estimates of the minimum and maximum parameters: $a = 100$ percent and $c = 200$ percent. Again, unless there were reason to choose some other value, we would generally set the exponent $n = 2$.

If the average monthly base family housing costs in recent years across all applicants at a base has been $x = 146$ percent, then $m_{(100,200,2)}(146) = 0.292$

K.1.5 Measures that worsen smoothly

This fuzzy set membership function, FS5, a downward sloping S-shaped function, may have its median, b, mid way between its minimum and its maximum parameters, a and c, as shown on the left in figure K.5. FS5's median may be some other point between the two extremes, as shown on the right. FS5 describes groups of objects with membership values that decrease very slowly at both extremes but decrease rapidly for intermediate elements. The membership function for FS5 is

$$m_{(a,b,c)}(x) = \begin{cases} 1 & : \quad x < a \\ 1 - \frac{1}{2}\left(\frac{x-a}{b-a}\right)^2 & : \quad a \leq x < b \\ \frac{1}{2}\left(\frac{c-x}{c-b}\right)^2 & : \quad b \leq x < c \\ 0 & : \quad x \geq c \end{cases}$$

As example, let FS5 be the set of military bases that are close to a private hospital. The fuzziness arises in trying to quantify the meaning of the word *close*. Assume that

- Bases closer than 20 miles to a private hospital are considered unquestionably close to it, membership $m_{(a,b,c)}(x) = 1$ for $x < 20$ miles.
- Bases farther away than 60 miles from the nearest private hospital are decidedly not close, membership $m_{(a,b,c)}(x) = 0$ for $x \geq 60$ miles.

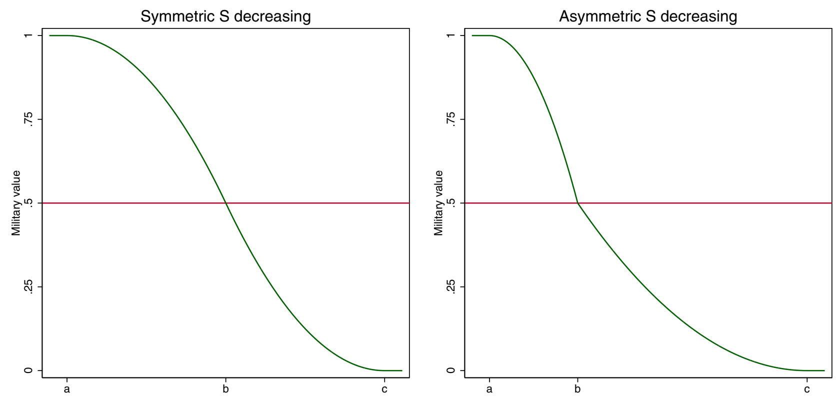

Figure K.5: Fuzzy function 5

- In terms of convenient access, bases 40 miles from a private hospital are considered moderately close, membership $m_{(a,b,c)}(40) = 0.5$.

Reasonable estimates of the minimum, median, and maximum parameters would be as follows: $a = 20$ miles, $b = 40$ miles, and $c = 60$ miles, respectively. If a particular military base is 32 miles from the nearest hospital, then its membership in FS5 is: $m_{(20,40,60)}(32) = 0.820$

K.2 Large values are desirable

In this section, we describe a set of fuzzy functions that may be used in cases for which larger values are desirable and smaller values are not desirable.

K.2.1 Binary measures

This set membership function, FS6, an upward step function, is the opposite of the downward sloping FS2 described above. A picture of this function is shown in figure K.6. The membership function for FS6 is given by:

$$m_{(d)}(x) = \begin{cases} 0 & : \quad x \leq d \\ 1 & : \quad x > d \end{cases}$$

As an example, let FS6 be the set of military bases at sites that have at least one 12,000 foot runway. There is no fuzziness associated with which

