

AD-A152 007

DEPLOY: AN INTERACTIVE GOAL PROGRAMMING MODEL FOR THE
RAPID DEPLOYMENT OF... (U) AIR FORCE INST OF TECH
WRIGHT-PATTERSON AFB OH SCHOOL OF ENGI... D O TATE

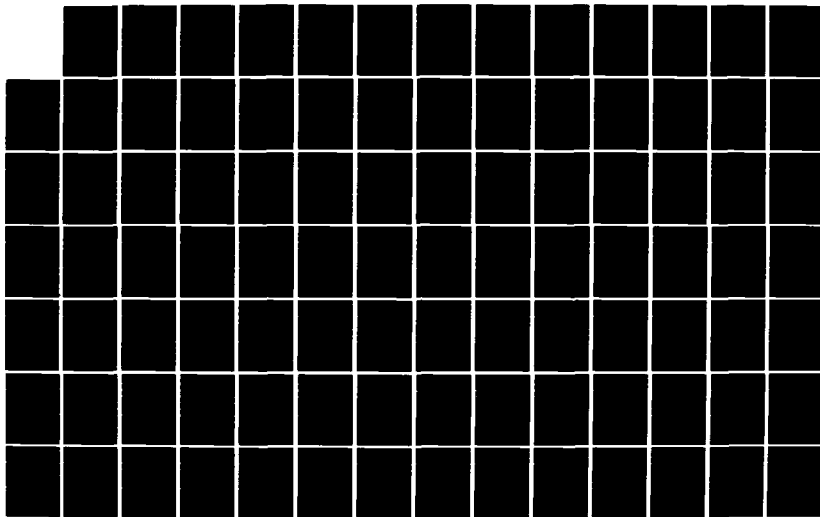
1/3

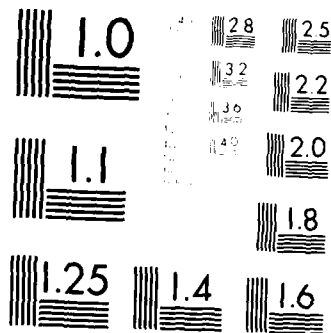
UNCLASSIFIED

06 DEC 84 AFIT/GOR/OS/84D-14

F/G 15/7

NL





MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A

①

AD-A152 007

DEPLOY:
An Interactive
Goal Programming Model
for the
Rapid Deployment of Armed Forces

THESIS

David C. Tate
Captain USAF

AFIT/SCF/OS/84D-14

DTIC FILE COPY

DTIC
MAR 28 1985
B

Approved for public release; distribution unlimited

85 03 13 160

SECURITY CLASSIFICATION OF THIS PAGE

REPORT DOCUMENTATION PAGE									
1a. REPORT SECURITY CLASSIFICATION UNCLASSIFIED					1b. RESTRICTIVE MARKINGS				
2a. SECURITY CLASSIFICATION AUTHORITY					3. DISTRIBUTION/AVAILABILITY OF REPORT Approved for public release; distribution unlimited.				
2b. DECLASSIFICATION/DOWNGRADING SCHEDULE									
4. PERFORMING ORGANIZATION REPORT NUMBER(S) AFIT/GOR/OS/84D-14					5. MONITORING ORGANIZATION REPORT NUMBER(S)				
6a. NAME OF PERFORMING ORGANIZATION School of Engineering			6b. OFFICE SYMBOL (If applicable) AFIT/ENS		7a. NAME OF MONITORING ORGANIZATION				
6c. ADDRESS (City, State and ZIP Code) Air Force Institute of Technology(AU) Wright-Patterson AFB, Ohio 45433					7b. ADDRESS (City, State and ZIP Code)				
8a. NAME OF FUNDING/SPONSORING ORGANIZATION			8b. OFFICE SYMBOL (If applicable)		9. PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER				
8c. ADDRESS (City, State and ZIP Code)					10. SOURCE OF FUNDING NOS				
					PROGRAM ELEMENT NO.		PROJECT NO.		TASK NO.
11. TITLE (Include Security Classification) See box 19									
12. PERSONAL AUTHOR(S) David O. Tate, Capt, USAF									
13a. TYPE OF REPORT MS Thesis			13b. TIME COVERED FROM _____ TO _____		14. DATE OF REPORT (Yr, Mo., Day) 6 December 1984			15. PAGE COUNT 208	
16. SUPPLEMENTARY NOTATION									
17. COSATI CODES			18. SUBJECT TERMS (Continue on reverse if necessary and identify by block number) Goal Programming, Air Transportation, Military Transportation, Theater Level Operations, Optimization						
FIELD	GROUP	SUB. GR.							
15	07								
19. ABSTRACT (Continue on reverse if necessary and identify by block number) Title: DEPLOY: An Interactive Goal Programming Model for the Rapid Deployment of Armed Forces Thesis Advisor: Ivy D. Cook, Jr, Lt Col, Ph.D., USAF James K. Feldman, Maj, Ph.D., USAF									
20. DISTRIBUTION/AVAILABILITY OF ABSTRACT UNCLASSIFIED/UNLIMITED ☒ SAME AS RPT ☐ DTIC USERS ☐					21. ABSTRACT SECURITY CLASSIFICATION UNCLASSIFIED				
22a. NAME OF RESPONSIBLE INDIVIDUAL Ivy D. Cook, Jr., Lt Col, Ph.D., USAF					22b. TELEPHONE NUMBER (Include Area Code) (513) 255-3362			22c. OFFICE SYMBOL AFIT/ENS	

Approved for public release: IAW AFR 190-17.
 JOHN F. WILSON, Professional Development
 F-100
 10/1/68

DEPLOY is an interactive Goal Programming Model for the Rapid Deployment of Armed Forces which offers decisive advantages over any current methodology. The front-end user-friendly package allows the user to easily enter the necessary data either interactively or via external files. Furthermore, data files can be easily created and altered to perform sensitivity analysis on any of the parameters in the model.

DEPLOY accounts for both intertheater and intratheater airlift, and can be used to optimally plan movement schedules for predetermined forces or optimally choose and move a force from a list of available units and airlift resources to meet specified goals. In addition, further analysis can be performed to determine the least number of aircraft or the least costly aircraft inventory necessary to accomplish the specified goals.

This report describes the physical and mathematical, scenario limitations, and input requirements of DEPLOY. Further, the report includes a user guide, variable definitions, subroutine definitions, and computer listings. The scenario used to demonstrate DEPLOY used 188 decision variables and 135 separate equations. Finally, this demonstration serves as an example to any potential users of DEPLOY.

