NAVAL POSTGRADUATE SCHOOL
Monterey, California

THESIS

RANGER AIR LOAD PLANNER

by

Maximo A. Moore III

June 2000

Thesis Advisor:	Gordon H. Bradley
Second Reader:	Gerald G. Brown

Approved for public release; distribution is unlimited.

DTIC QUALITY INSPECTED 4

20000807 071

REPORT DOCUMENTATION PAGE		Form Approved OMB No. 0704-0188

Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instruction, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188) Washington DC 20503.

1. AGENCY USE ONLY *(Leave blank)*	**2. REPORT DATE** June 2000	**3. REPORT TYPE AND DATES COVERED** Master's Thesis
TITLE AND SUBTITLE : Ranger Air Load Planner		**5. FUNDING NUMBERS**
6. AUTHOR(S) Maximo A Moore III		
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) Naval Postgraduate School Monterey, CA 93943-5000		**8. PERFORMING ORGANIZATION REPORT NUMBER**
9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES) Air Force Office of Scientific Research 801 North Randolph Street Arlington, VA 22203-1977		**10. SPONSORING / MONITORING AGENCY REPORT NUMBER**

11. SUPPLEMENTARY NOTES The views expressed in this thesis are those of the author and do not reflect the official policy or position of the Department of Defense or the U.S. Government.

12a. DISTRIBUTION / AVAILABILITY STATEMENT Approved for public release; distribution is unlimited.	**12b. DISTRIBUTION CODE**

ABSTRACT *(maximum 200 words)*

The United States Army 75th Ranger Regiment conducts combat parachute operations as part of United States Special Operations Command (USSOCOM). The Rangers are the largest deployable asset of USSOCOM, and are required to plan and execute large-scale parachute assaults into hostile theaters with little or no notice. Generally fighting numerically superior enemy, far from the support of the conventional Army, Rangers must arrive capable of self-sustaining combat operations in any operational environment. This thesis provides Ranger air load planners a tool to rapidly plan feasible mission equipment loads. The Ranger Air Load Planner (RAP) is simple to learn and operate, provides load plans selected from, pre-approved, United States Air Force load templates, and supports dynamic decision support with rapid solution response. An optimization model is used in the thesis to objectively assess the quality of RAP load plans. RAP is a working product that can be adapted for use in air load mission planning by all units under USSOCOM.

14. SUBJECT TERMS Mission Planning, Analysis, Special Operations, MPARE, Java, Loosely Coupled Components	**15. NUMBER OF PAGES** 82
	16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
Unclassified	Unclassified	Unclassified	UL

NSN 7540-01-280-5500

Standard Form 298 (Rev. 2-89)
Prescribed by ANSI Std. 239-18

Approved for public release; distribution is unlimited.

RANGER AIR LOAD PLANNER

Maximo A. Moore III
Captain, United States Army
B.S., State University of New York College at New Paltz, 1992

Submitted in partial fulfillment of the
requirements for the degree of

MASTER OF SCIENCE IN OPERATIONS RESEARCH

from the

NAVAL POSTGRADUATE SCHOOL
June 2000

Author: _____
Maximo A. Moore III

Approved by: _____
Gordon H. Bradley, Thesis Advisor

_____ ORSW
Gerald G. Brown, Second Reader

Richard E. Rosenthal
Richard E. Rosenthal, Chairman
Department of Operations Research

ABSTRACT

The United States Army 75th Ranger Regiment conducts combat parachute operations as part of United States Special Operations Command (USSOCOM). The Rangers are the largest deployable asset of USSOCOM, and are required to plan and execute large-scale parachute assaults into hostile theaters with little or no notice. Generally fighting numerically superior enemy, far from the support of the conventional Army, Rangers must arrive capable of self-sustaining combat operations in any operational environment. This thesis provides Ranger air load planners a tool to rapidly plan feasible mission equipment loads. The Ranger Air Load Planner (RAP) is simple to learn and operate, provides load plans selected from pre-approved, United States Air Force load templates, and supports dynamic decision support with rapid solution response. An optimization model is used in the thesis to objectively assess the quality of RAP load plans. RAP is a working product that can be adapted for use in air load mission planning by all units under USSOCOM.

DISCLAIMER

The reader is cautioned that the computer programs developed in this research may not have been exercised for all cases of interest. While every effort has been made, within time available, to ensure that the programs are free of computational and logic errors, they cannot be considered validated. Any application of these programs without additional verification is at the risk of the planner.

TABLE OF CONTENTS

I. INTRODUCTION 1

A. BACKGROUND 1

B. AN AIRBORNE RANGER MISSION 3

C. HOW AIR LOAD PLANS ARE MADE 4

D. TIME IS THE ENEMY 9

E. PROPOSED AUTOMATED AIR LOAD PLANNER 10

F. THESIS ORGANIZATION 10

II. SOLUTION STRATEGY 11

A. SOLUTION OPTIONS 11

B. OBJECTIVE FUNCTION 12

C. TRANSPARENCY 12

D. AND THE WINNER IS... 13

III. DATA 15

A. DATA REQUIREMENTS 15

B. LOAD TEMPLATES 15

C. SYNTHETIC TEMPLATE GENERATOR 18

1. Overview 18

2. Template Creation Algorithm 19

3. Template Synthesizer User Interface 22

IV. OPTIMIZATION FORMULATION 25

A. OPTIMIZATION MODEL OVERVIEW 25

B. OPTIMIZATION MODEL 25

V. HEURISTIC 31

A. INTRODUCTION 31

1. Description 31

2. Heuristic Solution Technique 31

3. Ranger Air Load Planner Overview 31

B. DETAILS OF THE HEURISTIC 32

1. Rudimentary Algorithm 32

2. Construction of Objective Function 32

3. Solution Sequence 36

C. THE HEURISTIC PRESENTED AS AN ABSTRACT ALGORITHM 38

D. GRAPHICAL USER INTERFACE 44

1. Template Data Input 44

2. Aircraft and Cargo Item Input 44

3. Aircraft Loading Guidance 45

4. Penalties and Diversity 47

5. Relative Equipment Values 47

VI. IMPLEMENTATION AND ANALYSIS 49

A. SAMPLE PROBLEM 49

B. DATA 49

1. Aircraft 49

2. Template Data 50

3. Cargo Items 51

C. OPTIMIZATION IMPLEMENTATION 51

D. RANGER AIR LOAD PLANNER IMPLEMENTATION 52

E. COMPUTATIONAL RESULTS 52

1. Comparison of the Optimization and Ranger Air Load Planner 52

2. Runtime Complexity of the Optimization and Ranger Air Load Planner 54

VII. WHEN THINGS GO WRONG 55

A. HEURISTICS CAN BE RISKY 55

B. RESTRICTED SAMPLE PROBLEM 56

C. COMPUTATIONAL RESULTS 56

1. The Optimization Works 56

2. But RAP Fails 56

D. PLANNER GUIDANCE IS ALWAYS KEY 57

1. How the Planner Can Control Loads 57

2. Suggesting Specific Templates By Locking Guidance 58

VIII. CONCLUSIONS AND RECOMMENDATIONS 61

A. CONCLUSIONS 61

B. RECOMMENDATIONS 62

LIST OF REFERENCES 63

INITIAL DISTRIBUTION LIST 65

LIST OF ABBREVIATIONS AND/OR ACRONYMS

SOC	Special Operations Command
MEL	Minimum Equipment List of mission-essential gear
DEL	Desired Equipment List of mission-desirable gear
Deficit Cargo	MEL cargo not loaded
Frustrated Cargo	DEL cargo not loaded
Load Template	List of items and amounts that fit in an aircraft type
Approved Load Template	USAF agrees the items fit and are airworthy
ATL	Approved Template List
NALT	Non-approved Template List
TLM	Type Load Method gives instructions for designing a non-approved load template
Template Diversity	The number of different items in a template
RAP	Ranger Air Planner
Greedy	Choice made for immediate gain
Asphalt Cargo	Cargo not yet loaded
GUI	Graphical User Interface
S3A	Responsible for air load planning
Locked Goal	Inflexible limit on the number of an item to load on an aircraft
Loaded Value	Total military worth of loaded items
Relative Equipment Value	Item values expressed as a fraction of the most valuable item

Template Best Value	The best total relative equipment value of any approved template
Floating Diversity Penalty Function	A heuristic reward for loading a proportionate share of items remaining to load on aircraft remaining to be loaded
Mission Diversity Value	Weight the planner assigns to diversity

EXECUTIVE SUMMARY

The United States Army 75th Ranger Regiment must deploy anywhere in the world in less than 24 hours to conduct combat operations as part of the United State Special Operations Command (USSOCOM). Using specially configured Air Force transport aircraft, the Rangers must move men and equipment from the continental United States to an area of operations and land ready to sustain combat operations. The primary entry method is parachute assault.

Additionally, to ensure aircraft airworthiness, Air Force representatives must approve all loads prior to takeoff. The Air Force and Rangers have agreed on a classified list of approved aircraft loads that, if adhered to, will guarantee load plan acceptance by the Air Force. The manual air load planning technique used today generally takes one man eight hours to complete.

While preparing to conduct any mission, Rangers rehearse every aspect of an operation until all leaders are satisfied the operational plan is well understood by all Rangers. Assembly after a parachute infiltration is a key rehearsal normally conducted several times for clarity. However, the exact assembly plan rehearsal cannot take place until the air load plan is approved by the Ranger chain of command. Time for rehearsal is critical.

This thesis develops an automated air load planning aid for the 75th Ranger Regiment. The Ranger Air Load Planner (RAP) generates air load plans and presents them visually. The planner can view candidate loads and suggest loading guidance to shift the numbers of cargo items between aircraft, thus tailoring load plans to meet specific mission requirements or just follow the better judgment of the planner.

Using a Ranger battalion-sized example, RAP has been tested against an optimization model, also created for this thesis, and both construct feasible load plans that are essentially indistinguishable. RAP generates load plans in less than a second on a personal computer. RAP

is easy to understand and the software is easy to manipulate. This allows Ranger planners to audit the air load plan and explain the what and why of it to all Rangers. Written in Java, the heuristic can be maintained by a Java programmer with modest experience in data structures and computation, and does not require the skills of a trained analyst.

Before the Rangers can adopt RAP in air load planning, software validation should include testing with classified data on past missions.

Because RAP performs thousands of times faster than the current manual method, it can be used early in the air load planning process to quickly suggest feasible solutions to the air load problem, perhaps while the mission plan is still in flux. Any candidate plan suggested by RAP provides a foundation the Rangers could use to develop the final air load plan, or to suggest improvements to the mission manifest to make the assault more effective.

RAP's major attraction is its speed. If RAP were validated for use, Rangers would be able to disseminate air load plans quickly and accurately to subordinate units, and guarantee that the Air Force would accept the approved load plan.

I. INTRODUCTION

A. BACKGROUND

On 17 December 1989, the National Command Authority decided to commit specially trained airborne units to military action in Panama. The President established H-hour for 0100 on 20 December, just three days after the decision to intervene. The complex operation was centrally planned due to the need for thoroughly synchronized operations. The mission assigned to the airborne force was to quickly isolate, neutralize, and, if needed, destroy units of the Panamanian Defense Force [PDF] by overwhelming combat power. These forces were then to link up elements of the 7th Infantry Division (Light), the 5th Infantry Division (Mechanized), and the 193d Infantry Brigade.

Deploying by strategic airlift from multiple bases in the continental United States, paratroopers jumped into action on two principal DZs [drop zones]. Ranger task forces seized airfields at Rio Hato and Torrijos-Tocumen Airport. Another task force built around 1st Brigade of the 82nd Airborne Division followed the Rangers. Their mission was to jump, assemble, and conduct immediate air assaults to eliminate PDF garrisons at Fort Cimarron, Tinajitas, and Panama Viejo. These initial offensive operations were later followed by ground combat and stability operations. They were sustained by air lines of communication from the US and by CSS [combat service support] units already in Panama.

Largely through airborne operations, capable and aggressive combined arms task forces were brought to bear on short notice against a dispersed enemy. Thirty-two separate objectives were attacked at the same time, paralyzing the enemy. The resounding success of Operation Just Cause was due mostly to the parachute assault and rapid follow-on missions possible by the airborne operation. Operation just Cause demonstrated once again the capability, flexibility, and value of airborne forces.

FM 90-26, Airborne Operations

The United States Army 75th Ranger Regiment operates as an elite conventional Airborne Infantry brigade in the US Special Operations Command (SOC). The Regiment is composed of three Battalions located in the United States. The 75th Regimental HQ (Regiment) is located at FT Benning, GA.

Rangers are wholly composed of US Army volunteer men who have excelled in conventional Army units. Ranger qualification is an arduous process that begins with the completion of Ranger School at FT Benning early in the Ranger's career. After completing US Army airborne school and at least one assignment, ranger leaders may

apply to Regiment. If a candidate meets the very high entry screening requirements, he is invited to the final assessment phase, Ranger Orientation Program. The Ranger Orientation Program (ROP) seeks to eliminate all but the most dedicated, physically strong, and mentally focused candidates. Unlike other units, at ROP officers and enlisted men are evaluated equally. Only the best are invited to become members of the 75th Ranger Regiment.

Regardless of rank, Rangers are always expected to behave and perform a cut above all other soldiers. The Ranger leadership has the unique ability to return a Ranger to the conventional Army for any lapse in conduct or performance. Self-cleansing allows the Regiment to operate at peak efficiency, an advantage not enjoyed by all Army units.

Rangers conduct combat operations in support of SOC. Ranger actions are characterized by lightning quick assaults on a numerically superior enemy in a hostile environment. These operations are generally conducted at night. All Ranger operations are raids, ambushes, reconnaissance, or cordon-searches.

In May 1999, the author accompanied two other Naval Postgraduate School students and two faculty members to FT Benning, GA. We received a detailed brief on the Ranger Regiment and the missions they conduct. The Regimental Executive Officer expressed the desire for better decision support tools in the field.

In September 1999, LTC Joel Parker, Naval Postgraduate School faculty member, and the author attended a Ranger training exercise. As part of this exercise, we observed a Ranger Battalion staff conduct air load planning. We observed how the air load and several contingency plans are rehearsed. The Rangers practice loading and unloading men and equipment into the aircraft that will carry them into the training area.

We met with various members of both the Regimental and Battalion staffs. These Rangers agreed that a planning assistance tool for air load planning would be welcome.

B. AN AIRBORNE RANGER MISSION

In September, 1999, the 75th Ranger Regiment conducted an airborne training exercise. The manual air load planning took one man 6-8 hours using a procedure that has not changed for 15 years. Planning for possible aircraft failure, an essential detail that provides an alternate plan for the failure of any single aircraft, requires more man-hours to complete.