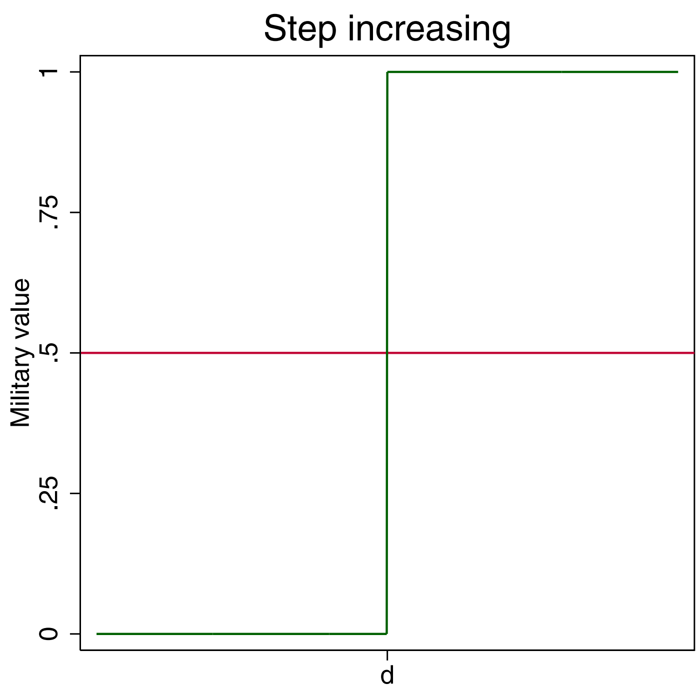

Figure K.6: Fuzzy function 6

bases are included in this set. Base membership increases as we go to longer runways so we would choose FS6 over the downward step FS2 described above. Assume that

- A bases with no runway or runways shorter than 12,000 feet long don't meet the requirement for inclusion in the set so $m_{(d)}(x) = 0$ for $x \leq 12,000$.

- A base with one or more runways 12,000 ft or more long would be included in the set so $m_{(d)}(x) = 1$ for $x > 12,000$.

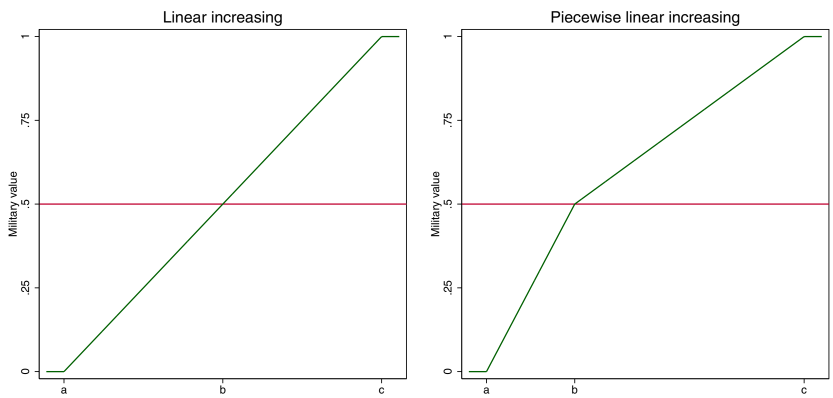

Figure K.7: Fuzzy function 7

K.2.2 Measures that improve at a constant rate

This fuzzy set membership function, FS7, an upward sloping linear function, may have its median, b, mid way between its minimum and its maximum parameters, a and c, as shown in figure K.7 on the left. Or, FS7's median may be some other point between the two extremes, as shown on the right in figure K.7. FS7 describes groups of objects with membership values that increase at an approximately constant rate. The membership function for FS7 is

$$m_{(a,b,c)}(x) = \begin{cases} 0 & : \quad x < a \\ \frac{1}{2}\left(\frac{x-a}{b-a}\right) & : \quad a \leq x < b \\ 1 - \frac{1}{2}\left(\frac{c-x}{c-b}\right) & : \quad b \leq x < c \\ 1 & : \quad x \geq c \end{cases}$$

Let FS7 be the set of military bases in communities where good, local retail shopping available. The fuzziness arises in trying to quantify the meaning of the words *good retail shopping*. Assume that

- Bases in communities where there are 20 or fewer retailers within a 10 mile radius are not considered to have good shopping opportunities so $m_{(a,b,c)}(x) = 0$ for $x < 20$.

- Bases in communities where there are 100 or more retailers within a 10 mile radius are considered to provide a full range of local, competitive shopping opportunities, i.e., good shopping, so $m_{(a,b,c)}(x) = 1$ for $x \geq 100$.