DEPLOY
An Interactive
Goal Programming Model
For the
Rapid Deployment of Armed Forces

THESIS

Presented to the Faculty of the School of Engineering
of the Air Force Institute of Technology
Air University

In Partial Fulfillment of the
Requirements for the Degree of
Master of Science

by: David G. Tate
Captain USAF

Graduate Operations Research

December 1964

Approved for public release; distribution unlimited

Preface

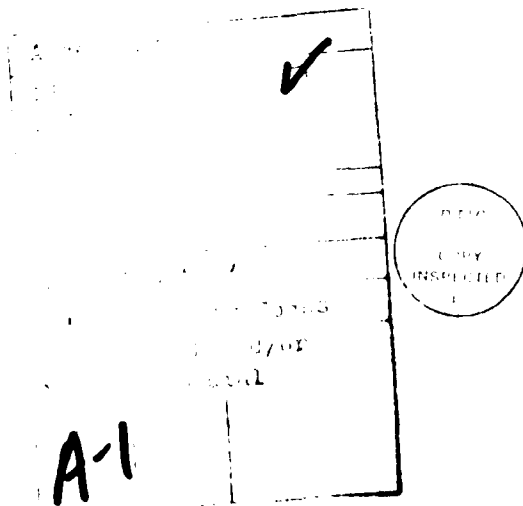
DEPLOY is an interactive Goal Programming Model for the rapid deployment of armed forces. The project was suggested by LTC. Ivy D. Cook as a continuation of Captian James Cooke's thesis.

Special thanks go to my thesis advisor LTC. Cook who not only gave me a wide latitude to work independently, but also gave many pertinent suggestions along the way. I am also grateful to Major James K. Feldman, my reader, whose words of encouragement always came at the right time.

Furthermore, I am especially indebted to my wife, Linda, and son, Bryan, whose love and needed distractions helped me keep my perspective and enjoy life in the midst of this overwhelming project.

Finally, I acknowledge my Lord and Savior, Jesus Christ, for giving me the true meaning and perspective of life. This is best stated up in Psalms 73:25 which says, "Whom have I in heaven but Thee? And besides Thee, I desire nothing on earth."

David G. Tate



Contents

Preface	ii
List of Tables	v
Abstract	vi
I. Introduction	1-1
Background	1-2
Problem Statement	1-4
Research Objective	1-4
Research Scope	1-4
Research Methodology	1-5
Format Structure	1-6
II. Literature Review	2-1
Aircraft Loader Model	2-1
Aircraft Loading Model (ALM)	2-2
Air Movement Planning System (AMPS)	2-4
DELIVER	2-4
M-14 (Model 14)	2-5
Military Airlift Capability Estimator (MACE)	2-6
MACRO Model 12	2-7
POSTURE System	2-8
Rapid Intertheater Deployment Simulator (RAPIDISM)	2-9
RECIEVER ONE	2-10
Simulation for Transportation Analysis and Planning (SITAP)	2-11
Transportation Model (TRANSMO)	2-12
III. Model Developement	3-1
Variable Parameters	3-1
Goals	3-5
Bounding Constraints	3-10
Aircraft Limitations	3-10
Airport Limitations	3-13
Unit Limitations	3-17
Shipment of Units	3-17
Shipment of Supplies	3-19
Unit Linkage	3-24
Unit Abilities	3-25
IV. The General Model	4-1
Goal Programming	4-1
PAGP	4-4

V.	Model Characteristics	5-1
	Limitations	5-1
	Model Restrictions	5-1
	Scenario Limitations	5-3
	Input Requirements	5-5
VI.	Verification and Demonstration	6-1
	Verification	6-1
	Matrix Formulation	6-1
	Sensitivity	6-3
	PAGP	6-3
	Demonstration	6-3
	Input Parameters	6-4
	Output Results	6-6
VII.	Conclusion	
	Model Summary	7-1
	Major Accomplishments	7-1
	Computer Package	7-2
	Types of Analysis	7-2
	Flexibility	7-3
	Recommendations	7-4
	Bibliography	8-1
	Appendix A: User Guide	A-1
	Appendix B: DEPLOY Subroutine Listing	B-1
	Appendix C: DEPLOY Variable listing	C-1
	Appendix D: DEPLOY listing	D-1
	Appendix E: Deploy Flowchart	E-1
	Appendix F: PAGP Subroutine Listing	F-1
	Appendix G: PAGP Variable listing	G-1
	Appendix H: PAGP Flowcharts	H-1
	Appendix I: PAGP listing	I-1

List of Tables

Tables		Page
5-1	Input Requirements for the Aircraft Inventory	5-5
5-2	Input Requirements for the APDD Complex	5-5
5-3	Input Requirements for the Deployable Units	5-6
5-4	Input Requirements for the Goals	5-6
6-1	Constraint Output	6-2
6-2	Aircraft Data	6-4
6-3	APDD Data	6-5
6-4	Deployable Units Data	6-5
6-5	Goals for Deploy	6-5
6-6	Simulated FAGP Output	6-7
A-1	Example of Aircraft Data File	A-4
A-2	Example of APDD Data File	A-5
A-3	Example of Unit Data File	A-6
A-4	Menu of the Different Types of Analysis	A-7
A-5	Menu to Define the Goals	A-8
A-6	Example of Defined Goals	A-8
A-7	Sensitivity Analysis Main Menu	A-9
A-8	Aircraft Submenu	A-9
A-9	APDD Submenu	A-10
A-10	Deployable Units Submenu	A-10

ABSTRACT

DEPLOY is an interactive Goal Programming Model for the Rapid Deployment of Armed Forces which offers decisive advantages over any current methodology. The front-end user-friendly package allows the user to easily enter the necessary data either interactively or via external files. Furthermore, data files can be easily created and altered to perform sensitivity analysis on any of the parameters in the model.

DEPLOY accounts for both intertheater and intratheater airlift, and can be used to optimally plan movement schedules for predetermined forces or optimally choose and move a force from a list of available units and airlift resources to meet specified goals. In addition, further analysis can be performed to determine the least number of aircraft or the least costly aircraft inventory necessary to accomplish the specified goals.

This report describes the physical and mathematical, scenario limitations, and input requirements of DEPLOY. Further, the report includes a user guide, variable definitions, subroutine definitions, and computer listings. The scenario used to demonstrate DEPLOY used 188 decision variables and 126 separate equations. Finally, this demonstration serves as an example to any potential users of DEPLOY.