The success of any military tactical operation depends on preparation before attempted execution. Tactical rehearsals prior to every operation allow Rangers to visualize their operation from start to finish. They prioritize, then practice key pieces of the operation until leaders are satisfied every Ranger knows his part of the operation.

One of the essential portions covered in an airborne pre-execution rehearsal is the assembly after the jump. Rangers units must assemble in a central area to account for all personnel and equipment after landing in a potentially hostile environment. If not rehearsed before an operation, unit assembly can be a time consuming and chaotic ordeal. The Rangers cannot rehearse movement from the landing site to the assembly area until the Regimental Air Operations Officer (S3A) tells them in which aircraft they will be arriving at the objective area.

The S3A seeks a faster, more effective, planning assistance tool. He wants a decision support aid to expedite his portion of the overall tactical plan. The S3A must quickly generate, seek approval for, and disseminate his load plan to allow subordinate units to schedule and conduct effective rehearsals.

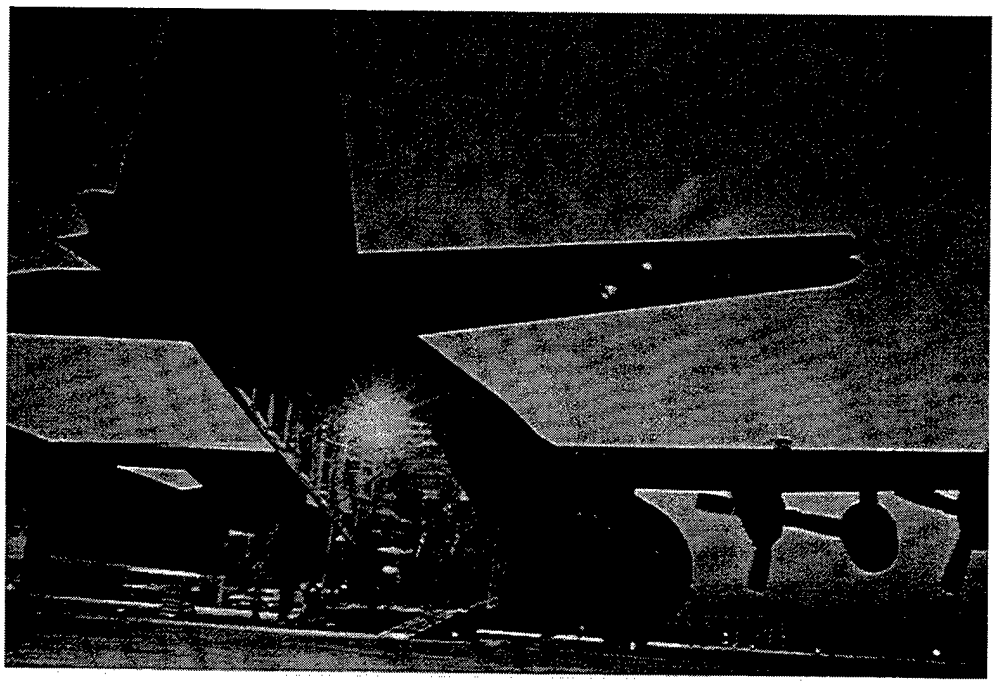

Figure 1.1. Air Load Rehearsal. Rangers conduct mission analysis upon receipt of a mission tasking from the US Special Operations Command. The Air Operations Officer (S3A) is responsible for load planning when a mission requires the Rangers to use aircraft as a method of infiltration. He establishes a load plan for each Air Force aircraft as a function of the mission, opposing enemy forces, terrain in the target area, friendly troops uncommitted, and time available. The Air Force must test each tentative load plan the S3A generates for proper fit prior to mission execution time. Proper load planning ensures Ranger men and equipment will fit when they practice actually loading. (Image from - [USAF, 1999])

C. HOW AIR LOAD PLANS ARE MADE

Once given an airborne mission, SOC staffs determine how many aircraft the Rangers will receive based on a SOC estimate of mission requirements. Any Air Force cargo aircraft equipped for airborne assaults are capable of performing all Ranger airborne missions. The Rangers conduct the military decision making process to determine force composition needed to accomplish the assigned mission [US Army, 1993].

The US Army uses the Type-Load Method (TLM) to build an aircraft-specific load template [US Army, 1990]. The Air Force load-masters and Rangers physically evaluate each template with actual equipment during airborne training events. Once they deem a load feasible for a specific type of aircraft, that particular load template is called *approved*. Each aircraft type has an *approved template list* (ATL). Loading aircraft from these lists saves the S3A time.

Approved templates are composed of the following cargo types:

Passengers(PAX) – personnel
Rolling stock (RS) – equipment such as trucks, and jeeps
Pallets (PL) – 463L pallets that ride in AF aircraft and occupy space on the ramp
Other (OT) – other space consuming cargo

The number of each cargo item that any given aircraft can hold is classified SECRET. Any mention herein of capabilities or equipment names is for problem illustration only.

This sample list of cargo items is not inclusive.

<u>PAX:</u>
Airborne(ABN) – PAX that jump
Airland (ALN) – PAX that air land
Departure Airfield Control Operators (DACO) – PAX that stay with the aircraft as a team
Tactical Operation Center (TOC) – Ranger command and control cells
Air Force Liaison (ALO) – Air Force command and control teams

<u>RS:</u>
Ranger Wheeled Vehicle (RSOV) – truck
Medical Wheeled Vehicle (MEDV) – ambulance
Wheeled Mortar Truck (MORTV) – mortar carrier
Motorcycle (BIKE) – light(LBIKE), heavy(HBIKE), 4 wheel (QUAD)
Air Force Transport(AFT) – modified wheeled vehicle
Helicopter (HELO) – any helicopter

PL:
Class I (CHOW) – pallet Army meals or water
Class V Supplies (AMMO) – Ammunition pallet
Class III Supplies (POL) – any petroleum, oil, or lubricant pallet
Class VIII Supplies (MED) – any medical supplies

OTHER
Door Bundles (DB) – small cargo item dropped with a parachute
Ramp Bundle (RB) – cargo item that is ramp deployed
Low Altitude Parachute Extraction - items deployed very close to the ground using a parachute
Patients (LITTER) – injured personnel that are incapable of moving.

Figure 1.2: A Chalk. An initial portion of a military operational planning cycle is devoted to determining what forces are required to conduct an operation. During the task organization assessment phase, smaller units are combined to form larger ones. For airborne operations, this can mean distributing these larger units over several aircraft so they exit their various aircraft over the same point on the ground. Careful load planning facilitates rapid ground assembly after the jump. During planning and execution, aircraft cannot be referenced by loaded unit names because these aircraft contain mixed units. Once assigned to each other, men, equipment, and one aircraft are called a chalk. A chalk is labeled with a unique integer number that designates its order of flight and the aircraft position on the parking ramp. For the reminder of the air movement phase into and out of an objective area, the parent organization remains organized in chalks. Generally, chalks into an objective area are composed differently than corresponding chalks coming out. (Image from - [USAF, 1999])

The S3A plans subordinate unit cargo lists for each aircraft, delineated by cargo items, using TLM. Each aircraft is assigned a chalk number, a reference for all personnel participating in the airborne operation. Once a load has been assigned to an aircraft, the aircraft is referred to by this chalk number only. Large airborne operations require an additional internal chalk number scheme. The internal chalk numbers, called stick numbers, refer to a specific group of parachutists in an aircraft and the order in which they will exit the aircraft.

The same aircraft will generally not carry the same cargo items both into and out of the objective area. The tactical situation generally requires that infiltration chalks be different TLM loads than exfiltration chalks. This aspect of Ranger operations makes them very time consuming to plan. The planner must complete two full independent load plans.

Rangers cross load aircraft. Cross loading requires that leaders and key equipment be separated across several chalks. Cross loading precludes the total loss of command and control or unit effectiveness in the event of aircraft failure.

Aircraft do fail. To ensure the Ranger commander has the minimal force he needs to accomplish his mission, the S3A must have an aircraft failure contingency plan, called a bump plan. A 1-bump plan is a contingency order to re-load cargo items from any single failed aircraft onto the remaining functional ones. Cargo that cannot be loaded into remaining aircraft is called aborted cargo. Aborted cargo is left behind.

After he has feasibly loaded all aircraft, the S3A must do two additional plans for each aircraft: a 1-bump infiltration plan and a 1-bump exfiltration plan. Assuming there is unused cargo capacity in the remaining aircraft, the S3A must plan to distribute as much

equipment as possible from damaged aircraft to functional ones. A successful 1-bump plan loads all cargo items from an incapable aircraft to mission capable ones, with minimal upset to previously planned loads.

The load plan must be approved prior to dissemination as an order. The load plan must attempt to simultaneously satisfy the identified mission requirements and the personal desires of all subordinate unit commanders. During load plan development, each subordinate element is given the opportunity to lobby for space on specific chalks based on its specific mission assessments. Subordinate commanders suggest load plan changes until the parent unit commander's delegate has approved the plan. The S3A also receives requests and directives from external agencies that must be incorporated into each load plan. For example, the Air Force may request space on chalk 2 for an additional five-man Air Liaison Team.

Generally, a load plan iterates several times between the subordinate unit commanders and staff to the S3A before it can be finalized. Each suggested change must be evaluated by the S3A. Superficially simple proposed changes can have complicated consequences with a complete plan. The movement of one cargo item to accommodate a suggested change may upset the load of several chalks.

New Ranger missions frequently necessitate new load plans not in their approved template list. The Ranger "Standard Operating Procedure" has a set of tactical air load planning rules to apply in designing new load plans [US Army, 1990]. The S3A must follow this rule set or a new load will not be tactically or physically feasible. An example rule could be, 'Never load fewer than two ABN or ALN from the same unit' or 'Do not

load more than five RSOVs from the same unit on any one plane of this type.' The rules change with new Ranger equipment and new operational techniques.

The S3A must also consider the effect of new loads on the bump plan. Consider a given aircraft and a new cargo item that restricts PAX movement in its cargo hold. Any aircraft failure that requires getting PAX from this chalk to another would prove difficult unless the new item is loaded forward in the hold. Something like this would not be discovered until the Rangers practice loading the aircraft late in the mission planning cycle.

D. TIME IS THE ENEMY

Air load planning is an art. The iterative nature of air load planning makes concise mathematical definition of good or better air load plans problematic. When many leaders are involved, it becomes very difficult to quantify what makes one solution better than another. The operational plan requirements may change several times before the commander finalizes them. The S3A may be required to modify his initial load solution until all subordinate units commanders are satisfied or planning time expires. The load planner must additionally respond to external agencies that may require special accommodation very late in air load planning.

The enemy of all operational planning is time. Given enough time, the manual Ranger Airborne planning technique provides solutions to accomplish an airborne mission. The experience of the planner and the number of special considerations requested influence planning time.

E. PROPOSED AUTOMATED AIR LOAD PLANNER

This thesis develops the Ranger Air Load Planner (RAP) that generates an infiltration load plan for each aircraft. RAP presents the S3A a load plan in a graphical user interface (GUI). The GUI helps the planner visualize changes. RAP relieves the planner from tedious manual load planning and re-planning sessions. During negotiations for plan changes, the S3A will be able to quickly assess the quality of subordinate unit proposals.

F. THESIS ORGANIZATION

Chapter II explains why the author recommends a heuristic solver and not a mathematical optimization to the Rangers, although both are completely developed in this thesis.

Chapter III explains data requirements for air load planning and describes load templates in greater detail. Because unclassified template data is not available for this thesis, a template generation algorithm and an interactive display are discussed from which representative feasible templates have been generated.

An air load planning mathematical optimization model is presented in Chapter IV. This model forms a basis for evaluating feasible heuristic solutions.

The Ranger Air Load Planner (RAP) is explained in Chapter V.

Chapter VI introduces representative scenario data and evaluates the load plans from both models.

Chapter VII illustrates how the human planner can use RAP for load plan negotiation.

Conclusions and recommendations appear in Chapter VIII.

II. SOLUTION STRATEGY

A. SOLUTION OPTIONS

Two alternate solution options appear capable of solving the Ranger air load problem:

1. An integer linear program to construct an optimal solution; or
2. A heuristic that will quickly construct a good, but not necessarily optimal, solution.

The formulation, modification, and maintenance of an integer linear program requires a trained analyst who is experienced in modeling complex restrictions in terms of algebraic equations. The solution of an integer program of the size and complexity associated with the Ranger air load planning problem turns out to require the use of a commercial software package that must be purchased for about $8,000 per seat, and maintained for continuous exigent use. The operator may need experience with tuning the mixed-integer solver, and the S3A would need to learn how to express his desires with explicit objectiveness.

The initial design of a heuristic requires a skilled computer programmer with some knowledge of the efficient use of data structures for computationally intense methods. The heuristic can be written in a high-level computer language that can be later modified and maintained by any skilled programmer.

If the planner is limited to approved load templates, then the constraints on the load planning are straightforward and can be unambiguously modeled in either formulation. The rules for identifying additional templates are less clear and are subject

to change over time. It will be easier to make changes to the heuristic than to make changes to the integer programming model.

B. OBJECTIVE FUNCTION

Constructing an objective function that captures the conflicting requirements of the Rangers is quite difficult. This problem is complicated by the possible conflicting requirements of various subordinate commanders; a single load plan that satisfies all the Ranger commanders may not exist. There is not a single well defined objective function that can unambiguously identify the best solution, rather the software will be used by the planner as part of an ongoing negotiation to attempt to satisfy conflicting demands that cannot be simultaneously fully satisfied. The objective we adapt is crude: we measure whether the loads fit, carry the cargo they should, and disperse cargo. Thus, we just seek feasible loads.

C. TRANSPARENCY

A heuristic can be explained to an air load planner and thus to the other Rangers. A heuristic incorporates a mathematical objective function to evaluate each template. A greedy heuristic loads the planes one at a time by computing the value of each candidate template and choosing the best. The planner can see the algorithm work and reproduce the value of any template by hand. By contrast, the working of an integer program that selects the templates for all planes simultaneously is less easy to understand.

D. AND THE WINNER IS...

Both the optimization and the heuristic are developed in this thesis. Chapter V explains the heuristic implementation recommended to the Rangers to solve their problem. The optimization is used as a surrogate for a hypothetical *perfect load* against which the quality of the heuristic is judged.

A heuristic is preferred over a formal mathematical optimization; but testing the heuristic is essential to reassure us of robustness and effectiveness.