- Bases in communities where there are 50 or more retailers within a 10 mile radius are considered to provide moderately good local shopping, so $m_{(a,b,c)}(50) = 0.5$.
- We would set the minimum, median, and maximum parameters: $a = 10$, $b = 50$, and $c = 100$ retailers within 10 miles, respectively.

If 61 retail establishments are near a base, then $m_{(20,50,100)}(61) = 0.610$

K.2.3 Measures that improve at faster and faster rates

FS8 is an upward sloping convex fuzzy set membership function with minimum and maximum parameters, a and c, respectively, and an exponent parameter n that is greater than or equal to 1. FS8 describes groups of objects with membership values that increase at an accelerating rate, i.e., increase very slowly at first then increase more rapidly till they reach 1 at c as shown in figure K.8. The membership function for FS8 is

$$m_{(a,c,n)}(x) = \begin{cases} 0 & : \quad x < a \\ \left(\frac{x-a}{c-a}\right)^n & : \quad a \leq x < c; n \geq 1 \\ 1 & : \quad x \geq c \end{cases}$$

Let FS8 be the set of off-base family housing units generally experiencing acceptable peak ambient noise levels. The fuzziness here arises in trying to quantify the meaning of the words *acceptable noise level* which varies with the distance housing units are from the base's live-ordinance firing range and from decision-maker to decision-maker. Assume that

- Peak noise levels at housing units less than one nautical mile from the base firing range are considered totally unacceptable by essentially everyone. In this case, $m_{(a,c,n)}(x) = 0$ for $x < 1$.
- Peak noise levels at housing units six nautical miles or farther from the base firing range are generally considered quite acceptable by essentially everyone. In this case, $m_{(a,c,n)}(x) = 1$ for $x \geq 6$.

These are estimates of the minimum and maximum parameters: $a = 1$ and $c = 6$. For simplicity, we again use $n = 2$. Suppose the average distance from the firing range to off-base housing in recent years has been 3.6 nautical miles at a base, then $m_{(1,6,2)}(3.6) = 0.270$

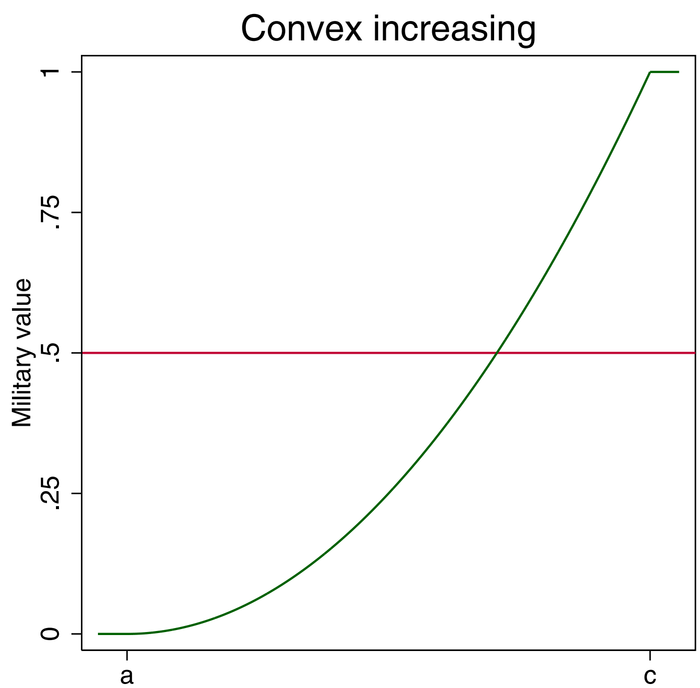

Figure K.8: Fuzzy function 8

K.2.4 Measures that improve at slower and slower rates

FS9 is an upward sloping concave fuzzy set membership function with minimum and maximum parameters a and c, respectively, and an exponent parameter n that is greater than or equal to 1. FS9 describes groups of objects with membership values that increase at an diminishing rate, i.e., increase rapidly at first then increase more and more slowly till they reach 1 at c as shown in figure K.9. The membership function for FS9 is

$$m_{(a,c,n)}(x) = \begin{cases} 0 & : \quad x < a \\ 1 - \left(\frac{c-x}{c-a}\right)^n & : \quad a \leq x < c; n \geq 1 \\ 1 & : \quad x \geq c \end{cases}$$