DEPLOY
An Interactive
Goal Programming Model
for the
Rapid Deployment of Armed Forces

I. Introduction

In 1984 Army Captain James Cooke, an AFIT/GST student, demonstrated for the first time a particular methodology which determines how the interactions of unit weight, combat attributes, logistics needs, and airlift resources can be jointly optimized.(7,4) In his thesis he determined analytically the optimal force mix or the incremental advantage in deployed power attainable by an incremental change in airlift resources. The thesis DEPLOY was developed in response to recommendations in Cooke's thesis "Optimizing Force Development and Force Structure for the Rapid Deployment Force." (7:ff) The thesis by Cooke originated from "a perceived need to approach the deployment of combat forces using a total systems methodology, rather than the present method of separate mission/service optimization studies." (7:1) The model developed here takes the methodology demonstrated by Cooke and expands it to an interactive user-friendly computer program. Furthermore, this model allows the user to "generate unit movement and force structure requirements for the Rapid Deployment Force." (7:132) A background section is included in this chapter to inform the reader of the perceived need for a project like DEPLOY. Also included are sections on the Research Scope and Methodology. (7:4-8)

Background

In 1980, The United States government indicated a need for the capability to move large conventional ground forces to any location within a broad geographic area centered on Southwest Asia. As a result of this requirement, the Rapid Deployment Joint Task Force (RDJTF), now redesignated the Central Command (CENTCOM), was formed. This force consists of the 82nd Airborne Division, the 101st Airborne Division (Airmobile), the 24th Infantry Division, two Ranger battalions, and Air Force, Navy and Marine forces. The RDJTF Army units are all forces which facilitate air transportability.

Although CENTCOM was developed in direct response to a perceived threat, the Air Force has insufficient aircraft resources to support the full and rapid deployment of CENTCOM assets. The Army, in its deployment planning, can consider only the inventory of the Military Airlift Command (MAC) assets. When planning a deployment, therefore, critical decisions must be made as to how best to use limited air transportation. The basic decision is to increase the combat potential available to the ground commanders at the risk of reducing sustainability of the already deployed force. Combat potential is a complex and currently subjective index of the fighting capabilities of the force. These capabilities are based on deployed troops, tanks, and artillery. Sustainability is thought of as the continued ability of a force to fight through maintenance of in the event its supply line.

The Army must plan the deployment of its troops in such a

way as to balance combat potential with sustainability. Currently, scenario war plans are developed by operations analysts, who determine what forces they believe are required and by what date the forces are needed for a given scenario. Logisticians then calculate required supplies to support the force, using tabulated data. Given this data, the transportation officers attempt to move the required force packages by the required dates. If this is deemed impossible, the force planners either extend the required due dates or reduce the planned force.

The implementation of this process is a result of Under Secretary of Defense Chayes' report requiring between 50 and 150 C-5A equivalents in airlift to meet urgent national security needs. (3:20) Limited airlift assets prevent the Army from moving all cargo and troops required at a single time to respond to a major threat. Therefore, decision makers must be able to evaluate, at each point in time, how best to use the forces and resources available. This could be done by incorporating into the process the reduction of airlift resources caused by each force unit's "tail", the logistical requirements of the force.

Although this is a critical consideration for U.S. planners, research to date has addressed the problem only peripherally. The U.S. Army War College 1983 list of "strategic issues...most relevant and urgent for the Army to study" specifically identifies the issue of sustainability and mobility as an unsolved research area. Col. Daley, Chief of the War College Military Operations and Planning Group; LTC. Murphy, Air Force Liaison to the Doctrine and Force Issues Group of the War College Military Strategic Department; and Mr. J.E. Trinnamen, head of

There are model limitations and time requirements that are important when using RAPIDISM. All significant limitations in the model are computer dependent; therefore, given a computer with enough core, limitations are reduced to nil. The time considerations will be discussed now. After requiring approximately two weeks to acquire the data base, another one week to two man-months are necessary to structure the data into the proper format. The computer time is about 1 minute per model cycle and the model requires four hours to two man-days to analyze and evaluate the output.

J. RECEIVER ONE

RECEIVER ONE was developed and is used by HQ USAF/SAGM five times a week. This model analyzes the problem of "airfield saturation during force deployment." Each aircraft in the model is considered from the approach to landing until the takeoff departure. Included also in the model are the possible effects of weather, interdiction, and air base losses. There are basically three input parameters for RECEIVER ONE.

1. Base characteristics, such as parking space and amount of material handling equipment (MHE)
2. Aircraft characteristics, such as aircraft payload
3. Average expected arrivals of each type aircraft at each destination APOD (may change daily).

The output information is as follows:

1. Computer printouts on utilization of APOD facilities
2. Queueing delays at various stages of base processing
3. Aircraft elapsed times
4. Tons delivered
5. Tons diverted from the primary APOD.

The model can examine up to seven different APOD's and twenty

level of deployment activities, and basing and readiness levels of resources." (33:515)

There are limitations to this model and time requirements that must also be considered. Some limitations are that the vehicles are fractionalized and all events in the model are deterministic. Also, the cargo requirement integrity is not maintained in the model. This model requires two weeks to acquire the data base, and one week to two man-months to structure the data into the proper format, about 40 minutes to an hour of CPU time is required, and four hours to two man-days are necessary to analyze and evaluate the results.

I. Rapid Intertheater Deployment Simulator (RAPIDISM)

RAPIDISM is used by the JCS (J-4) approximately 260 times a year and was developed by the General Research Corporation. This model simulates the rapid deployment of combat units along with the resupply of these units for a military contingency operation. By using this model, the minimum time to deliver the units to these destinations via the air or the sea is determined. Different priorities can be assigned to the units and supplies. The model is deterministic in nature. The input parameters of the model are as follows:

1. Available number of aircraft by class
2. Available number of ships by class
3. APOEs, convoy route, and APODs
4. Transportation modes
5. Time periods for initial ship availability
6. Commodities and units
7. Attrition rate of vehicles (33:539)

As a result of this input, a detailed log of movements and numerous summary reports are produced.

3. Specified locations are aggregated into notational locations
4. Aircrew resources are not addressed
5. Tanker aircraft are assumed available by the model. (33:423)

This model requires only three man-hours to structure the input format. Then after about 30 minutes of CPU time per 90-day war scenario, 6 man-hours are necessary to analyze and evaluate the results.