THIS PAGE INTENTIONALLY LEFT BLANK

III. DATA

A. DATA REQUIREMENTS

Before air load planning can begin, the S3A needs resource and constraint information to define the air load problem. The S3A gathers an aircraft resources list and a desired equipment list from a staff planning session. Provided by the Air Force, the aircraft resources list enumerates all aircraft types and tail numbers that will support the airborne operation. The Desired Equipment List (DEL) specifies what equipment, by cargo item, the unit commander desires to move in the planned operation. The Mission Essential List (MEL) is a proper subset of the desired equipment list, specifying the minimum cargo required to accomplish the operation. Any load plan that does not accommodate the MEL is infeasible. A mission essential list item that cannot be loaded is called *deficit cargo*. A Non-MEL item that cannot be loaded is called *frustrated cargo*.

B. LOAD TEMPLATES

A unit's tactical standard operating procedure manual (TACSOP) provides additional guidance. The Ranger TACSOP contains a list of US Air Force *approved load templates* (ATL) the S3A should use when planning a mission load solution. The actual Ranger ATL is classified SECRET.

A load template is a text representation of a combination of feasible cargo items that may be loaded together on a particular type of aircraft. *The load template gives no physical description of how cargo items are arranged on a particular aircraft, or in what restricted order they must load and/or unload, only that they all fit.*

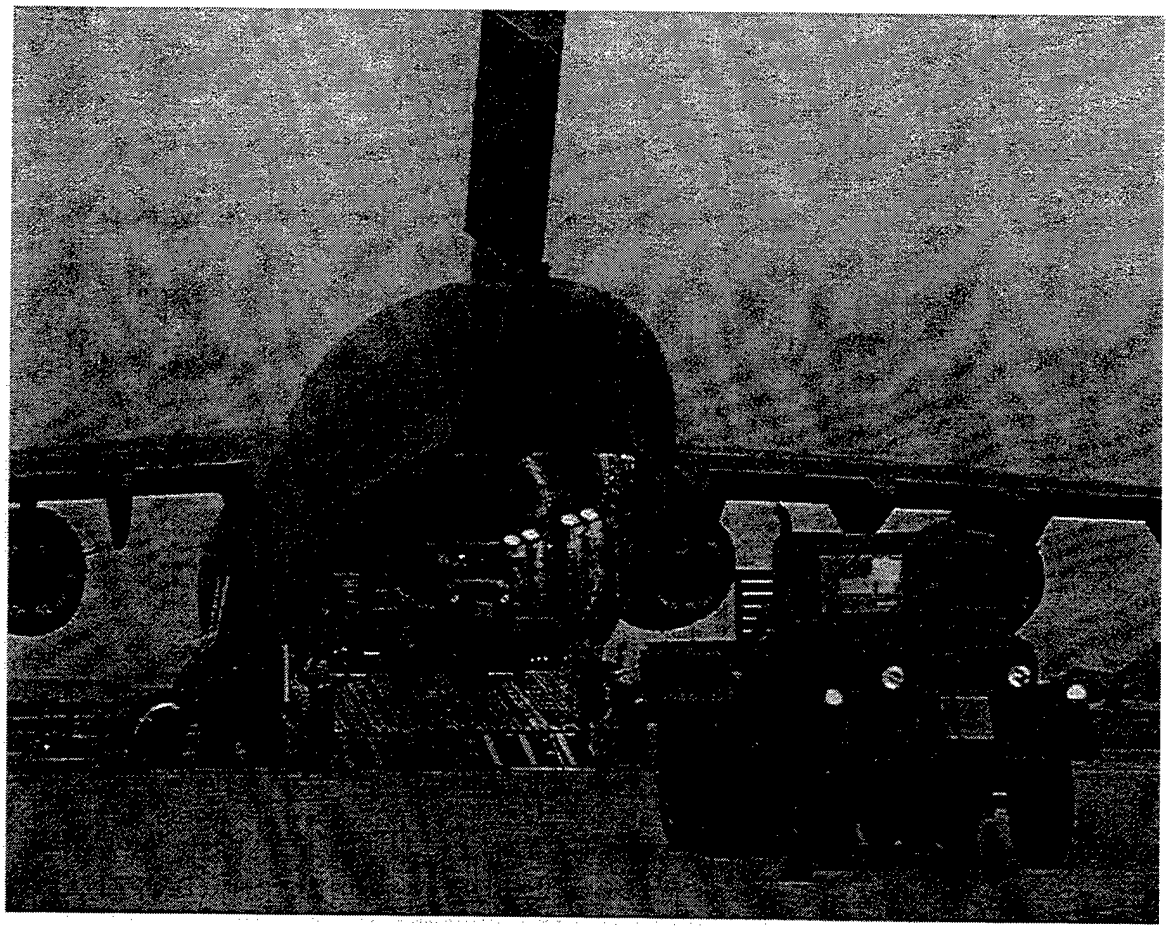

Figure 3.1. Template Approval. For each aircraft type, a proposed set of cargo items must be physically loaded by Air Force loadmasters before the set can be certified for use in an airborne operation. The Rangers record the approved combinations in the approved template list (ATL). The approval process involves loading each proposed piece of cargo on an aircraft, then arranging it so it is balanced in accordance with specific aircraft limitations. An ATL entry dictates what combinations of cargo items may be placed aboard a specific aircraft type. (Image from - [The Aviation Zone, 2000])

A template is approved only after an Air Force loadmaster has physically supervised loading the set of cargo items on a specific aircraft type and deemed the load airworthy. Before approving a cargo item set, the loadmaster adjusts the cargo items in the aircraft cargo hold until the load lies within the aircraft takeoff weight and balance tolerances. If the nominated cargo set cannot be rearranged to accommodate aircraft airworthiness restrictions, the cargo set is reduced and the loading procedure is repeated until the set can be balanced and is airworthy. This iterative process is time consuming.

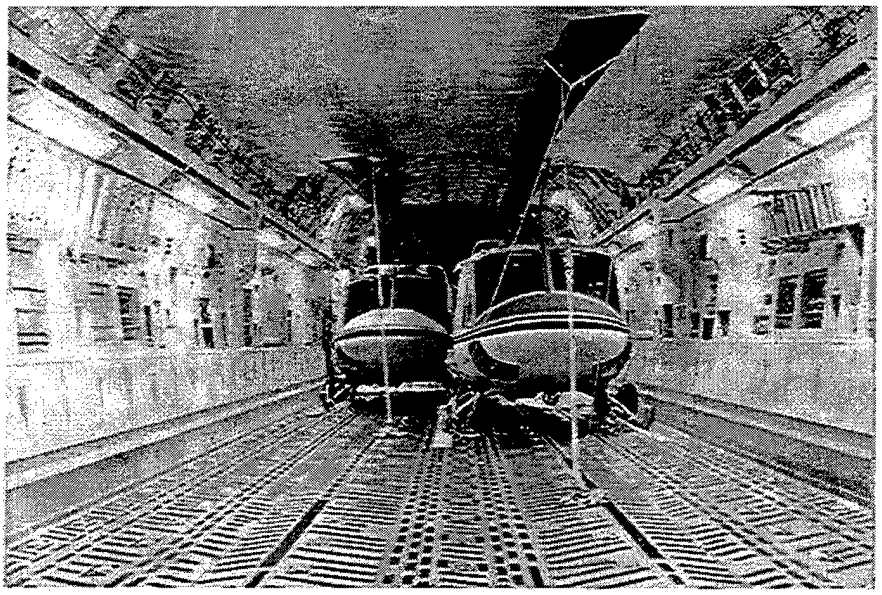

Figure 3.2. Approved Load Templates. Templates are text representations of feasible combinations of cargo items that have been approved by Air Force loadmasters. The Ranger templates do not capture the spatial orientation ---the geometry or partial order of loading and/or unloading --- of feasible cargo item sets. Templates cannot be modified with any assurance that the result is acceptable. Each new template must be attempted and approved before it can be recorded in the approved template list. (Image from - [The Aviation Zone, 2000])

If the S3A can find enough approved load templates to carry his desired equipment list, he is guaranteed the Air Force will accept his load plan.

For each aircraft type, the TACSOP has rule sets suggesting how new templates can be nominated. A new template, until it is tried and approved, is called a *non-approved load template* (NALT). A NALT must be tried and approved by a loadmaster before the S3A can be sure the cargo items it represents will be available to the airborne unit commander. NALT approval takes time, and may stall the entire planning process.

The S3A generally avoids suggesting a non-approved load template during airborne planning. Depending on a non-approved template load that turns out to be non-airworthy can lead to frustrated cargo in the plan. If a NALT is approved, it is

immediately recorded as a new template and added to ATL. This new template is available for subsequent airborne operations.

C. SYNTHETIC TEMPLATE GENERATOR

1. Overview

No unclassified source of actual Ranger load templates exists. Notional load templates have been created using first principles and open source data on the Internet.

Assuming the complex aircraft-loading problem is a one-dimensional bin-packing problem [Martins, 1999] suggests a simple procedure to generate a plausible template list. This myopic approach to template creation is in lieu of any actual classified templates.

The template synthesizer is implemented as a platform independent Java application [Cornell and Horstmann,1997]. The program reads physical dimensional information about each cargo item and each aircraft type from a text file called a property file (e.g. Figure 3.3).

The application creates aircraft specific template lists and saves them to text files. Additionally, the application writes a text file table capable of being read into other applications. The template creator uses a "greedy", linear bin-packing algorithm to create notional feasible templates. The only dimension considered here is length.

The algorithm uses upper bounds on the number of cargo items to eliminate unrealistic templates. For example, although six helicopters fit on an aircraft, templates with more than three are undesirable. The returned templates represent maximal loads, therefore dominated templates do not appear in the file.

```
[c5]
width = 228
length = 1161
weight = 240000
height = 162


[c141]
width = 123
length = 1000
weight = 40000
height = 108

[c130]
width = 123
length = 499
weight = 45000
height = 108
```

Figure 3.3. Air Properties File. The template synthesizer receives all necessary aircraft data from text files. In this example of aircraft properties, three aircraft types C5, C141, and C130 are presented. Additional information appears below each aircraft name. For example, the C141 cargo area is 123 inches wide, 1000 inches long, and 108 inches tall. The template synthesizer uses only the length of an aircraft. The aircraft allowable cabin load is 40,000 pounds. These text files can be modified to incorporate new aircraft or to modify existing ones at run time. (Data from - [The Aviation Zone, 2000]).

2. Template Creation Algorithm

The template synthesizer uses direct recursion to solve the linear packing problem. At each recursive call, the computer solves a much simpler problem, which gets reported as a template when appropriate. Figure 3.4 illustrates the pseudo-code for solving the template generation problem. At the completion of each run, the list of templates is saved to a file (Figure 3.5). The template file contains all possible dominating (i.e., maximal) combinations of cargo items for each aircraft type.

While creating realistic templates, the algorithm relies on the validity of the aircraft and cargo item data files. Every effort has been made to provide the algorithm

with comprehensive, notional aircraft and cargo dimensions that resemble actual, classified, aircraft and cargo. Additional assumptions include:

1. Cargo items are identical with respect to others of the same type;
2. Aircraft are identical with respect to others of the same type;
3. Templates selected from the generated list are equally feasible; and
4. No partial ordering of cargo items, within each ATL template, exists (i.e. no information is given on a template about the order in which cargo items must be loaded or unloaded).

The number of different cargo items in a template is called *template diversity*. If the mission spreads cargo types among aircraft by use of highly diverse templates, then if a particular aircraft fails, the diverse templates generally prevent all of one cargo item from becoming frustrated. Diverse templates are usually preferred over non-diverse templates.

```
algorithm Template Synthesizer;
begin
    for ∀i define
        LENGTHi ← cargo length inches ; # integer array of cargo item lengths #
        UPPERi ← cargo item upper bounds ; # integer array of cargo item upper bounds #
        TEMPi ← cargo item upper bounds ; # integer array of cargo items #
        PLANNEDi ← 0; # integer array of cargo item to load in a given template #
    end
    roomLeft ← 0; # scalar #
    lastItemIndex, ← index of last item in cargo item set; # scalar #
    planLoadLength ← 0; # scalar #
    lengthLoaded ← 0; # scalar - linear length of all cargo loaded #
    aircraftHoldLength ← cargo hold length of aircraft; # scalar #
    nextItem ← 0; # scalar - counter #
   addItem(0);
end
function addItem(index)
    begin
        if (index == lastItemIndex - 1)
        begin
            PLANNEDindex ← roomLeft / LENGTHindex   [integer division]
            if (PLANNEDindex > UPPERindex) → (PLANNEDindex ← UPPERindex)
            lengthLoaded ← lengthLoaded + (PLANNEDindex * LENGTHindex)
            output template
            lengthLoaded ← lengthLoaded - (PLANNEDindex * LENGTHindex)
            PLANNEDindex ← 0
        end
        else
        begin
            i ← TEMPindex
            for ( i to -1; STEP -1)
            begin
                PLANNEDindex ← i;
                planLoadLength ← i * LENGTHindex
                if (lengthLoaded + planLoadLength ≤ aircraftHoldLength )
                begin
                    lengthLoaded ← lengthLoaded + planLoadLength
                    roomLeft ← aircraftHoldLength - lengthLoaded
                    nextItem ← index + 1
                    while (nextItem ≤ lastItemIndex - 1 and LENGTHnextItem > roomLeft)
                    begin
                        nextItem ← nextItem + 1
                    end
                    if (nextItem == lastItemIndex) → (output template)
                    else( addItem(nextItem))
                    boolean bounded ← {(aircraftHoldLength - lengthLoaded)} ≥ LENGTHindex +1 * UPPERindex +1
                    lengthLoaded ← lengthLoaded - ( i * LENGTHindex )
                    if (bounded) → break
                end
            end
        end
```

Figure 3.4. Template Synthesizer Pseudo-code. In direct recursion, a function simplifies a problem and calls itself to solve the new simpler problem (e.g., Cornell, 1997). This pseudo-code outputs text representations of notional load templates to external data files.

		a	b	c	d	e	f	g	h	i	j	k
c5	.c5T6114	0	0	0	0	0	0	0	3	2	3	2
c5	.c5T6115	0	0	0	0	0	0	0	3	2	2	3
c5	.c5T6116	0	0	0	0	0	0	0	3	1	3	3
c5	.c5T6117	0	0	0	0	0	0	0	2	3	3	3
c141	.c141T6118	2	2	0	0	0	0	0	0	1	0	0
c141	.c141T6119	2	2	0	0	0	0	0	0	0	1	0
c141	.c141T6120	2	2	0	0	0	0	0	0	0	0	1
c141	.c141T6121	2	1	2	0	0	0	0	0	0	0	0

Figure 3.5. Template File. Tables provide a way to express large amounts of data in a readable form. This table provides aircraft type, template identification number, and cargo items in each template. For example, the last item, C141 template number 6,121 represents the load that contains two items "a", one "b", and two items "c". Shown is a sample from a larger table that provides an optimization model and a heuristic with data they require. This table is generated from a Java procedure.