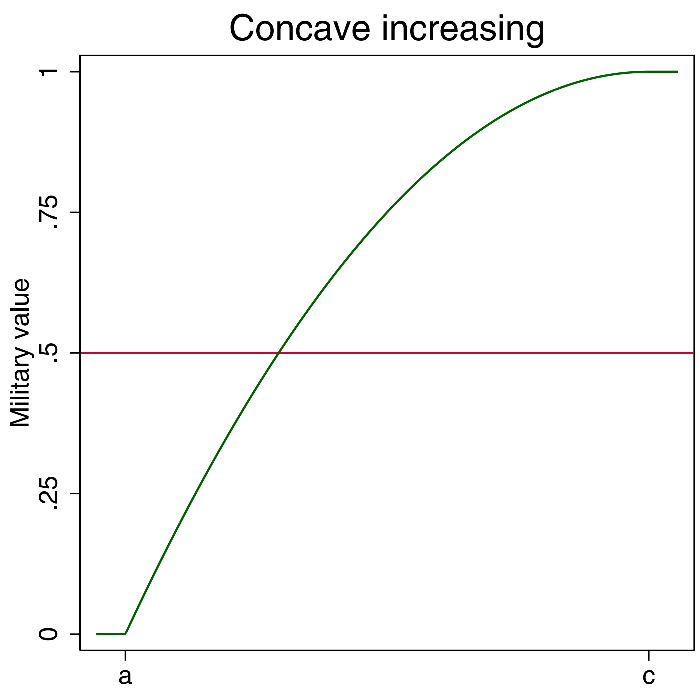

Figure K.9: Fuzzy function 9

Let FS9 be the set of bases with good emergency medical services. The fuzziness arises in trying to quantify the meaning of the word *good* which varies from potential patient to potential patient and from decision-maker to decision-maker. Assume that

- Fewer than one paramedic or physician being available or on-quick-response call 24/7 is deemed unacceptable by most people, so we have $m_{(a,c,n)}(x) = 0$ for $x < 1$.
- Three or more paramedics or physicians being available or on-quick-response call 24/7 is deemed very good to excellent service by most people, so $m_{(a,c,n)}(x) = 1$ for $x \geq 3$.

These are estimates of the minimum and maximum parameters: a equal one

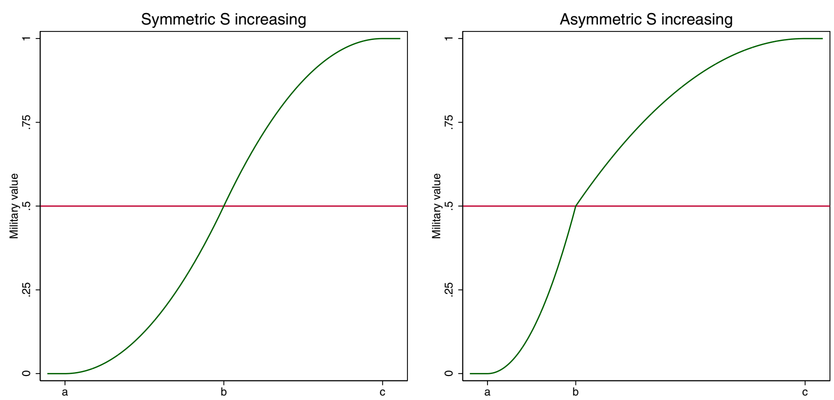

Figure K.10: Fuzzy function 10

paramedic or physician available 24/7 and $c = 3$ paramedics or physicians. We again set the exponent $n = 2$.

If the average minimum number of skilled medical service providers available in any 24 hour period at a base is $x = 1.15$ persons, then the base's membership in FS9 is $m_{(1,3,2)}(1.15) = 0.144$

K.2.5 Measures that improve smoothly

This upward sloping, S-shaped fuzzy set membership function, FS10, may have its median, b, mid way between its minimum and its maximum parameters, a and c, as shown in figure K.10 on the left. Or, FS10's median may be some other point between the two extremes, as shown on the right. FS10 describes groups of objects with membership values that increase very slowly at both extremes but increase rapidly for intermediate elements. The membership function for FS10 is given by:

$$m_{(a,b,c)}(x) = \begin{cases} 0 & : \quad x < a \\ \frac{1}{2}\left(\frac{x-a}{b-a}\right)^2 & : \quad a \le x < b \\ 1 - \frac{1}{2}\left(\frac{c-x}{c-b}\right)^2 & : \quad b \le x < c \\ 1 & : \quad x \ge c \end{cases}$$