H. POSTURE System

POSTURE is used by the JCS (J-4) approximately five times a year and was developed by the General Research Corporation. This model determines "the strategic mobility resources required for contingency situations and assesses the delivery capability of a given set of resources." The model accomplishes this by discovering the least-cost system to meet the stated requirements. Further, the model can determine the maximum deployment capability of a given set of mobility resources. In order to accomplish this, linear programming is utilized. The model utilizes land, air and sea mobility resources. The input parameters are as follows:

1. Origin/destination sets for force transfers
2. List resources
3. Cost parameters for the resources
4. Time-phased requirements by contingency and unit type
5. Vehicle characteristics, speed, payload
6. Allowable routes and route distances
7. Operational delay assumptions
8. Attrition factors (if used)
9. Convoy limits by theater and time period
10. Resource availability
11. Cargo characteristics.

The result of this input is a "10-year system cost, fleet sizes,

G. MACRO Model 12

The MACRO Model 12 is used by MAC Headquarters and was developed by the MACRO Task Force. The model measures "the total delivery performance of the MAC system reflecting the effects of aircraft queueing." Other effects that are considered are: station denial, air refueling, alternate routing, various fleet mixes, and limited ground support. The model is stochastic in nature and uses Q-GERT simulation language as the primary solution technique. The input requirements are divided into four different areas as follows:

1. Applied aircraft (number and type)
2. Initial aircraft availability schedule
3. Cargo requirements (from-to by cargo class)
4. Current configuration of wartime scenario (if different from the present configuration).

From this input the model gives the following output:

1. Flying hour requirements
2. Aircraft UTE rates
3. Route usage by leg segment
4. Aircraft waiting times
5. Movement closure time
6. Closure by cargo class and aircraft type
7. Delivery rate by cargo class
8. Location work load
9. Number of aircraft queueing by location
10. Aircraft handling requirements
11. Ground time histograms
12. System onload/offload history
13. Periodic reports reflecting aircraft and cargo status by location or various time intervals
14. Standard Q-GERT output including trace options
15. Designed user-specified output as desired. (33:423)

There are model limitations and time requirements that must be considered in using the MACRO-12 model. There are five major limitations to the model as follows:

1. All cargo is measured in terms of aircraft loads
2. The routing algorithm is completely probabilistic

F. Military Airlift Capability Estimator (MACE)

The Military Airlift Capability Estimator (MACE) is used by the JCS (J-4) and was developed by MAC. The model assists the transportation planner to quickly estimate the force closure times for the airlift of large-scale troop and cargo movements. The model is used about 75 times a year. The input parameters are three-fold; force definitions, aircraft ground time, and different requirements such as the distance between the APOD and Aerial Port of Embarkation (APOE). As a result of this input, the following output is given:

1. Schedule of the daily movement capability of the aircraft employed
2. Closure times at the destination of the force being moved
3. Individual requirement tracers
4. Aircraft mission traces
5. Aircraft utilization summaries
6. Requirement closure summaries. (33:412)

There are limitations and time requirement problems that are associated with the use of the MACE model. The first of the three limitations is that airlift is the only mode of transportation considered. Second, "aircraft can be prepositioned for the first acquisition only," after this, the aircraft are just positioned where needed. Third, the "time-phased processing of requirements" is not considered. Also, there are several time factors to be considered. After acquiring the data base which takes 1 man-month, 1-1/2 man-weeks are required to structure the data into the model input format. Then, after approximately 30 minutes of CPU time, 1-1/2 man-weeks are needed to analyze and evaluate the results. (33:412)

has three basic input parameters to the model.

1. Sorties required to move a given unit by different aircraft types
2. The number and Utilization (UTE) rate of each type of aircraft
3. The flight time for a round trip.

As a result of this input, the model gives the transportation completion times of the different cargo types and of the entire project. The model can utilize a maximum of six different aircraft types, only three of which can be capable of handling outsize cargo. (33:229)

E. M-14 (Model 14)

Model 14 was developed and is used by HQ MAC/XPSR about once a month. The model identifies and helps resolve the "strategic aircraft chokepoints at the airbase level during wartime surge situations." The model is a computer simulation of the MAC system involving four types of aircraft and 422 airbases. The aircraft are heuristically "routed to use idle resources and to avoid facility and personnel saturation." There are two main input parameters.

1. Station data, e.g., location, resources, climatology
2. Aircraft operating envelopes as movement requirements and policies.

The output generated is the raw form of "mission itineraries, requirement histories, GASP statistics, queueing files, and other simulation information." From the examination of this output many different types of studies can be performed. This model requires approximately two man-weeks to acquire the data base and to sequence the data into the proper format. Finally, one man-week is required to analyze the data. (33:412)

analysis time are variable with the size of the problem.

C. Air Movement Planning System (AMPS)

The Air Movement Planning System (AMPS) is used and was developed by the US Army Logistic Center. This model optimizes a given load plan to determine its feasibility. It is accomplished by planning, diagramming and manifesting individual "aircraft loads of equipment and personnel for movement on C-5, C-141, and C-130 aircraft." (33:43) To optimize the load plan, specific characteristic, balance and safety constraints are considered. There are two input parameters into the model, the cargo and aircraft list. The output is a schematic of the load plan and a manifest of the cargo and passengers.

There are limitations and time requirements that are associated with the AMPS model. The first limitation is that the cargo is considered as cubes with a certain weight and center of gravity, rather than as a specific piece of cargo. As a result of this limitation, such characteristics as axle location and vehicle overhang are not considered. The second limitation is that the "vehicle tie-down space is determined on the worst case basis rather than specifics." (33:43) When considering the time requirements for this model, one man-month is required to prepare the data base for the input parameters. Further, once the input is made, one hour of CPU time is needed to determine the output.

D. DELIVER

The DELIVER model was developed and is used by HQ USAF/SAGM about 250 times a month. The model estimates "the time to deliver a given military force by airlift [by providing] computerized analysis of sortie output from the ALM." The model

aircraft." This model is further used not only to evaluate current and future aircraft designs but also to determine the effect of changes in military vehicles. The model "determines the number of sorties required to load a military force of any size." The input parameters are in the form of three input files: vehicle characteristics, unit description, and an overall input file. These files are further described as follows:

1. Vehicle characteristics are dimensions and weight.
2. Unit descriptions include the number of personnel, weight of unit equipment, and the number of each type of vehicle.
3. The overall input file includes aircraft descriptions, unit accompanying supply formulae, aircraft loading sequences, selections of units, and input and output options.

The output of this model can give a multitude of information, including individual loads and "distribution of loads by weight and aircraft type." Furthermore, the loadability of vehicles is given along with "unit and overall summaries of payload and sortie statistics." (33:37)

The ALM is very versatile as it can use a multitude of different input parameters. The model is limited to considering 10 different aircraft types, 999 vehicle types, and 1,000 units. Each aircraft cargo compartment can be divided into ten different sections, and "up to ten groups of vehicles can be limited or excluded from an aircraft." (33:37) Furthermore, one type of special load can be forced into any aircraft per load.