3. Template Synthesizer User Interface

The template synthesizer graphical user interface controls notional template creation. The interface allows modification of aircraft and cargo item data at execution time. This flexibility allows notional allowable template lists to be created for many combinations of aircraft and cargo items without directly editing the master data files.

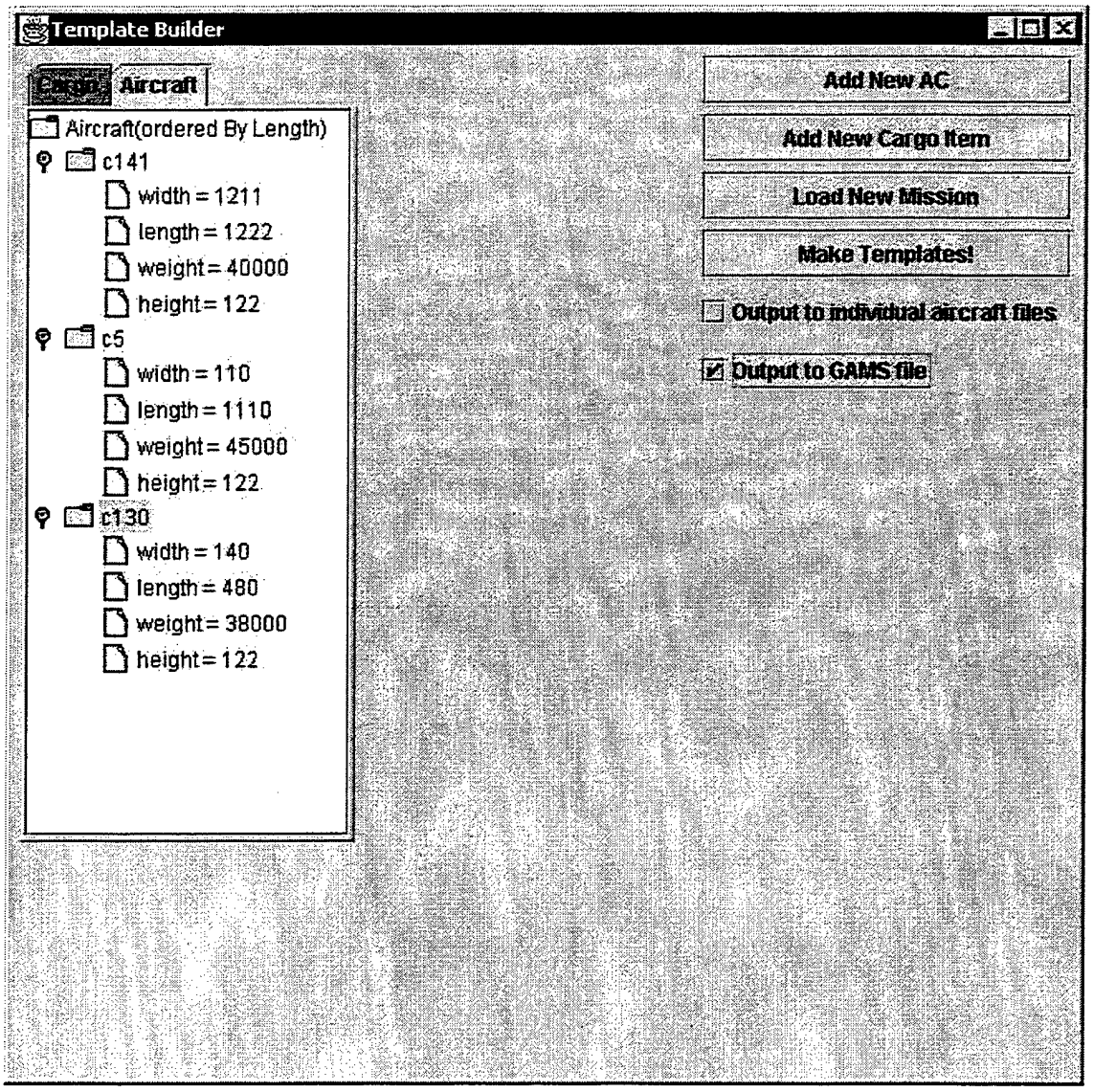

Figure 3.6. Template Interface. Highlighted on the left is the C130. The specific cargo area data is presented under its aircraft name entry. The data can be changed before a loading algorithm creates aircraft load templates. A similar display exists for cargo item physical data. (Data from - [The Aviation Zone, 2000])

THIS PAGE INTENTIONALLY LEFT BLANK

IV. OPTIMIZATION FORMULATION

A. OPTIMIZATION MODEL OVERVIEW

The optimization model is mixed-integer linear program that suggests a mission load plan by choosing the best set of load templates based on maximizing a combination of desired loading goals. The model establishes an objective basis from which to judge any other attempts solve the air load planning problem.

For a recommended load plan, the model reports for each aircraft the selected approved template list unique identification number, the aircraft identification number, a diversity measure, item capacities on this template, and an itemized load inventory. Additionally the model displays any frustrated or deficient cargo.

B. OPTIMIZATION MODEL

Index Use

a aircraft model name (e.g. C130, C141)

s serial identification number (e.g. 1, 2 ,3, 4)

i cargo item type (e.g. a, b, c, d)

l loading limitation (e.g. min, max)

t unique, non-dominated, load template (e.g. C5T1600)

r requirements and penalties (e.g. mel, penmel, del, pendel)

$fleet_{as}$ (a,s) pairs in aircraft fleet (i.e. a specific aircraft)

$loads_{at}$ (a,t) pairs of load templates t that may be used by aircraft model a (i.e. templates approved for an aircraft type)

Data

$loadcap_{ati}$	maximum number of items i for aircraft model a contained on template t
$diversity_{at}$	diversification score for each template t associated with aircraft model a
req_{ri}	value for each requirement or penalty r associated with each item i
$goal_{asli}$	goal values for every aircraft model a, serial number s, specific limitation l, and item i that exist
$goalpen_{li}$	penalty for violating goal for each specific limitation l, and item i

Decision Variables

GOODNESS	the total goodness measure of selected load plan
$LOADED_{asi}$	number of items i loaded on aircraft model a serial number s
$OVERGOAL_{asi}$	number of items i loaded on aircraft model a serial number s above the specified goal "*max*"
$SLACK_{asi}$	number of items i loaded on aircraft model a serial number s below the specified goal "*max*"
$UNDERGOAL_{asi}$	number of items i loaded on aircraft model a serial number s below the specified goal "*min*"
$FUSTRATED_i$	number of mission essential items i not loaded
$DEFICIT_i$	number of desired items i not loaded
Z_{ast}	for each aircraft model a, serial number s, select template t ($\in(0,1)$)

Formulation
(MIP1)

MAXIMIZE
Z

$$\begin{aligned} \text{GOODNESS} = & -\sum_{i} req_{\text{"penmelu"}i} * FRUSTRATED_i \\ & -\sum_{i} req_{\text{"pendel"}i} * DEFICIT_i \\ & -\sum_{fleet_{as}} \sum_{i} goalpen_{\text{"min"}i} * UNDERGOAL_{asi} \\ & -\sum_{fleet_{as}} \sum_{i} goalpen_{\text{"max"}i} * OVERGOAL_{asi} \\ & +\sum_{fleet_{as}} \sum_{loads_{at}} diversity_{at} * Z_{ast} \end{aligned} \tag{1}$$

SUBJECT TO

$$\sum_{loads_{at}} Z_{ast} \leq 1 \qquad \forall\, fleet_{a,s} \tag{2}$$

$$LOADED_{asi} \leq \sum_{loads_{at}} loadcap_{ati} * Z_{ast} \qquad \forall\, fleet_{a,s}, i \tag{3}$$

$$LOADED_{asi} \geq \sum_{loads_{at}} Z_{ast} \qquad \forall\, fleet_{a,s}, i \tag{4}$$

$$\sum_{fleet_{as}} LOADED_{asi} + FRUSTRATED_i \geq req_{\text{"}mel\text{"}i} \qquad \forall\, i \tag{5}$$

$$\sum_{fleet_{as}} LOADED_{asi} + DEFICIT_i \geq req_{\text{"}del\text{"}i} \qquad \forall\, i \tag{6}$$

$$LOADED_{asi} + SLACK_{asi} = goal_{as"\max"i}$$
$$+ \text{OVERGOAL}_{asi} - \text{UNDERGOAL}_{asi} \qquad \forall\, \text{fleet}_{a,s}, i \qquad (7)$$

$$SLACK_{asi} \le goal_{as"\max"i} - goal_{as"\min"i} \qquad \forall\, \text{fleet}_{a,s}, i \qquad (8)$$

Binary Variables

$$Z_{ast} \in \{0,1\} \qquad \forall\, a,s,t \qquad (9)$$

All other variables are nonnegative

The objective (1) evaluates the weighted average desirability of any load plan. A high penalty is inflicted for any frustrated cargo, and a lesser penalty for any deficit cargo. There may be goals on the minimum and/or maximum number of any cargo item to load on any particular aircraft; violating these goals incurs a penalty. Finally, each load template has a diversity score that rewards templates that have many different cargo items more than those that have few.

Packing constraints (2) require that each aircraft be assigned at most one load template.

Constraints (3) limit the number of cargo items loaded on each aircraft to the capacity of the selected load template for that aircraft, and constraints (4) require that if a selected template has capacity for a cargo item, then at least one unit of that item must be loaded. (Otherwise, a template might be selected for high diversity, but not used with high diversity.)

Constraints (5) and (6) respectively account for any frustrated or deficit cargo items.

Elastic constraints (7) express any goals on the minimum and/or maximum number of any cargo item that should be loaded on any particular aircraft. Together with variable bounds (8), the amount of each cargo item loaded on an aircraft determines whether this load is within, under, or over its goal. The under- or over-goal amounts determined here appear in the objective function where they are penalized.

Variable bounds (9) require that all decision variables have non-negative values, and that the decision to select a particular load template for a particular aircraft be binary (i.e., adopting fractional templates is not allowed).

THIS PAGE INTENTIONALLY LEFT BLANK

V. HEURISTIC

A. INTRODUCTION

1. Description

A heuristic is a feasible solution derived from prescriptive analysis that is not guaranteed to yield an exact optimal answer. Losses from settling for a heuristic instead of exact optimal solutions are often dwarfed by variations associated with questionable model assumptions and doubtful data.

Rardin, 1998

Heuristics are generally simple to understand and more intuitive to use than a formal mathematical optimization.

2. Heuristic Solution Technique

Some heuristics are called *greedy* heuristics. "Greedy heuristics elect the next variable to fix and its value that does least damage to feasibility and most helps the objective function, based on what has already been fixed in the current partial solution" [Rardin, 1998].

3. Ranger Air Load Planner Overview

Given a set of goals for the aircraft and a desired equipment list, the Ranger Air Load Planner (RAP) attempts to greedily solve the air load planning problem. Any cargo items not yet loaded on an aircraft but not frustrated are called *asphalt* cargo. RAP uses simple *constructive search* [Rardin, 1998]. The strategy is to load each aircraft by selecting a template for it. A single greedy pass through all the aircraft may fail to produce a feasible load plan, even if one exists. For example, exclusively loading the first planes with many similar cargo items may lead other items to be left unloaded. RAP uses a "*floating diversity penalty function*" to reduce the number of greed-induced

infeasible load plans. A template choice made early, that appears good, may ultimately necessitate poor selections later.

B. DETAILS OF THE HEURISTIC

1. Rudimentary Algorithm

RAP is very easy to understand. The algorithm can be explained to a planner who may then explain it to other Rangers involved in air load planning. Ease of understanding eliminates any ambiguity as to how a particular load plan result is achieved. The algorithm relies on an objective function that can compute a *loaded value* for each template. The RAP algorithm is:

For *the smallest unloaded aircraft, given cargo still to load -*
  Evaluate *objective function for each template*
  Choose maximum *objective function value and associated template*
  Subtract *the number of each cargo item in the selected template from cargo still to load*
Do for all *remaining aircraft*

2. Construction of Objective Function

a. *Get Base Value*

A cargo item value is the value of that item to the mission. Cargo item values are normalized to fractions of the most valuable item type and each cargo item gets a new *relative equipment value*. Item base value is the number of specific items loaded multiplied by the relative equipment value. The sum of all base values is called the *template base value*.

b. ***Determine Violation Penalties --- following manual planner guidance***

(1) Upper and Lower Goals. An upper goal is the most number of a specific cargo item the planner wants to see loaded on a specific aircraft. A lower goal is the minimum number of a specific cargo item a planner wants to see loaded on a specific aircraft. RAP assesses a penalty if a candidate template violates a defined upper or lower goal. The amount of penalty depends on the magnitude of the deviation above an upper goal or below a lower goal. Loaded amounts that fall between an upper goal and a lower goal are not penalized.

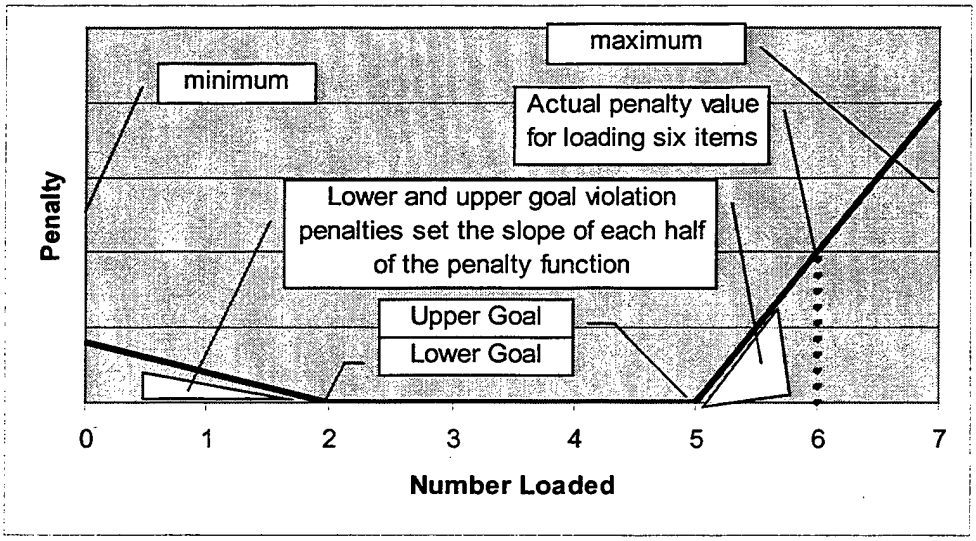

Figure 5.1. Visualizing Planner Guidance. The figure above shows the goal violation penalty function when a planner sets an upper and lower goal for a particular cargo item on a specific aircraft. For this cargo item on this aircraft the minimum load is zero and the maximum load is seven, while the lower goal is two and the upper goal is five. Changing violation penalties changes the slope of the goal violation function tails. By setting a upper violation penalty of 10 and a goal of five items, RAP will pay a large penalty for trying to load six cargo items. Loading one item accrues less penalty because the lower goal violation penalty is only two.