Consider the set of military bases with live ordinance training capabilities. Among other requirements, these bases must be far removed from communities with high population densities. Though fuzziness exists in two dimensions (both in being far removed and in specifying high population density), consider the meaning of *far removed*. Assume that

- Bases closer than 5 miles to high density populations are considered too close to allow conducting live-fire training, so $m_{(a,b,c)}(x) = 0$ for $x < 5$ miles.
- Bases farther away than 10 miles from high density populations are decidedly far removed, so $m_{(a,b,c)}(x) = 1$ for $x \geq 10$.
- Noise and pollution from live-fire training at least 8 miles away are generally to be within acceptable levels. To reflect these noise and pollution standards we peg the fuzzy set membership $m_{(a,b,c)}(8) = 0.5$.

With these assumptions, we set the minimum, median, and maximum parameters to $a = 5$, $b = 8$, and $c = 10$ miles, respectively. Under these assumptions, if the distance to the nearest high density population area from a base of interest is six miles, $m_{(5,8,10)}(6) = 0.056$.

K.3 Intermediate values are desirable

In this section, we define fuzzy functions that are appropriate in cases for which the desirable values are in a range and values outside this range are less desirable.

K.3.1 Binary measures

This set membership function, FS11, an upward step function followed by a downward step function, is another example of a crisp set, i.e., it has a binary membership function. FS11 describes groups of objects with a well-defined range for the parameter that determines its membership. The membership function for FS11 is a concatenation of FS6 and FS2:

$$m_{(d)}(x) = \begin{cases} 0 & : & x \leq d \\ 1 & : & d < x \leq D \\ 0 & : & x > D \end{cases}$$

If we define the set of military bases at sites that offer limited live-fire training opportunity to be bases that have 1, 2, or 3 live ordinance firing ranges, we have the following relationships.

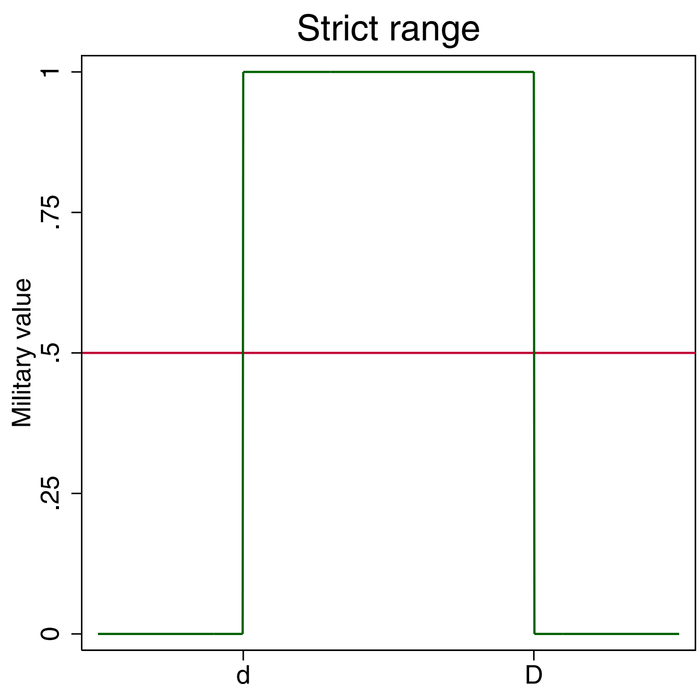

Figure K.11: Fuzzy function 11

- Bases with no live fire ranges are by definition not in the set, $m_{(d,D)}(0) = 0$.

- Bases with 1, 2,or 3 live fire ranges are in the set so $m_{(d,D)}(x) = 1$ for $x = 1, 2, 3$.

- Bases with more than 3 live fire ranges can usually schedule all range time when requested so are not in the limited opportunity set, i.e., $m_{(d,D)}(x) = 0$ for $x > 3$.

As described here, the parameters are $d = 1$ and $D = 3$.