The time requirements to use the ALM are variable. The CPU time to run the model on a Mechanized Brigade requires approximately 45 seconds. Both the preparation and the output

As a result of this input, the model gives a "statement of loadings of each aircraft by chalk number, consisting of a detailed listing of each 'loaded' aircraft." Also, a summary is given that yields the following specific output:

1. Number of sorties required
2. Cargo and passengers not loaded (i.e. those things that are too large, too heavy, or passengers for whom there are no seats on the aircraft)
3. Number, weight, and floor space of vehicles and bulk cargo loaded
4. Number of passengers loaded
5. Total fleet weight, floor space, and passenger seats that were available for loading. (33:29)

There are two major limitations to this model. First, the "sortie generation" technique does not necessarily give the optimal loading plan. As a result, the number of sorties required for the mission is not necessarily a minimum estimate. Second, the possibility of using a mixed type of fleet is not available as only one type of aircraft is considered.

When using this model there are a few time requirements to be considered. Approximately one month is required to acquire the data base for the input. Once the data base is finished, very little, if any, work must be done to format the data base for the input. Once inputted, approximately 10 minutes of CPU time per model is required to compute the output. Finally, it takes one man-day to analyze and evaluate the output.

B. Aircraft Loading Model (ALM)

The Aircraft Loading Model (ALM) is used and was developed by USAF/SAGM. Furthermore, the model is used about 100 times per month. The purpose of the model is to "provide computerized analysis of the loadability of military vehicles on airlift

II. Literature Review

This chapter will review the different strategic deployment models that are presently being used in the military. Several models will be discussed, but the central theme is the movement of personnel and cargo via airlift. The user and the developer of the model will be discussed first, followed by the purpose and description of the model. Then the input parameters will be examined along with the output statements. Finally, the model limitations and time requirements of the model will be considered.

A. Aircraft Loader Model

The Aircraft Loader Model is used by the JCS (J-4) and was developed by the Institute of Defense Analysis (DIA). The purpose of the model is to estimate "the number of airlift aircraft required to perform a stated transport mission." To accomplish this purpose, the model simulates loading the aircraft. Further, this model can be used to plan operations, compare aircraft sorties required for different aircraft types, and study alternative loading configurations. Numerical analysis is used to solve the mathematics of the model. This model is used approximately ten times per year. Some of the input parameters of the model are as follows:

1. Weight allowable cabin load (WACL) for the aircraft type and range of operation
2. Dimensions of the cargo compartment
3. Number of passenger seats available
4. Allowable stacking height for the bulk cargo
5. A complete vehicle and bulk cargo list with their dimensions
6. The number of passengers and their total weight.

models. In each model the input and output parameters will be examined. Finally, prevailing assumptions and approaches of the different models will be addressed.

Chapter III will present the mathematical formulation of the model utilizing the work of Capt Cooke as much as possible. This will include the formulation of the goals and constraints.

Chapter IV will review goal programming and present a description of the assignment algorithm, PAGP.

Chapter V will present some characteristics of DEPLOY. This will include model restrictions, scenario limitations, and input requirements. Chapter VI will provide both a verification and a demonstration of DEPLOY.

Chapter VII will present a summary of the modeling process, a description of the uses of DEPLOY, and a discussion of the potential improvements of the model.

variance from them is allowable but undesirable, are the following:

1. Keep a certain fixed or minimum ratio between combat and combat support units.

2. Keep unit sustainability, as reflected by percentage of required supplies delivered, at a fixed level.

3. Maximize the anti-tank strength, defensive frontage or combat power of the deployed force.

4. Maximize utilization of the Air Force airlift assets, in tonnage and allowable cubage.

5. There is a limit on the number of each type of aircraft available.

Constraints which are absolute are the following:

1. Outsize cargo will fit only outsize capable aircraft. Oversize cargo will fit on outsize aircraft or on oversize capable aircraft. Bulk cargo will fit on all cargo-carrying aircraft.

2. A deployed unit consumes supplies at a fixed rate over the entire period of analysis.

3. All aircraft types are capable of landing at the Aerial Port of Debarkation (APOD).

4. Unit sustainability has a floor. United States National Command Authority will not build up force strength at the expense of letting troops already deployed run out of ammunition or food.

5. Supplies delivered to the APOD must be trucked or airlifted to the combat forces at the front.

Format Structure

Chapter II will review many of the currently used deployment

address only air-deployed and air-supported units. The close presence, in a crisis, of logistics ships or Marine Amphibious units is fortuitous, but cannot be relied upon with worldwide commitments.

2. Once deployed into the theater, the standard supply consumption rates will be used.

3. This study will incorporate any type of military or Civilian Reserve Air Fleet (CRAF) aircraft to move CENTCOM assets.

4. Sustainability will be kept as a constant measure across the entire deployed force. That is, all forces, if unsupplied, would run out of all supplies at the same time. This is assumed as a logical result of proper logistical planning, as it would not be acceptable, in combat, to run out of ammunition but to have ten days of food available.

5. The input data file may either already exist, be created interactively, or be changed interactively and saved if necessary for later use.

Research Methodology

The overall objective of this study is to develop DEPLOY, a user-friendly computer program to maximize combat power delivered to a theater as a function of time without diminishing a certain sustainability of the forces already deployed. This problem is broken into two subordinate and equally important goals: maximize the combat power delivered, and minimize the delivery time. These goals are optimized only after certain constraints are satisfied. Constraints which can be expressed as goals, because

Mediterranean Studies for the Strategic Studies Institute, have all separately proposed that immediate research be conducted on this topic. (7,18-26)

Problem Statement

The methodology demonstrated by Capt Cooke exists in mathematical form, and is therefore not immediately helpful to the user. There is no interactive user-friendly computer program that performs the kind of analysis demonstrated by Capt Cooke.

Research Objective

The primary objective of this research effort is to develop an interactive user-friendly computer program to optimize combat power delivered to a theater during a specified time, within acceptable levels of force sustainability. The methodology demonstrated by Capt Cooke will be examined concerning its completeness and extended from its mathematical form to an interactive computer program. This program will be composed of two parts, the construction of constraints and the allocation of deployed units with aircraft. The information necessary to build the goals and constraints will be input in such a way that the user can easily change parameters and goals as desired. The allocation routine, PAGP, will minimize the deviations from the user-defined goals. Here, the objective measure of merit simultaneously incorporates unit weights, requirements, combat power attributes, and in-theater constraints to find the global optimal solution.