(2) Locked Goals. During planning, it may be desireable to absolutely fix the number of specific cargo items on an aircraft. When the planner does not want a goal violated he *locks* the goal. Upper and lower goals may be locked at different numbers, indicating an absolute range of accepted values or at the same number indicating it is the only acceptable number a feasible solution should return for a specific aircraft load plan. Locking a goal implies an infinite penalty for violating it. However, this is implemented in RAP as a large penalty. Thus RAP may violate even locked goals for aircraft when feasibility is threatened.

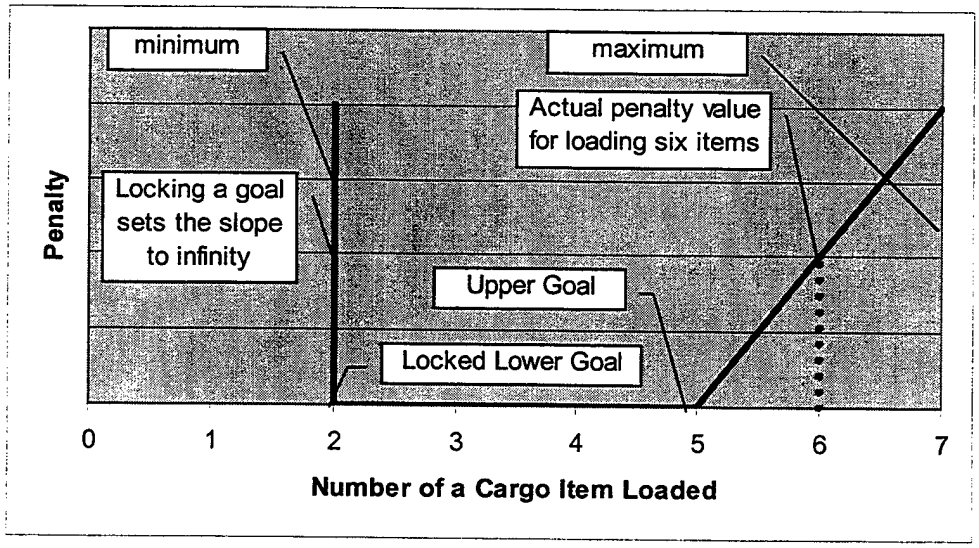

Figure 5.2. Locking One Goal. The figure above is another representation of the goal violation penalty function. The planner has established an upper goal of five and a lower goal of two. To indicate he does not want to see any solution with fewer than two of this cargo item on this aircraft, he *locks* the lower value. Violating a locked goal induces a conceptually infinite penalty.

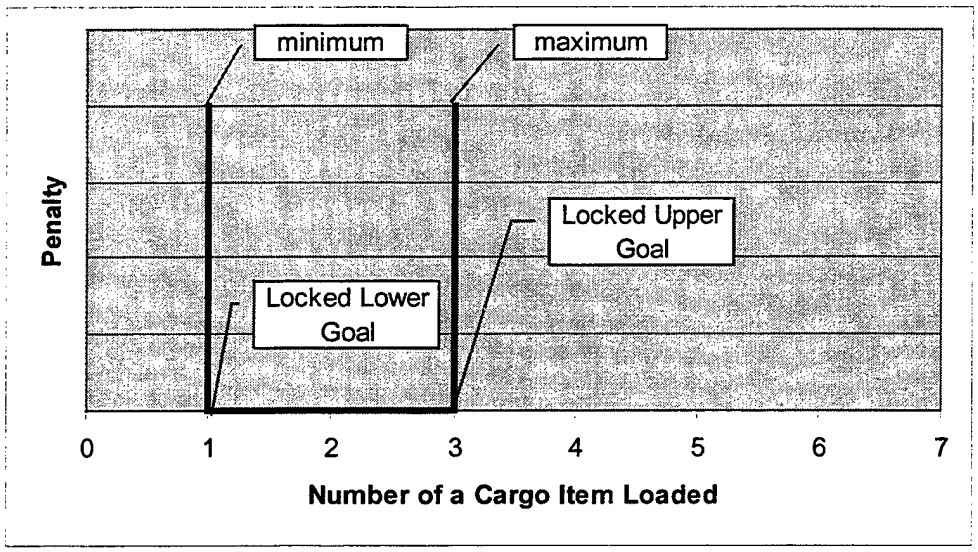

Figure 5.3. Locking both Goals. During load plan negotiation, on a given aircraft for a given cargo item type, the planner may set the absolute range of the acceptable number of items by locking an upper and lower goal. In this figure, the planner will only accept solutions if the number of this cargo item is between one and three. If the planner has a particular number in mind, he can set both the upper and lower value to that number. Moderation is a virtue: Locked goals restrict the approved load templates that can be used, and may even rule out all approved templates.

(3) Diversity Value. *Mission diversity* is the degree to which cargo item numbers are evenly distributed across all mission aircraft. If even loading is possible, each aircraft should take a specific number of cargo items of a specific type and taking any more would reduce mission diversity. For example, if there are three identical aircraft and three identical cargo items, the highest mission diversity would exist when each aircraft loads one cargo item. Loading any more than one on any aircraft would reduce mission diversity. Aircraft should take numbers of cargo items commensurate with their cargo hauling capacity and the number of cargo items available to load. Aircraft that load in this manner are said to load their "fair-share" of cargo items. Mission diversity is desirable.

RAP penalizes aircraft that attempt to take too many or too few of a single item with a penalty based on the deviation from the aircraft's "fair-share" of the

remaining items raised to a user-determined power. This thwarts the natural tendency for the aircraft that are loaded first to select all the high value items and thus reduce diversity. This penalty is relaxed when fewer aircraft are available to load. Incrementally relaxing this penalty when few planes remain ensures that loading all cargo items takes precedence over attempting to load all cargo items evenly.

The *mission diversity value* is the relative weight applied to total mission diversity when RAP calculates the potential contribution to the objective function of a candidate template. The planner uses a high mission diversity value when he wants the returned load plan to use templates that lead to mission diversity.

3. Solution Sequence

RAP requires a list of feasible templates, loading goals for each aircraft, goal violation penalties, relative equipment values, and mission diversity value before it constructs an air load plan.

For each aircraft in the mission, RAP selects a template that seeks to maximize diversity and loaded equipment value, while meeting predetermined goals. RAP loads aircraft in order from smallest to largest. Once an aircraft load template is selected, the equipment set in that template is subtracted from the remaining cargo items to load and is not available to any other aircraft in the mission. A mission load plan is feasible when the sum of all selected template cargo item numbers equals the desired equipment list.

RAP solves each aircraft air load problem in five phases: Setup, Base-value Determination, Penalty Assessment, Load Plan Adjustment and Template Evaluation.

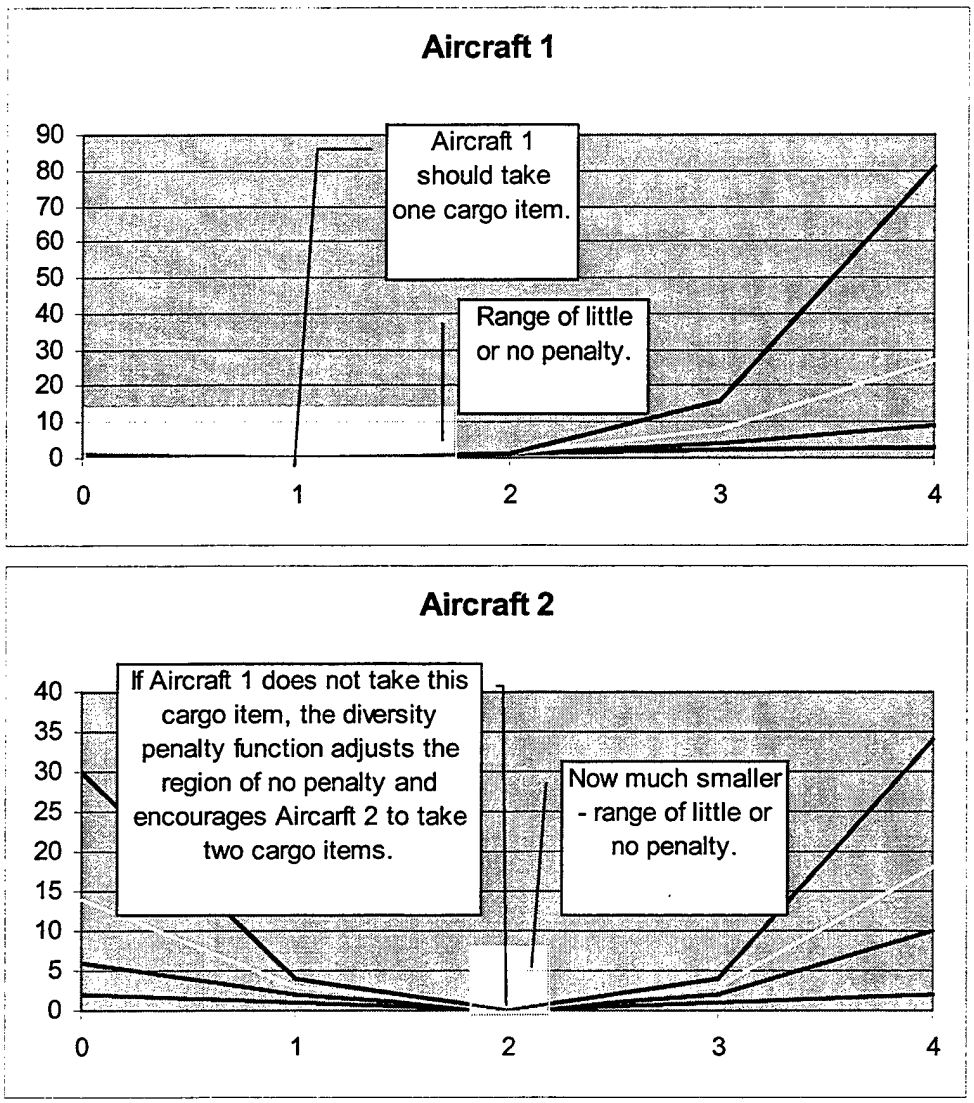

Figure 5.4. Moving Diversity Penalty. The diversity penalty function (DPF), graphically displayed above at four levels, assesses a penalty for loading too many of a given cargo item on a given aircraft. The DPF calculates how many items an aircraft should take based roughly on how many the remaining aircraft not yet loaded should take. In the example above, there are four identical cargo items that must be loaded on any of four identical aircraft. Each aircraft should get one item to earn each aircraft one diversity point each. If Aircraft 1 does not load any cargo items, Aircraft 2 is motivated to take two items because taking any less would hurt his objective score.

C. THE HEURISTIC PRESENTED AS AN ABSTRACT ALGORITHM

```
algorithm RAP (array of aircraft to load, asphalt, all templates);
begin
    define
        begin
        asphalt ← array of integers that is the numbers of each cargo item in the
                  mission
        numberMEL ← array of integers that is the numbers of mission essential
                    cargo items in the mission
        templates ← array that is every approved template for an aircraft type
        UVPEN ← upper goal violation penalty
        LKGVPEN ← locked goal violation penalty
        LVPEN← lower goal violation penalty
        DIVERPEN ← diversity penalty
        DIVERPOW ← exponent reflecting diversity importance {eg: 1... 4}
        k ← maximum number of cargo items any template will ever have
        j ← number of unique cargo item types in mission (column length of asphalt
            array)
        MELBONUS ← scalar, added to objective function for loading mission essential
                   cargo items
        UPPERGOAL ← two-dimensional array of integers that is the planner
                    determined maximum number of items an aircraft should load.
                    The dimension of this array is [number of aircraft in the
                    mission, j].
        UNDERGOAL ← two-dimensional array of integers that is the planner
                    determined minimum number of items an aircraft should load.
                    The dimension of this array is the same as UPPERGOAL.
        values ← three dimensional array that holds scoring values
        itemValues ← scalar array that is the total military worth of each cargo item
        end
    loop (a ∈ aircraft to load)
        setup( a, asphalt)
        base-value (a, asphalt)
        loadPlanAdjust (a)
        evaluation (a, asphalt, templates_a)
    end
end

function setup( a, asphalt )                                        (1)
begin
    numberOfUniqueCargoItems ← j
    dim values_a (k, numberOfUniqueCargoItems)                      (2)
    dim tenPlan_a (k, numberOfUniqueCargoItems)                     (3)
```

```
end

function base-value ( a, asphalt )                                    (4)
begin
    for ( ∀l ∈ 0 to k-1 )
    begin
        for ( ∀i ∈ cargoitems )
        begin
            tenPlan_ail ← min { l, asphalt_i }                        (5)
            values_ail ← tenPlan_ail * itemValue_i
            penaltyAssessment ( a, l, i, asphalt_i )                  (6)
        end
    end
end

function penaltyAssessment (a, l, i, asphalt_i)                       (7)
begin
    # upper goal calculations #
    overpen_ali ← 0                                                   (8)
    if (Is this cargo item's upper limit locked ?)
      if ( tenplan_ali - uppergoal_ai) > 0
        overpen_ali ← | tenplan_ali - uppergoal_ai | * UVPEN * LKGVPEN   (8a)
    else        # upper goal not locked #
      if ( tenplan_ali - uppergoal_ai) > 0
        overpen_ali ← | tenplan_ali - uppergoal_ai | * UVPEN          (8b)

    # lower goal calculations #
    underpen_ali ← 0                                                  (9)
    if (Is this cargo item's lower limit locked ?)
      if ( tenplan_ali - lowergoal_ai) < 0
        underpen_ali ← | tenplan_ali - lowergoal_ai | * LVPEN * LKGVPEN   (9a)
    else        # lower goal not locked #
      if ( tenplan_ali - lowergoal_ai) < 0
        underpen_ali ← | tenplan_ali - lowergoal_ai | * LVPEN         (9b)

    # diversity calculations #
```

$$\text{diversityPen}_{li} \leftarrow DIVERPEN * \left| \left(\text{asphalt}_i * \frac{\text{len}_a}{\sum_{i=a}^{n} len_i} \right) - \text{tenplan}_{ali} \right|^{DIVERPOW} \qquad (10)$$

```
    # bonus calculation #
    loadingBonus ← 0
    loadingBonus ← ( min (numberMEL_i , tenPlan_ail ))* MELBONUS          (11)
    # values table adjustment #
    values_ail ← values_ail + loadingBonus
                        - ( overpen_ali + underpen_ali + diversityPen_li )   (12)
end

function loadPlanAdjust (a)                                               (13)
begin
    BEST ← - ∞
    BESTIDX ← -1
    for (∀i ∈ cargoitems)
    begin
        for (∀l ∈ l=0, k-1)
        begin
            if (values_ail > BEST)                                        (14)
              BEST ← values_ail
              BESTIDX ← l
        end
        x ← BESTIDX
        tenPlan_ail ← tenPlan_aix                                         (15)
    end
end

function evaluation (a, asphalt, templates_a)                             (16)
begin
    BEST ← - ∞
    BESTT ← -1
    for (∀t ∈ templates_a)
      SCORE ← 0
      begin
      for (∀i ∈ cargoitems)
          begin
            y ← t_i
            SCORE = SCORE + values_aiy                                    (17)
        end
        if (SCORE > BEST)                                                 (18)
           BEST ← SCORE
           BESTT ← t
    end
    for (∀i ∈ BESTT)
       b ← BESTT_i
       begin
```