K.3.2 Measures that improve and then worsen at constant rates

This fuzzy set membership function, FS12, an upward sloping linear function followed by a downward sloping linear function, may have its medians, b on the positive slope and B on the negative slope, mid way between its minimum and its maximum parameters, a and c and A and C, respectively, as shown in figure K.12 on the top. Or, FS12's medians may be some other points between the two pairs of extremes, as shown in the figure on the bottom. FS12 describes groups of objects with membership values that increase then decrease at approximately constant rates. The membership function for FS12 is

$$m_{(a,b,c)}(x) = \begin{cases} 0 & : \quad x < a \\ \frac{1}{2}\left(\frac{x-a}{b-a}\right) & : \quad a \leq x < b \\ 1 - \frac{1}{2}\left(\frac{c-x}{c-b}\right) & : \quad b \leq x < c \\ 1 & : \quad c \leq x < A \\ 1 - \frac{1}{2}\left(\frac{x-A}{B-A}\right) & : \quad A \leq x < B \\ \frac{1}{2}\left(\frac{C-x}{C-B}\right) & : \quad B \leq x < C \\ 0 & : \quad x \geq C \end{cases}$$

Let FS12 be the set of military bases with good employment opportunities for military dependents. On small bases, the government provides many of the ancillary services employing many military dependents. For small bases, then, civilian job opportunities are numerous and turnover rates, being tied to military tour cycles, are high. On large bases, private contractors tend to deliver most of the ancillary services using their own permanent work forces. For large bases, then, civilian job opportunities for military dependents are few and the turnover rates are low. To capture this in a fuzzy membership function, we assume that the number of employed military dependents first increases directly with increases in the overall base populations, reaches a plateau, then decreases proportionally with further increases in base populations. The fuzziness arises in trying to quantify the meaning of the notions of *good* military dependent employment opportunities and *small* and *large* base populations. Assume that

- Bases with total duty personnel of 800 or fewer have very few dependent employment opportunities, $m_{(a,b,c,A,B,C)}(x) = 0$ for $x < 800$.
- Bases with total duty personnel between 800 and 3,000 have an in-

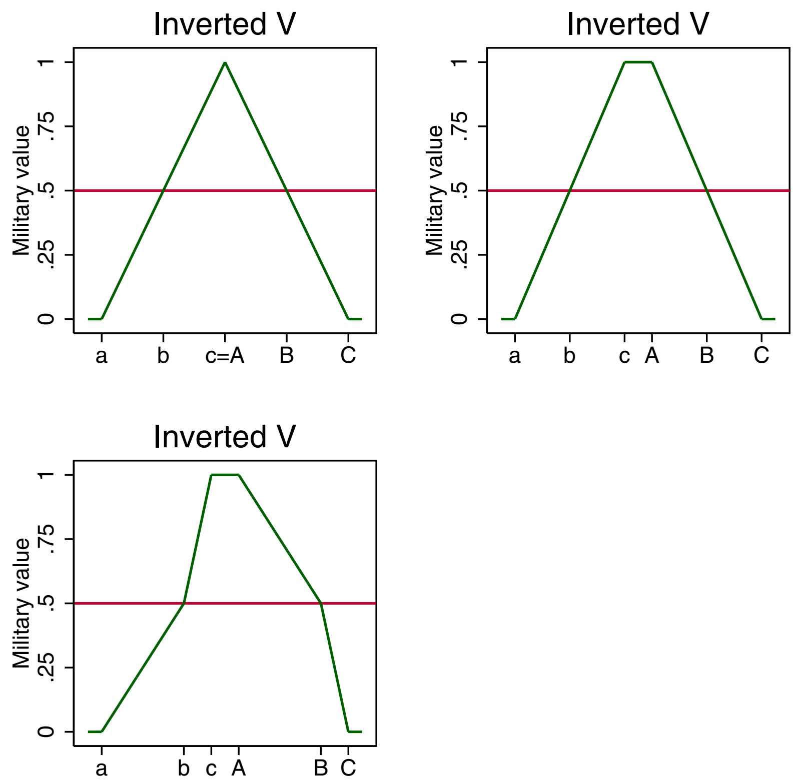

Figure K.12: Fuzzy function 12

creasing variety of civilian job openings roughly in proportional in number to the population of duty personnel.