Research Scope

1. With the emphasis on rapid deployment, this study will

different aircraft types. In order to create a scenario, eight to ten man-hours are needed, while only one hour is needed for moderate changes in the scenario. Once the model is complete, only one to two hours is needed to analyze the output. (33:555)

K. Simulation for Transportation Analysis and Planning (SITAP)

The SITAP model is used by the JCS (J-4) about 50 times a year and was developed by the Computer Sciences Corporation (CSC). This model analyzes a wide spectrum of transportation vehicles and uses a network system to complete the analysis. The "different movement demands, vehicles, and defined network are controlled by the analyst." The input parameters are as follows:

1. The network design
2. Vehicle characteristics and movement
3. Cargo description and quantities.

As a result of this input the following output is given:

1. Mean response times between cargo ordering and delivery
2. Cargo and vehicle throughput
3. Resource, manpower, and facility utilization
4. Vehicle waiting times, service times, and idle times for each vehicle type node.

All the limitations of the model are directly related to the size of the computer. In order to adequately use the model, one to two man-weeks are required to acquire the data base, after which another man-week is needed to format the data. After the computer run is made, which may take between 10 minutes to 60 minutes of CPU time, one hour to two days are necessary to analyze and evaluate the results. (33:627)

L. Transportation Model (TRANSMO)

The TRANSMO Model was developed and is used by the US Army Concepts Analysis Agency. The purpose of the model "is to determine the arrival time of US forces in overseas theaters of operation." (33:769) This is accomplished by scheduling the deployment of forces with specified transportation vehicles. The input is divided into three types of characteristics: force, lift vehicle, and general. The force characteristics include troop strengths, weights, origin, destination, resupply, consumption and cargo type. The lift characteristics are the speed, load and unload times, and capacity for each type of cargo. Finally, the general characteristics involved are port restrictions, distances between ports, and attrition factors. The output of the model provides a detailed summary of the deployment schedule. To utilize TRANSMO, one week is needed to acquire the data base and two man-weeks to transform the data into the proper input format. Then after about 15 minutes of CPU time, one man-week is necessary to analyze and evaluate the output.

Summary

In this chapter the different strategic deployment models have been reviewed. Though many models exist concerning the deployment of forces, none meet the objective of this research project. This is so for many reasons. First, the models are response oriented, as the specific units and the order of their deployment is determined previously. Second, the models do not address the problem of transshipping the supplies to the combat units forward of the APOD.(5:22). Third, the port capability of the APOD in receiving the intertheater shipments is not examined.

Finally, none of the models use goal programming to optimize the amount of delivered combat power. Therefore, none of the existing deployment models adequately addresses the objective of this project.

III. Model Development

In this chapter the mathematical expressions of the goals and constraints used in the model will be developed. These goals and constraints are taken from the thesis of Capt. Cooke and are adjusted to the form used in the computer program. Any differences between the two will be discussed in full detail. Prior to developing the goals and constraints, the variables, multipliers and subscripts will be defined. Next, the goals followed by the constraints will be derived. Following each constraint will be the number of that type constraint necessary for the model. Finally, a summary which will contain all the goals and constraints in their mathematical expression will be given.

Variable Parameters

The definition of the parameters will be discussed prior to the presentation of the model that was developed by Capt. Cooke.

(4:46-76) The variables are defined as follows:

Variables used:

$X(i,j,k,l)$: the number of delivered aircraft loads.

[AC/PERIOD]

$P(k,l)$: slack and surplus variables used in the equality constraints.

$U(y,m,l)$: the number of units deployed. [UNITS/PERIOD]

S_l : the amount of supplies transshipped in period l from the APOD to the front. [TONS/PERIOD]

Subscripts used:

a : the APOD.

- i: a specific type of aircraft, range is from 1 to I. The aircraft are identified by the user.
- j: a specific type of mission, range is from 1 to J. In this model the missions are divided into four areas as follows: 1 = direct delivery, 2 = APOD delivery, 3 = intratheater delivery, and 4 = airborne delivery.
- k: a specific type of cargo, range is from 1 to K. Here, the cargo is divided into five categories as follows: 1=outsized, 2=oversize, 3=bulk, 4=passengers only, and 5=supplies (considered to be bulk size cargo).
- l: a specific period, range is from 1 to L.
- m: the mode of delivery, range is from 1 to M. Here, the mode is divided into three methods as follows: 1=delivery to the front, 2=delivery via the APOD and remain at the APOD or move to the front on their own, and 3=delivery to the APOD and move to the front via intratheater airlift.
- y: a specific type of unit, range is from 1 to Y. The units are identified by the user. The first Y' are all those units deployed at the front.

Multipliers used:

- AB(y): the airborne capability indicator for unit type y. The value is either 0 or 1.
- AT(y): the anti-tank capability of type y unit expressed in equivalent TOEs.
- AVAIL(i): the availability of type i aircraft.
- CI(y): the Combat Indicator: 1 if a combat unit, -1 if a combat support unit, and 0 if neither a combat or combat

support unit.

CPI(1): the combat power index for arriving at the front in period 1.

DAPDFT: the distance between the APOD and the front. [KM]

DUSAPD: the distance between the US and the APOD. [KM]

DUSFRT: the distance between the US and the front. [KM]

EAS(i,k): the standardizing factor for the MHE required to unload aircraft type i with type k cargo.

FLT(y): the front line trace capability of type y unit. [KM]

FP(y): the firepower capability of type y unit.

GAT(1): the desired goal for anti-tank power by period 1.
[TOE]

GFLT(1): the desired goal for front line trace by period 1.
[KM]

GFP(1): the desired goal for combat power by period 1.