$asphalt_i = asphalt_i - tenPlan_{abi}$ (19)
$numberMEL_i = numberMEL_i - tenPlan_{abi}$ (20)
end
end

Function (1) accepts as input the specific aircraft to be loaded and an *asphalt* array that represents the number of each cargo item, *i*, currently not loaded. Statements (2) and (3) define two, two-dimensional arrays that are *k* rows, by *j* (the number of unique cargo item types long). Each row represents a cargo loading capacity limit on any given template. For example, row $l = 0$ indicates there is no capacity for a cargo item. The defined maximum amount of any cargo a template can have is *k*-1 items. Function (4) receives the aircraft and the asphalt and is responsible for populating both two-dimensional arrays.

Assignments (5) populate the two-dimensional array, tentative plan, with the minimum of *l* and what is currently on the ground, $asphalt_i$. To gain computational speed the Penalty Assessment Function (7) is called (6) from the BaseValue Function (4).

Assignments (8), (8a), and (8b) use absolute differences to penalize loading more than a planner's guidance. When the planned load does not exceed an upper goal, assignment (8) sets the upper violation penalty to zero. Similarly conditional assignments (9), (9a), and (9b) handle cases where the planned load falls below a planner's guidance.

Assignment (10) calculates the diversity penalty. Based on the amount total cargo loading space still available, assignment (10) first computes the expected number of item *i* aircraft *a* should load if item *i* is distributed proportionally among remaining aircraft. Assignment (10) next raises the absolute difference between the amount that should be

loaded and the planned load to an exponent. The changing diversity exponent allows the planner to induce more or less diversity in early stages of mission planning.

Assignment (11) calculates a reward that is added to the objective function when a tentative plan loads mission essential cargo.

Assignment (12) combines the value, any mission essential loading bonus, and all penalties for loading l cargo items of cargo item i. This is the objective function value.

Function (13) modifies the tentative load plan array to reflect the best value based loading policy. This array is used to compare template capacity, the row, and what should be loaded given that capacity. For example, if there are 2 items on the ground and a template has a capacity for 10, the policy should be to load 2 or fewer. Traversing down every column, i, of the values array, conditional statement (14) screens for the highest value. If candidate $value_{li}$ exceeds the best incumbent value thus far, this improved incumbent value and its row index, *BestIDX*, are recorded. Assignment (15) ensures that any template selected loads the best amount it can, which is not necessarily the maximum capacity.

Function (16) evaluates all maximally-loaded feasible templates for aircraft a and selects the best one. The template score is calculated by looking in the ith position of each template for a loading amount that corresponds to a row l, the load potential, in the value table. For each template, all identified values are summed (17) and conditional statement (18) screens for the best template value. Once the best maximally-loaded template has been identified, the heuristic enters each capacity in the tentative plan array as a row index and receives the actual amount to load. Finally the amount to load, from the tentative plan array (not from the template), is deducted from the asphalt cargo (19).

Assignment (20) recalculates the numbers of mission essential cargo that remain unloaded.

D. GRAPHICAL USER INTERFACE

1. Template Data Input

The template data file contains load templates for all cargo items and aircraft used in the mission; the data is blank-delimited. The template data file contains all maximally-loaded candidate templates the heuristic will try when loading aircraft.

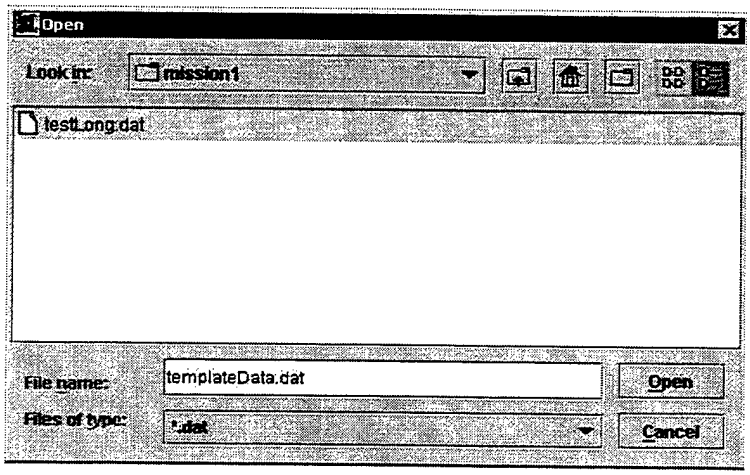

Figure 5.5. Template Data Dialog. The template data file contains all the feasible candidate load templates RAP considers for load planning a given mission.

2. Aircraft and Cargo Item Input

Aircraft and cargo item physical attributes are defined in an external property file and are loaded at run time. The aircraft and cargo item names must correspond to template data entries exactly. The planner has the option of loading saved aircraft and cargo item files or creating them new for this planning session. Like the template data, aircraft and cargo item property files may be discovered at run time on any computer storage device the planner has access to. Figure 5.6 shows the graphical user interface display for this.

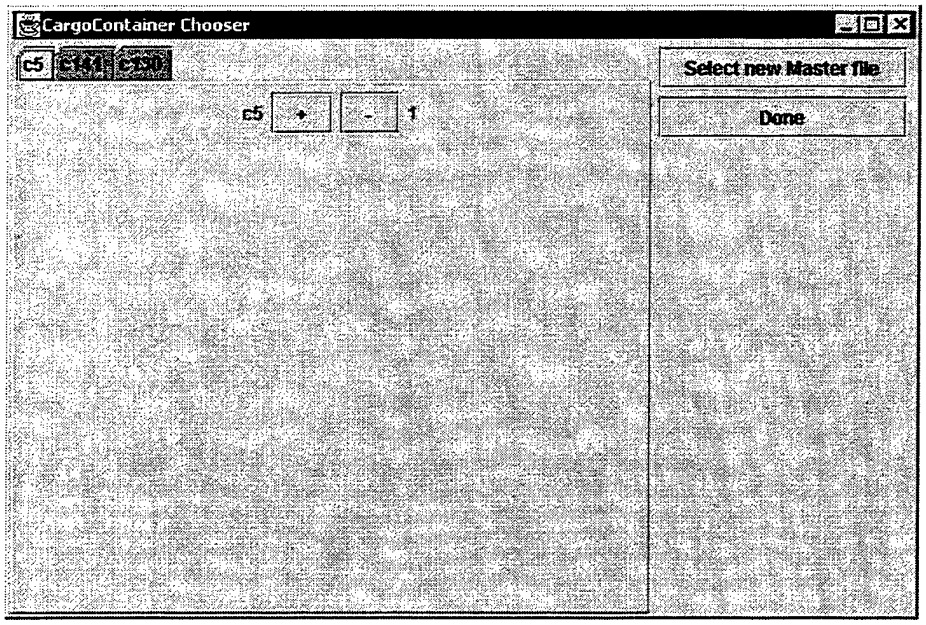

Figure 5.6. Cargo Aircraft Selector. Here, the planner selects the number of aircraft he plans to load. In this display, he plans to use one C5. He may add additional aircraft types by pressing the *Select new Master File* button in the upper right of this display or increase the number of aircraft by pressing the plus button. RAP discovers cargo items using a similar looking interface.

3. Aircraft Loading Guidance

Once the template data, aircraft, and cargo items have been discovered, the planner is prompted to suggest loading goals for each aircraft and each cargo item. If absolute adherence to a goal is sought, a planner may *lock* upper and/or lower goals. Locked goals appear yellow. It is possible make the overall load plan infeasible by locking goals. Blank lower or upper goals indicate planner indifference to lower or upper loading values. Figure 5.7 illustrates an example.

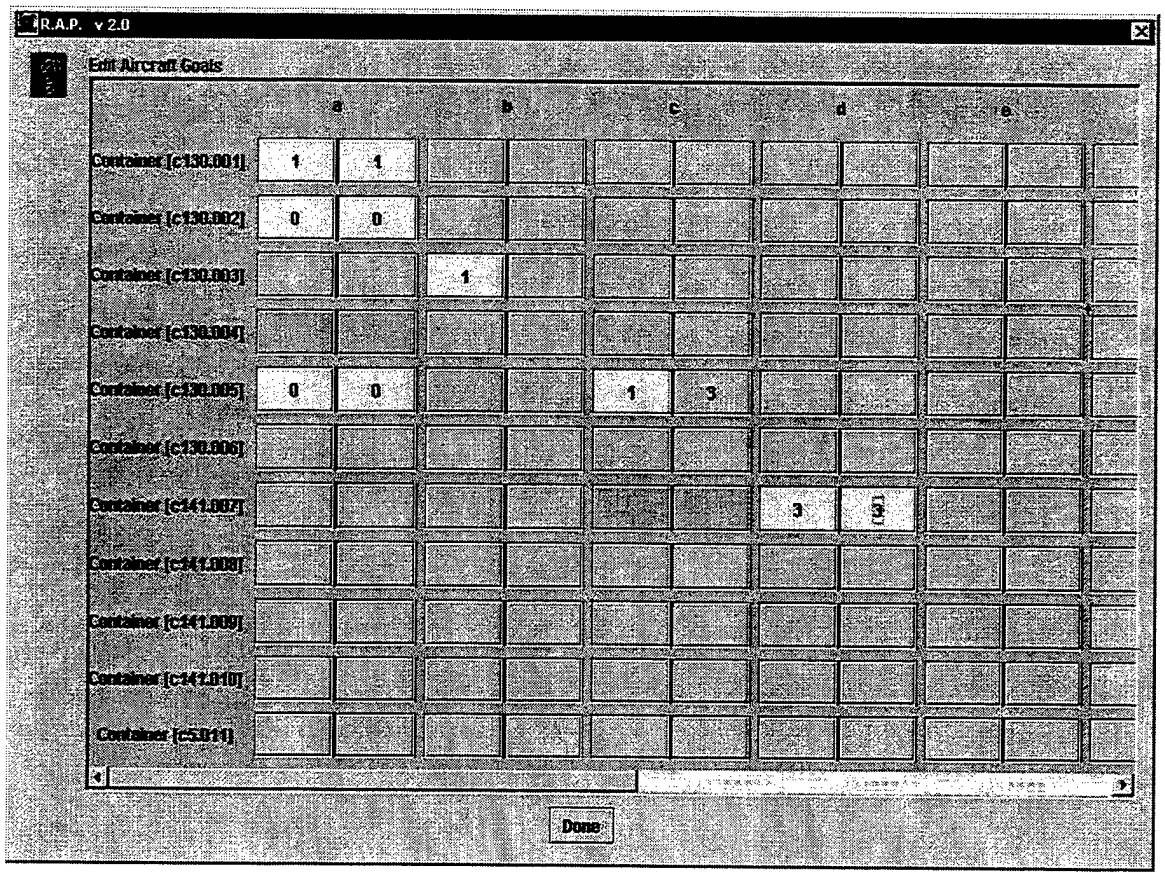

Figure 5.7. Aircraft Loading Guidance. An air load planner may suggest upper and/or lower loading goals for specific items on specific aircraft. Additionally, he may be intolerant to deviation from his goals - *locking* them - because of specific tactical concerns. For example, exactly one item *a* is locked on C130.001 and exactly zero item *a* on C130.002. Aircraft c130.005 has a lock of zero on item *a*, locked lower goal of one and an upper goal of three on item *c*. The planner wants exactly three *d* items on aircraft C141.007. Blank entries indicate any feasible numbers of particular cargo items are acceptable.

4. Penalties and Diversity

The planner can define violation penalties differently for lower and upper goals. He defines the goal violation penalty slope by using a "slider" bar.

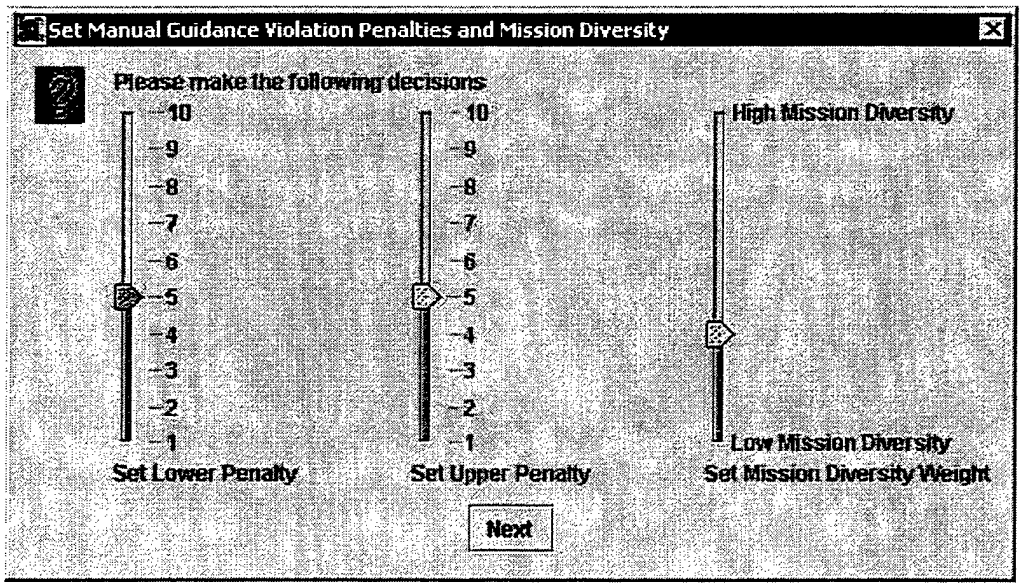

Figure 5.8. Relative Importance of Goals and Diversity. In the interest of feasibility, RAP may violate a user goal. The two sliders on the left indicate to what degree RAP should adhere to lower and upper goals. Displayed above, lower goal violation costs five units and upper goal violation cost eight. The diversity slider indicates a reasonably low diversity (exponent amplifying diversity in objective).