- Bases with total duty personnel between 3,000 and 12,000 have a wide variety and typically a significant number of civilian job openings, i.e., virtually all dependents applicants can find work, $m_{(a,b,c,A,B,C)}(x) = 1$ for $3,000 \leq x < 12,000$.

- Demand for military dependent workers on bases with total duty personnel between 12,000 and 30,000 steadily declines at a roughly constant rate as the number of duty personnel goes up.

- Bases with total duty personnel of over 30,000 or more contract out most base services and have relatively few military dependent employment opportunities on base, $m_{(a,b,c,A,B,C)}(x) = 0$ for $x \geq 30,000$.

These are estimates of the minimum, median, and maximum parameters: $a = 800$ duty personnel, $b = \frac{800+3,000}{2} = 1,900$, and $c = 3,000$ duty personnel, respectively for small bases. For larger bases contracting out more and more jobs, the estimated minimum, median, and maximum parameters for the declining opportunities are $A = 12,000$ duty personnel, $B = \frac{12,000+30,000}{2} = 21,000$, and $C = 30,000$ duty personnel, respectively.

If the usual number of duty personnel over time is approximately $x = 27,000$ at a particular base, then $m_{(800,1900,3000,12000,21000,30000)}(27,000) = 0.167$

K.3.3 Measures that smoothly improve and then smoothly worsen

This fuzzy set membership function, FS13, a possibly asymmetric, bell shaped function, may or may not have a plateau at its peak as shown in figure K.13 on the right and left, respectively. FS13 describes groups of objects with membership values that increase at first slowly, then more rapidly, and then slower before reaching their peak after which they decline in similar fashion. The membership function for FS13 is given by:

$$m_{(a,b,c,A,B,C)}(x) = \begin{cases} 0 & : \ x < a \\ \frac{1}{2}\left(\frac{x-a}{b-a}\right)^2 & : \ a \leq x < b \\ 1 - \frac{1}{2}\left(\frac{c-x}{c-b}\right)^2 & : \ b \leq x < c \\ 1 & : \ c \leq x < A \\ 1 - \frac{1}{2}\left(\frac{x-A}{B-A}\right)^2 & : \ A \leq x < B \\ \frac{1}{2}\left(\frac{C-x}{C-B}\right)^2 & : \ B \leq x < C \\ 0 & : \ x \geq C \end{cases}$$

For example, let FS13 be the set of military bases that offer desirable off-base housing options. The fuzziness arises in quantifying the meaning of the word *desirable*. It seems generally true that the commuting conveniences, the transient character of neighborhoods, and the sometimes higher cost to quality accommodation ratios associated with being close to a major military base must be balanced against the higher transportation costs, the generally greater community diversity, and the lower accommodations costs associated with living far from the base. Individuals make these kinds of personal

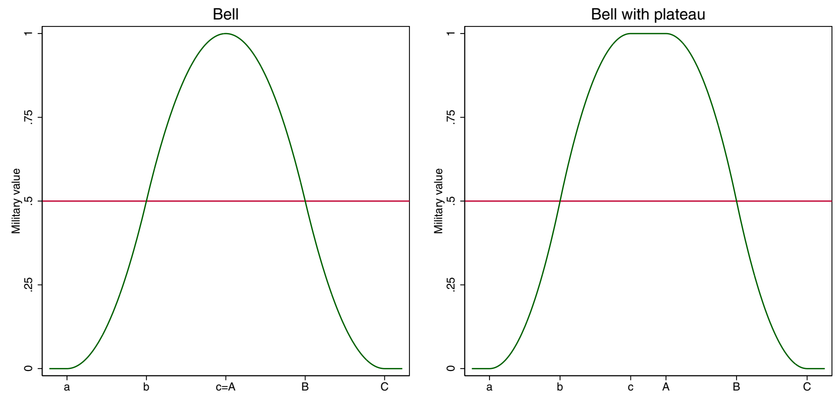

Figure K.13: Fuzzy function 13

preference choices differently, but over a large base-affiliated population, the demand for housing is expected to be bell shaped as one gets farther and farther from base. Assume a continuum[2] such that

- Housing closer than 0 miles from base is, by definition, undesirable for those who want to live off-base so $m_{(a,b,c,A,B,C)}(0) = 0$.
- Housing farther than 40 miles from base, is also undesirable even for those who want to live off-base so $m_{(a,b,c,A,B,C)}(x) = 0$ for $x \geq 40$.
- The most desirable housing is located in a band around a base from 5 to 12 miles away so $m_{(a,b,c,A,B,C)}(x) = 1$ for $5 \leq x < 12$.