GT(i): the ground time for type i aircraft to offload, upload, and/or refuel. [HOURS/AC]

Ha: the material handling equipment capacity in pallet equivalents existing at the APOD prior to the deployment. [PALLETS/DAY]

Hf: the same as Ha, but for the front. [PALLETS/DAY]

INTER(i): the number of days required between intertheater missions. [DAYS]

INTRA(i): the number of days required between intratheater missions. [DAYS]

KARG(i,k): the cargo capacity of type i aircraft with type k cargo. [TONS/AC]

MHE(i): the pallet capacity of type i aircraft. [PALLETS]
 My: the movement time required by a unit to go from the APOD
 to the front by ground transportation, rounded to the
 nearest integer. [PERIODS]
 L: the number of periods considered in the model.
 NPAL: the number of Pallet Equivalents (PE) that an ALCE
 unit can manage in one day. [PALLETS/DAY]
 NPRK(i): the number of type i aircraft that can park on the
 ramp of the APOD. [AC]
 NPK(y): the number of type y unit fighter aircraft that can
 park on the ramp of the APOD. [AC]
 NTAC(y): the number of fighter aircraft assigned to type y
 unit. [AC/UNIT]
 NUM(i): the number of type i aircraft in the inventory. [AC]
 PL: the number of days in one period. [DAYS/PERIOD]
 RC: the number of pallets the Army Riggers can prepare per
 day. [PALLETS/DAY]
 SPD(i): the no-wind cruise speed of type i aircraft. [KM/HR]
 TON(y,k): the amount of tonnage to move for type y unit of
 type k cargo. [TONS/UNIT]
 TON(y,5): the supply consumption rate of type y unit.
 [TONS/UNIT/DAY]
 TRP(y): the ton-mileage capability for the transportation of
 supplies to the front for type y unit.
 [TON*MILES/UNIT/DAY]
 TVL(y): the amount of distance that type y unit can move in
 one day. [KM/DAY]
 UNIT(y): the number of type y units available. [UNITS]

UTE(i,j): the utilization rate for type i aircraft on a type j mission. [HRS/DAY]

Goals

All combat attributes are not "mutually reinforcing", as some attributes are very effective in one area, but ineffective in other areas. If the attributes were "mutually reinforcing", then the objective could be met by simply deploying as much of the combat attributes as possible without regard to the type of units deployed. This is not the case, therefore decisions need to be made to balance the different units deployed. An aircraft squadron is a good example. Here, the aircraft are very effective at interdiction, but ineffective at maintaining terrain. The objective of this program is to meet the "requirements or specified goals" of the decision maker, and then to "maximize specific force attributes." Goal programming accomplishes this objective effectively, and is therefore utilized to meet the goals and force attributes.

The goals chosen to represent combat power are anti-tank (AT) capability, defensive frontage (front line trace or FLT), and firepower (FP). The first two goals do not depend upon time, while the last goal does depend upon time. The time-dependent variable implements a concept described in the 1981 Congressionally Mandated Mobility Study. (27:30) This concept proves that a small force delivered in the right place can be up to six times more effective than the same size force delivered at a later time. The time-independent goals will be discussed first, followed by the time-dependent goal. (5:50-6)

Included in the time-independent goals are the anti-tank (AT) and defensive frontage (FLT) goals. Since the anti-tank (AT) goals are independent of time, the amount of combat power each unit contributes is related only to the type of unit and the number of units deployed. Therefore, given the multiplier $AT(y)$, which represents the amount of anti-tank potential a specific unit exhibits, the following expression represents the total anti-tank potential at time period l' , assuming no attrition of units, for a given mode of transportation:

$$\sum_{y=1}^{Y'} \sum_{m=1}^3 \sum_{l=1}^{l'} AT(y) * U(y,m,l)$$

The expression would be accurate if the units are delivered to the front and are available to act in combat upon arrival. Not all deliveries are made to the front by direct delivery, though. For units that arrive at the APOD, there are two means by which they can be delivered to the front. The first way is through self-delivery ($m=2$). That is, each unit has the capability to move itself by means that are organic to the unit. The second way for a unit to arrive to the front is through intratheater airlift ($m=3$). Each of these means will be discussed separately.

Self-delivery to the front ($m=2$) is the ability of a unit to move itself from one point to another point without the assistance of another unit. When considering this mode of travel, the time-distance relationship must be considered before the unit can be considered "closed to the combat area." This concept, "closed to the combat area," means that the unit is available to the ground commander to engage in action as needed. Therefore, all units delivered to the APOD are not considered as

contributing to the anti-tank potential until the unit has "closed". To consider this situation, "My" is defined as the closest integer of the distance from the APOD to the front (DAPDFT) divided by the distance the particular unit can travel in one day (TVL(y)) times the period length (PL) considered.

$$My = \text{NINT} [DAPDFT / (TVL(y) * PL)]$$

$$[dist]/([dist/day]*[days/period])=periods$$

The function NINT calculates the closest integer of the expression. Therefore, concerning self-delivery to the front from the APOD, the left-hand side (LHS) of the constraint is as follows:

$$\sum_{y=1}^{Y'-1'-My} \sum_{l=1}^{L'} AT(y) * U(y,2,l)$$

Intratheater airlift (m=3) is the airlift within a particular theater of operations. The time-distance relationship is not considered because the flying time from the APOD to the front is assumed to be short with respect to the length of time per period. Therefore, given that troops and equipment are available at the APOD, the following expression gives the LHS of the constraint:

$$\sum_{y=1}^{Y'-1'} \sum_{l=1}^{L'} AT(y) * U(y,3,l)$$

Finally, some units at the APOD may possess anti-tank potential and must be considered. A fighter squadron at the APOD is an example of such a unit. Units assigned to the APOD are assumed to be shipped only to the APOD. That is, a unit destined for the APOD would not be shipped to the front either by direct

on airborne delivery and then require transportation to the APOD. As a result of this, the time distance relationship between the APOD and the front is not considered. Therefore, the following expression considers units at the APOD that contribute to the total anti-tank power deployed:

$$\sum_{y=Y'+1}^Y \sum_{l=1}^{1'} AT(y) * U(y,2,l)$$

When all of these left hand sides are summed, the result is the total amount of anti-tank power deployed by time period $1'$. Therefore, given the fact that $GAT(1')$ is the amount of anti-tank power desired by period $1'$, the following is the mathematical expression of the goal:

$$\begin{aligned} \sum_{y=1}^{Y'} \sum_{l=1}^{1'} \sum_{m=1,3} AT(y) * U(y,m,l) &+ \sum_{y=1}^{Y'} \sum_{l=1}^{1'-M_y} AT(y) * U(y,2,l) \\ &+ \sum_{y=Y'+1}^Y \sum_{l=1}^{1'} AT(y) * U(y,2,l) \geq GAT(1') \end{aligned}$$

In the above expression, direct delivery ($m=1$) and intratheater delivery ($m=3$) are expressed in the first term. The second summation includes front units delivered to the APOD and able to move to the front independently. Finally, the third summation includes the units assigned to the APOD. (5:53)