5. Relative Equipment Values

Each cargo item in the mission has a relative value. The planner selects a value for the each cargo item with a slider. RAP automatically normalizes equipment values by dividing each value by the lowest value and using these normalized scores in subsequent calculations. Figure 5.8 shows the full display of this.

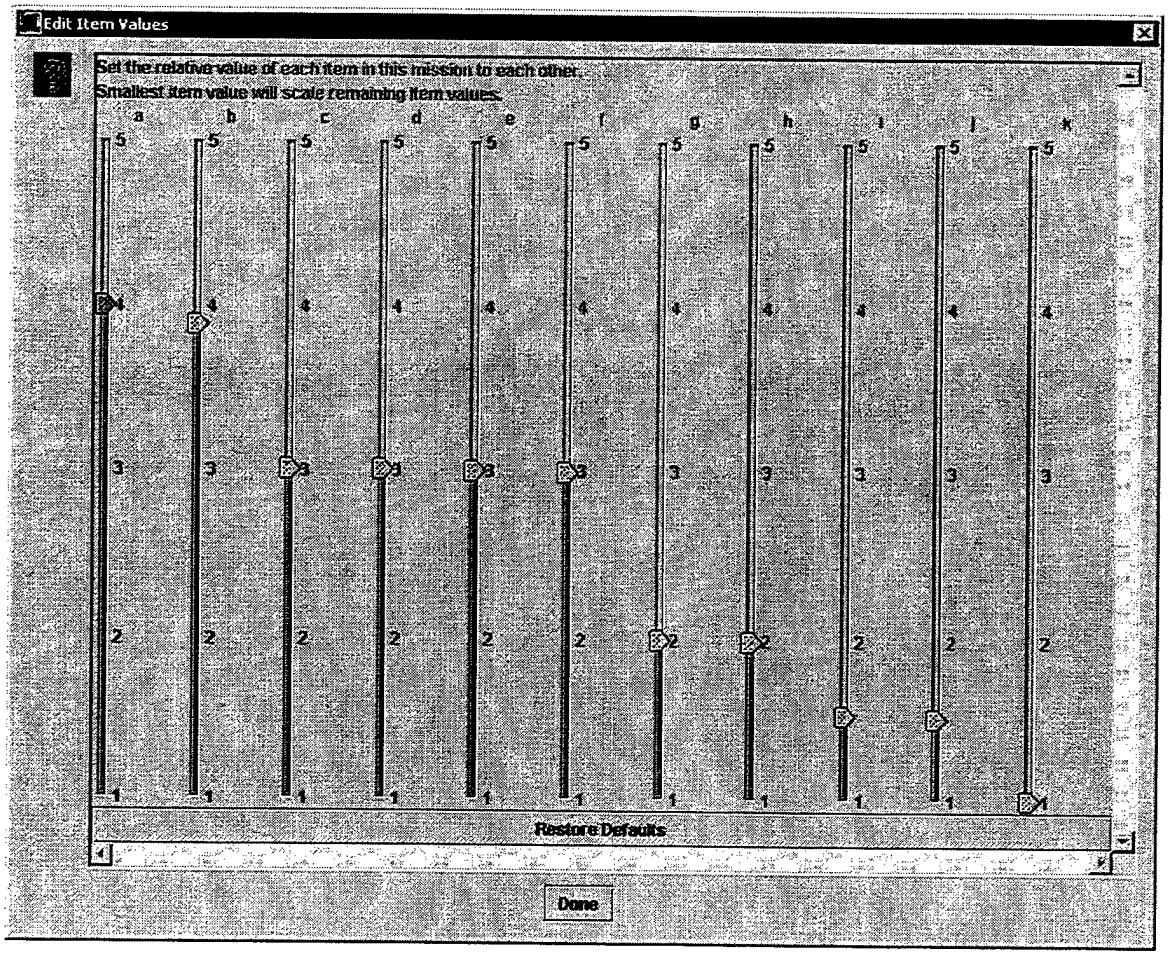

Figure 5.8. Relative Values. Generally mission-dependent, each cargo item has a relative mission value that expresses its worth. In the figure above item *a* is valued at four times item *k*. Using this display, the planner can easily adjust the value of cargo items in units of a base value, here, *k*.

VI. IMPLEMENTATION AND ANALYSIS

A. SAMPLE PROBLEM

The number and types of cargo items and aircraft used here suggest a battalion-sized operation. Because no Ranger load plan is available for this fictitious sample mission, the optimization is used as a benchmark to assess the quality of the heuristic solution. The optimized load plan is assumed to be what the planner would have chosen if he had solved the air load planning problem manually.

Using the template synthesizer described in Chapter III, 16,048 notional templates for 11 types of cargo and three types of aircraft are generated. Based on conversations with various Rangers involved in air load planning, the author is convinced this number of templates greatly exceeds the number maintained by the Rangers. The sample problem has 51 cargo items of 11 types to load on 11 mission aircraft.

B. DATA

1. Aircraft

The number of each aircraft type presented to both models is shown in Table 6.1. The selection of aircraft types and numbers are arbitrary, but suggest a battalion-level exercise.

Aircraft Type	Hold Length (in)	Hold Width (in)	Hold Height (in)	Allow Cabin Load (lbs)	Total Number in Mission
C5	1,161	228	162	263,000	1
C141	1,000	123	108	94,000	4
C130	499	123	108	42,000	6

Table 6.1. Aircraft Resource List. This table indicates how many aircraft, of specific types, need load plans. Using fewer than the indicated aircraft is undesirable because given a feasible solution, using fewer aircraft decreases diversity. For example, there is a feasible load plan to place all the cargo on the six C130 and two C141 aircraft, but this would not meet the mission diversity goal described in Chapter V. (Data from - [USAF, 1999])

2. Template Data

Table 6.2 displays the aircraft type specific number of templates considered. (This template list is artificially constructed by the template generator, and is not the classified list of templates used by the Rangers.)

Aircraft Type	Templates Considered
C5	6,117
C141	9,779
C130	152

Table 6.2. Number of Indicated Load Templates for each Aircraft Type. Shown is the number of templates considered for each aircraft type.

3. Cargo Items

The numbers of cargo items are arbitrary, and the names are suppressed here. Lengths have been obtained from open Internet sources. Equipment dimensions, specifically length, are relevant to synthetic template generation only. If a cargo item appears on a approved template, by definition, it fits on an aircraft.

Item Name	Length (in)	Width (in)	Height (in)	Weight (lbs)	Maximum In Any Template	Total Number MEL in Mission	Total Number DEL in Mission
a	276	65	98	3,378	2	0	2
b	265	75	74	8,166	4	0	4
c	200	75	74	7,040	4	0	3
d	199	86	102	8,000	4	0	2
e	198	75	74	7,050	3	0	12
f	197	86	72	7,100	3	0	4
g	169	65	62	5,200	3	0	2
h	163	63	57	2,816	3	0	2
i	84	36	48	240	3	0	12
j	82	47	50	690	3	0	12
k	72	29	42	160	3	0	6

Table 6.3. Cargo Items List. The numbers of cargo items to be loaded in the sample mission are in the last column. Physical dimension information is provided for reference and notional template generation only. The reader is reminded MEL is the mission essential list and DEL is the desired equipment list. (Data from - [Internet, 1999])

C. OPTIMIZATION IMPLEMENTATION

The optimization model is implemented in the General Algebraic Modeling System (GAMS) [Brooke et al., 1997] with the CPLEX solver, Version 6.5 [ILOG, 1999]. The model uses a branch and bound technique to arrive at a feasible solution to

the sample problem. The integer program has 277 equations and 46,294 variables. With a relative integer solution tolerance of 10 percent, it runs in less than three minutes on a personal computer with a Pentium II 266MHZ processor and 128 MB of random access memory. The sample problem solution exhibits a zero absolute integer gap (the difference between the solution objective value and an upper bound on this), indicating a globally optimal solution.

The optimization model mimics an air load planner who will accept any feasible solution that is the most diverse. All cargo item goals are relaxed, allowing diversity to dominate the objective function value. No cargo item values are used because we simply seek a feasible load plan.

D. RANGER AIR LOAD PLANNER IMPLEMENTATION

RAP is implemented in Java [Sun Microsystems, 1998], Version 1.3 Beta. On a Pentium II 266MHZ processor personal computer with 128 MB of random access memory, RAP runs in less than one second.

For the base case run, all RAP parameters are at default settings.

E. COMPUTATIONAL RESULTS

1. Comparison of the Optimization and Ranger Air Load Planner

The optimization returns an optimal solution with an objective function score of 49 and reports no frustrated cargo items. RAP also returns a feasible solution with an objective function score of 49 and reports loading all cargo items.

When the selected template identification numbers are compared, the optimization and heuristic differ in all but three of 11 aircraft. Given non-dominated templates in the template list and no loading goals, there are probably many alternate template selections

that constitute virtually indistinguishable load plans. Any template assigned to a specific aircraft can be freely exchanged with any other template assigned to another aircraft of the same type, with no loss of objective function value or threat of feasibility violation.

Optimization

	a	*b*	*c*	*d*	*e*	*f*	*g*	*h*	*i*	*j*	*k*
c5_001	1		1	1	1				1	1	1
c141_001		1	1		1	1			1		1
c141_002		1		1	1	1			1		1
c141_003	1	1	1		1		1		1		
c141_004		1			1		1	1	1		1
c130_001					1	1			1		
c130_002					2				1		
c130_003					1				1	1	1
c130_004					1	1			1		
c130_005					1			1	1		
c130_006					1				2		1
	2	4	3	2	12	4	2	2	12	2	6

RAP

	a	*b*	*c*	*d*	*e*	*f*	*g*	*h*	*i*	*j*	*k*
c5_001					1				1		1
c141_001	1	1			1		1	1	1		1
c141_002		1	1	1	1	1			1		1
c141_003		1			1	1	1	1	2		1
c141_004	1	1		1	2				1		
c130_001					1	1			1		
c130_002					1				1	1	1
c130_003			1		1				1		
c130_004					1	1			1		
c130_005					1				1	1	1
c130_006			1		1				1		
	2	4	3	2	12	4	2	2	12	2	6

Table 6.4. Load Plans. Manually constructed load plans are the product of many hours of planning and negotiation. Using computers, it is possible to arrive at quality feasible solutions in minutes rather than hours. Shown above are two summary tables, which are the load plans for a sample air load planning mission done on a personal computer. Aircraft names are listed on the left of both tables, while the cargo item names appear at the top of each column. The sum of specific cargo items loaded is the last row entry of each table. Both implementations load the same numbers of cargo items.

	a	*b*	*c*	*d*	*e*	*f*	*g*	*h*	*I*	*j*	*k*
c5_001	1		1	1						1	
c141_001	-1		1			1	-1	-1			
c141_002			-1								
c141_003	1		1			-1		-1	-1		-1
c141_004	-1			-1	-1		1	1			1
c130_001											
c130_002					1					-1	-1
c130_003			-1							1	1
c130_004											
c130_005								1		-1	-1
c130_006			-1						1		1
	0	0	0	0	0	0	0	0	0	0	0

Table 6.5. Load Plan Difference. Both the optimization and the heuristic returned feasible load plans that have the same objective function penalty value, but they do not use the exact same templates. Table entries above show discrepancies between the optimization and the heuristic load plans. A positive one for cargo item *a* on both c5_001 and c141_003 means the optimization suggests the planner load one more of these than the heuristic does. Negative quantities reflect the opposite case.

2. Runtime Complexity of the Optimization and Ranger Air Load Planner

RAP runtime is ***O*** (*number of templates * number of cargo item types * number of aircraft*). Computing time can be predicted with accuracy. On a Pentium II personal computer 266MHZ computer with 128MB of RAM, RAP returns the sample problem solution in 0.111 seconds. It is particularly important that runtime grows linear with *t* because this means that the heuristic will continue to operate efficiently as more templates are added.

Worst-case runtime for the optimization model is exponential, and on any computer envisioned would take longer than the estimated survival of the sun. In practice, runtime is considerably better, on the order of a minute or so on a workstation. Empirically, runtime grows linearly with the number of templates. Storage complexity of the optimization is comparable to that of RAP.

VII. WHEN THINGS GO WRONG

A. HEURISTICS CAN BE RISKY

Because RAP is a myopic, greedy, and non-backtracking heuristic, it will fail for some air loading problems, even when a feasible plan exists and is discovered by the optimization model. Haste has a price.

One example of first-pass RAP infeasibility arises when using a small number of approved templates. Few template combinations in the feasible solution set means myopic early template selections may have adverse consequences. As RAP loads aircraft from smallest to largest, it inadvertently selects poor template combinations that makes the initial returned solution infeasible.

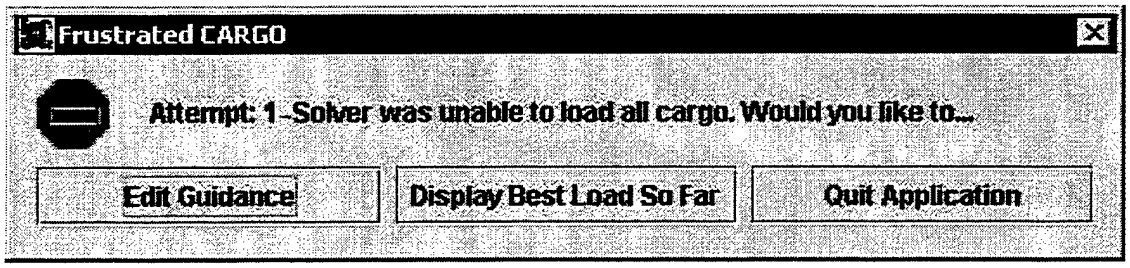

Figure 7.1. Frustrated Cargo. RAP may not find a feasible solution to an air load problem in a single pass and therefore leave cargo on the asphalt. This dialog box displays options the planner has at the end of a RAP run if frustrated and/or deficit cargo items are encountered.