These assumptions give us the needed estimates of the minima, medians, and maxima for the up hill and down hill parameters: $a = 0$ miles, $b = \frac{0+5}{2} = 2.5$ miles, $c = 5$ miles, $A = 12$ miles, $B = \frac{12+40}{2} = 26$ miles, and $C = 40$ miles.

If we let the average one-way commuting time, t, be a proxy for the distribution of local housing, then a base having $t = 35$ minutes and a typical as-the-crow-flies commute speed of 30 mph would have a base-to-housing distance $x = 17.5$ miles. In this case, we would have $m_{(0,2.5,5,12,26,40)}(17.5) = 0.923$

[2]There are no discontinuities at the two medians which implies $b = \frac{a+c}{2}$ and $B = \frac{A+C}{2}$.

References

[1] *DOD Base Closure and Realignment Report to the Commision: Department of the Navy Analyses and Recommendations (Volume IV).* Report, Department of Defense, March 1993

[2] *Military Bases: Analysis of DOD's Recommendations and Selection Process for Closures and Realignments.* Report GAO/NSIAD-93-173, General Accounting Office, April 1993

[3] *DOD Base Closure and Realignment Report.* Report, Department of Defense, March 1995

[4] *DOD Base Closure and Realignment Report to the Commision: Department of the Navy Analyses and Recommendations (Volume IV).* Report, Department of Defense, March 1995

[5] *Military Bases: Analysis of DOD's 1995 Process and Recommendations for Closure and Realignment.* Report GAO/NSIAD-95-133, General Accounting Office, April 1995

[6] *Military Bases: Lessons Learned from Prior Base Closure Rounds.* Report GAO/NSIAD-97-151, General Accounting Office, July 1997

[7] J. L. Anderson, S. W. Belcher, J. F. Nance, R. H. Nickel, and D. M. Wennergren. *CNA Support of the DON BRAC-95 Process: Methodologies and Analyses.* CNA Research Memorandum CRM 95-235, Center for Naval Analyses, 4401 Ford Avenue, Alexandria, VA 22302-1498, December 1995

[8] D. J. Barrett and R. E. Silverman. *SSH, The Secure Shell: The Definitive Guide.* Sebastopol, CA: O'Reilly & Associates, Inc., 2001

[9] J. J. Buckley, E. Eslami, and T. Feuring. *Fuzzy Mathematics in Economics and Engineering.* Physica-Verlag, 2002

[10] C. Carlsson and R. Fuller. *Fuzzy Reasoning in Decision Making and Optimization*. Physica-Verlag, 2002

[11] R. Fourer, D. M. Gay, and B. W. Kernighan. *AMPL: A Modeling Language for Mathematical Programming*. Pacific Grove, CA: Brooks/Cole–Thomson Learning, 2003, second ed.

[12] ILOG. *ILOG CPLEX 7.1: User's Manual*, March 2001

[13] S. Kotz and N. L. Johnson, editors. *Encyclopedia of Statistical Sciences*. John Wiley & Sons, 1983

[14] M. Sakawa. *Large Scale Interactive Fuzzy Multiobjective Programming: Decomposition Approaches*. Physica-Verlag, 2000

[15] J. Ts, R. Eckstein, , and D. Collier-Brown. *Using Samba*. Sebastopol, CA: O'Reilly & Associates, Inc., 2003, second ed.

[16] J. Vesperman. *Essential CVS*. Sebastopol, CA: O'Reilly & Associates, Inc., 2003

[17] M. Welsh, M. K. Dalheimer, T. Dawson, and L. Kaufman. *Running Linux*. Sebastopol, CA: O'Reilly & Associates, Inc., 2003, fourth ed.

[18] L. A. Zadeh. "Fuzzy sets." *Information and Control*, 1965: 8:338–353

[19] L. A. Zadeh. "Fuzzy algorithms." *Information and Control*, 1968: 12:94–102

[20] H.-J. Zimmerman. *Fuzzy Set Theory–And its Applications, 4th ed.* Kluwer Academic Publishers, 2001

List of Figures

List of Tables