A planner can help RAP find a feasible solution. The planner can adjust model parameters, "unload" some aircraft, and then give RAP another opportunity to solve the new air load problem. This "change parameters-unload aircraft -resolve" process also suggests a constructive way to conduct negotiations with subordinate unit commanders.

The sample problem in this Chapter is designed to induce an infeasible initial RAP solution and illustrate how to discover a feasible solution by negotiation.

B. RESTRICTED SAMPLE PROBLEM

The aircraft and cargo item data remain unchanged from before. A uniformly distributed random sample is made from the approved template list for each aircraft type. The new approved template list contains only 180 total templates.

Aircraft Type	Old Number of Templates	New Number of Templates
C5	6,117	50
C141	9,779	100
C130	152	30

Figure 7.1. Smaller Approved Template List. Small approved template lists hinder RAP by reducing the number of feasible template combinations possible to achieve a feasible load plan. Shown is the number of randomly chosen templates used in another run of both the optimization and RAP. The reduction in number of templates leads the myopic RAP to return an initial infeasible solution, which is repaired through planner guidance.

C. COMPUTATIONAL RESULTS

1. The Optimization Works

The optimization model has 265 equations and 46,278 variables, and solves in a minute. The sample problem exhibits a ten-percent integer gap, but the optimization model constructs a feasible solution.

2. But RAP Fails

RAP reports an infeasible solution in 0.070 seconds. The heuristic initial solution leaves one item *d* and one item *e* on the asphalt and loads all other mission equipment.

R.A.P. v 2.0 LOAD PLAN

Aircraft Name	a	b	c	d	e	f	g	h	i	j	k
c130.001	0	0	0	0	1	0	0	1	1	0	0
c130.002	0	0	0	0	1	0	0	1	1	0	0
c130.003	0	0	1	0	1	0	0	0	0	0	1
c130.004	0	0	0	0	1	0	0	0	1	0	0
c130.005	0	0	0	0	1	0	0	0	1	0	0
c130.006	0	0	0	1	0	1	0	0	1	0	0
c141.007	0	1	1	0	1	1	0	0	2	1	1
c141.008	0	1	1	0	1	1	1	0	1	0	1
c141.009	1	1	0	0	2	1	0	0	1	0	0
c141.010	0	1	0	0	1	0	0	0	2	1	1
c5.011	1	0	0	0	1	0	1	0	1	0	2
Asphalt	0	0	0	1	1	0	0	0	0	0	0

Close This Window

Figure 7.2. Load Plan Display. The above dialog box appears when a planner requests a RAP load solution. The last row indicates how many cargo items remain on the asphalt. This display shows the first infeasible RAP load plan, which does not load one item *d* and one item *e*.

D. PLANNER GUIDANCE IS ALWAYS KEY

1. How the Planner Can Control Loads

Negotiating with RAP is primarily based on intuitive trail and error parameter adjustment. For this sample problem, adjusting loading goals is all that is required to create a feasible load plan from an infeasible one. The sample problem approach is:

- Observe what cargo item types are not loaded,
- Lock these numbers on aircraft,
- Unload newly locked aircraft (at least two aircraft), and
- Resolve.

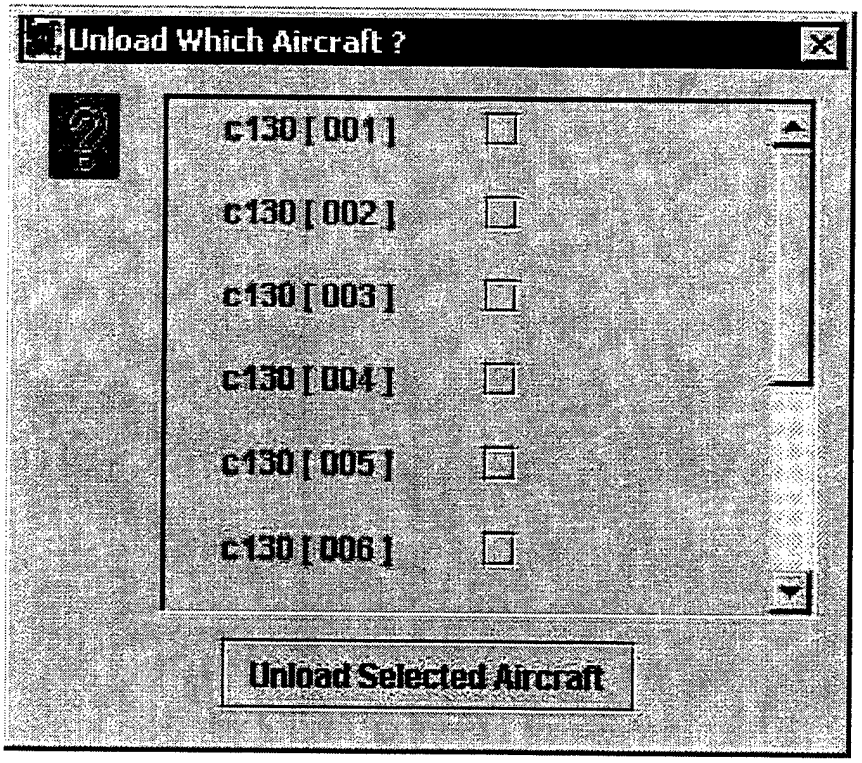

Figure 7.3. Unloading Aircraft. This display lists all aircraft in the current mission. A planner that chooses to remove all loaded cargo items from an aircraft and put them back on the asphalt, selects the aircraft to download and clicks the "Edit Selected" button. No aircraft have been selected in this example.

2. Suggesting Specific Templates By Locking Guidance

For any aircraft in RAP, the use of locked upper and lower guidance may select a specific template for that aircraft, if such a template exists and there is enough candidate cargo to fit the template.

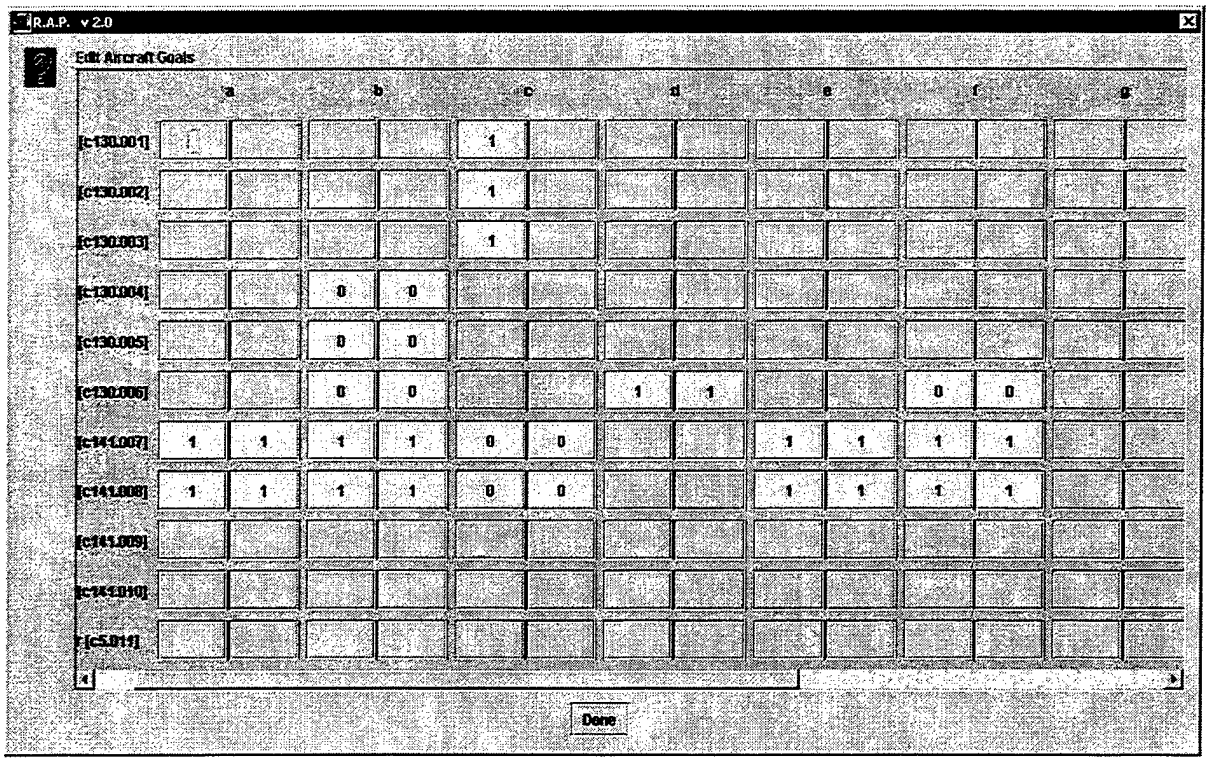

Figure 7.5. Final Negotiation. During negotiation, many RAP parameters can be adjusted to test alternate feasible loads. Aircraft loading guidance has the most impact. The planner graphically loads items on one aircraft and forbids others from being considered. For example, the planner wants at least one item *c* on aircraft c130.001 but does not want any items *b* on c130.004. The above figure shows the final adjustments the author made to the initially infeasible RAP load plan that made it feasible.

Optimization

	a	b	c	d	e	f	g	h	l	j	k
c5_001				1	1		1		3	1	2
c141_001		1			1	1	1		2		
c141_002		1			1	1			1		1
c141_003	1	1			2	1			1		
c141_004	1	1			2	1			1		
c130_001			1		1						1
c130_002					1			1	1		
c130_003					1			1	1		
c130_004			1		1						1
c130_005			1		1						1
c130_006				1					2	1	
	2	4	3	2	12	4	2	2	12	2	6

RAP

	a	b	c	d	e	f	g	h	l	j	k
c5_001				1	1		1		3		1
c141_001	1	1			2	1			1		
c141_002	1	1			2	1			1		
c141_003		1			1	1			2	1	1
c141_004		1			1	1	1		1		1
c130_001			1		1						1
c130_002			1		1						1
c130_003			1		1						1
c130_004					1			1	1		
c130_005					1			1	1		
c130_006				1					2	1	
	2	4	3	2	12	4	2	2	12	2	6

Figure 7.6. Final Load Plans. The final RAP load plan again differs from the optimization load plan. Initially RAP returns an infeasible solution to this air load problem because the approved template list in this example is much smaller than before.

VIII. CONCLUSIONS AND RECOMMENDATIONS

A. CONCLUSIONS

While specific Ranger operations do not receive much media attention, their impact has significant impact on the United States Army operational successes as evident in Operation Just Cause. Had Rangers not had the time to properly plan their airborne assault into Panama, the success of US operations in the region could have been seriously impacted.

The Ranger Air Load Planner provides a man-in-the-loop decision support tool an assistant operations officer can employ when constructing an air load plan. This tool is particularly valuable during air load plan negotiations with subordinate unit commanders. RAP returns solutions in less than a second, and instant plan visualization could be key to simplifying negotiations. With additional testing, to include validation with classified data, RAP may provide a key tool to accelerate effective airborne assault planning in the future --- and time is precious in special operations.

RAP is platform independent and runs in Java which is available free of charge [Sun Microsystems, 1998]. A Java programmer with experience in computation and data structures can easily maintain RAP.

RAP demonstrates the capability to put high-quality, simple-to-use operations research tools in the hands of the operational decision-maker. A validated version of RAP would have a positive impact on mission accomplishment by saving mission planning time and increasing time available for rehearsals. Funding for formal RAP development support, and deployment is highly encouraged.

B. RECOMMENDATIONS

RAP does not address the construction of 1-bump plans. The sole reason we cannot directly produce 1-bump plans is that the approved templates the Rangers use do not include a computer-readable encoding of loaded item ordering. Without some generic representation (preferably in some unambiguous digital code), we cannot deduce the order in which cargo must be loaded, unloaded, or, in particular, for 1-bump planning, added to an existing load without disturbing cargo already loaded.

We can approximate 1-bump plans by reserving the right amount of capacity on each loaded aircraft to absorb cargo on a failed aircraft. But we cannot be sure of a feasible reloading.

We are concerned that lack of available load template ordering may inhibit manual air load planners as well, unnecessarily raising the level of abstraction and uncertainty at precisely the time the air operations officer needs to concentrate and complete 1-bump plans quickly. This is a deficiency that we would like to help repair.

LIST OF REFERENCES

Brooke, A.,Kendrick, D., Meeraus, A. and Raman R., [1997], GAMS: A User's Guide, The Scientific Press.

Cornell, Gary, Horstmann, C., [1997], Core Java, Sun Microsystems Press.

ILOG [1999] "CPLEX 6.5," [http://www.cplex.com].

Internet Search [1999], [http://www.yahoo.com/]

Martins, Gustavo, [1999], Template Synthesizer [Java computer program contributed by a Naval Postgraduate School doctoral student].

Rardin, Ronald L. [1998], Optimization In Operations Research, Prentice Hall.

Sun Microsystems [1998], "Java 2,"[Computer programming language].

The Aviation Zone [2000], [http://www.theaviationzone.com/images.htm].

United States Air Force [1999], [http://www.af.mil/photos/].

United States Army [1990] Field Manual No.(FM) 90-26, *Airborne Operations*, Headquarters, Department of the Army, 18 Dec.

United States Army [1993] Field Manual No.(FM) 100-5, *Operations*, Headquarters, Department of the Army, 14 June.

THIS PAGE INTENTIONALLY LEFT BLANK

INITIAL DISTRIBUTION LIST

1. Defense Technical Information Center 2
   8725 John J. Kingman Rd., STE 0944
   Ft. Belvoir, VA 22060-6218

2. Dudley Knox Library 2
   Naval Postgraduate School
   411 Dyer Rd.
   Monterey, CA 93943-5101

3. Professor Gordon H. Bradley, Code OR/BZ 3
   Department of Operations Research
   Naval Postgraduate School
   Monterey, CA 93943-5000

4. Professor Gerald G. Brown, Code OR/BW 3
   Department of Operations Research
   Naval Postgraduate School
   Monterey, CA 93943-5000

5. LTC Joel R. Parker, Code OR/JP 3
   Department of Operations Research
   Naval Postgraduate School
   Monterey, CA 93943-5000

6. Air Force Office of Scientific Research 1
   Attn: Dr. Neal Glassman
   801 North Randolph Street
   Arlington, VA 22203-1977

7. CPT Maximo A. Moore III., USA 1
   46 Glen Gray Rd.
   Mahwah, NJ 07430