The defensive frontage (FLT) goal is developed in the same fashion as the anti-tank (AT) goal. The multiplier $FTL(y)$ is used to represent the amount of defensive frontage a specific unit contributes to the goal. Therefore, the expression

representing the desired amount of defensive frontage (FLT) by period l' is as follows:

$$\sum_{y=1}^{Y'} \sum_{l=1}^{l'} \sum_{m=1,3} \text{FLT}(y) * U(y,m,l) + \sum_{y=1}^{Y'} \sum_{l=1}^{l'-My} \text{FLT}(y) * U(y,2,l) + \sum_{y=Y'+1}^Y \sum_{l=1}^{l'} \text{FLT}(y) * U(y,2,l) \geq \text{GFLT}(l')$$

The third goal to be discussed is firepower. The only difference between this goal and the previously mentioned goals is that the firepower goal is time-dependent. This means that the worth of the same amount of firepower will change over time. If, though, the time-dependent portion of the goal is represented by a multiplier, then firepower would be similar to the other goals with the exception of the time-dependent multiplier. This time-dependent multiplier is defined as the combat power index $\text{CPI}(l)$, and is dependent upon the period that the unit is "closed on the combat arena." The term $\text{FP}(y)$ is the amount of firepower a specific unit contributes to the overall goal. Therefore, the expression for the desired amount of firepower by period l' is as follows:

$$\sum_{y=1}^{Y'} \sum_{l=1}^{l'} \sum_{m=1,3} \text{CPI}(l) * \text{FP}(y) * U(y,m,l) + \sum_{y=1}^{Y'} \sum_{l=1}^{l'-My} \text{CPI}(l+My) * \text{FP}(y) * U(y,2,l) + \sum_{y=Y'+1}^Y \sum_{l=1}^{l'} \text{CPI}(l) * \text{FP}(y) * U(y,2,l) \geq \text{GFP}(l')$$

Cooke does not adjust the time-dependent index to reflect the transportation delay from the APDD to the front. An adjustment is made here though because the index should reflect when the

unit is available for combat, not when it arrives at the APOD.
(5:55)

The three types of goals used to measure combat power are anti-tank, defensive frontage, and firepower capability. The first two goals are not dependent upon time, while the third goal is time-dependent. All three goals were derived and are similar with the exception of the firepower goal. This goal was modified to include an additional multiplier to represent the time-dependent portion of the goal. Next, the factors which constrain the attainment of these goals will be examined in detail.

Bounding Constraints

The different sets of constraints that limit the attainment of the different goals will be examined. Some of these constraints may later be discussed as goals, but will presently be classified as constraints. These constraints involve both aircraft and units separately and jointly. A complete description and development is examined independently for each of the following constraints.

Aircraft Limitations: The limitation of aircraft is divided into two sets of constraints, aircraft availability and average utilization rates. The first constraint set will deal with the number of missions possible from the number of aircraft in the Air Force inventory and their respective availability to participate in the deployment. Then the second constraint set gives the number of missions anticipated given the expected number of hours per day the aircraft will be available to fly missions.

1. The first constraint set will reflect the number of aircraft sorties dependent upon the number of aircraft available and their availability rates. An important factor here is the portion of a day that is required between like missions. Because most of the intertheater missions would require more than one day, but less than two days for a round trip, there is some overflow from the previous day. Therefore, the following expression satisfactorily explains the number of sorties available for a deployment:

$$\begin{aligned} \text{Inter}(i) * \sum_{j=1}^{2,4} \sum_{k=1}^K X(i,j,k,1) \\ + \text{Intra}(i) * \sum_{k=1}^K X(i,3,k,1) \leq \text{NUM}(i) * \text{AVAIL}(i) * \text{PL} \end{aligned}$$

There is no overflow considered between periods. Cooke considers the overflow between periods in his model because his period is equal to one day. The period length in DEPLOY is significantly larger than the Inter(i) term so realism is maintained by not considering overflow between periods. There is one constraint per aircraft type per period, or a total of (I*L).

2. This set of constraints considers the number of aircraft sorties in relationship to the expected utilization (UTE) rate. The UTE rate is "commonly used to define airlift capability" (12:7), and is the "fleetwide average flying hours per day per aircraft." The average number of sorties the fleet of a particular type of aircraft can be expected to fly in a period is as follows:

$$\sum_{i=1}^I \sum_{k=1}^K \text{KARG}(i,5) * X(i,4,5,1) - P(5,1-1) + P(5,1) = RC * PL$$

There is one constraint per period, or L constraints.

Summary

In this chapter, the mathematical expressions for the goals and constraints that are used in the model have been derived. The majority of the expressions were taken from the thesis of Capt Cooke with only a few minor changes. In addition, the variables, subscripts, multipliers and other terms used in the model were defined. Finally, a summary of the constraints is given with the total number of constraints that are in the model:

Goals-Time Independent

Anti-tank

$$\sum_{y=1}^{Y'} \sum_{l=1}^{L'} \sum_{m=1,3} \text{AT}(y) * U(y,m,1) + \sum_{y=1}^{Y'} \sum_{l=1}^{L'-My} \text{AT}(y) * U(y,2,1) + \sum_{y=Y'+1}^Y \sum_{l=1}^{L'} \text{AT}(y) * U(y,2,1) \geq \text{GAT}(1')$$

One for each goal.

Front Line Trace

$$\sum_{y=1}^{Y'} \sum_{l=1}^{L'} \sum_{m=1,3} \text{FLT}(y) * U(y,m,1) + \sum_{y=1}^{Y'} \sum_{l=1}^{L'-My} \text{FLT}(y) * U(y,2,1) + \sum_{y=Y'+1}^Y \sum_{l=1}^{L'} \text{FLT}(y) * U(y,2,1) \geq \text{GFLT}(1')$$

One for each goal.

support unit. This term adds one constraint to the problem.
(5:71)

2. A common example of a unit linkage floor is "the relationship between combat units and combat service support units, such as headquarters [HQ] units." One way to express at least a 1 to 5 ratio of HQ's units to battalion units is:

$$cmbt - 5HQ \leq 0.$$

A problem with this is that, in the initial deployment, a HQ unit must precede the combat battalion. This is clearly not desired in the model or reality. A better expression is,

$$cmbt - 3HQ \leq 2.$$

Here two combat units could deploy before a HQ unit is required. Furthermore, one HQ unit could handle a maximum of five battalions, and two HQ units could maintain up to eight battalions. This expression conforms more accurately to the present doctrine and practice. As a result, this expression "would add one constraint for each linkage between combat units and other type of units." (5:72)

Unit Abilities: Unit abilities is the last set of constraints to be examined. Here the airdrop capability is limited by the personnel responsible for the preparation of loads for the airdrop. Mathematically this is expressed as,

$$\sum_{i=1}^I \sum_{k=1}^K KARG(i,5) * X(i,4,k,1) \leq RC * PL$$

As in the case of the shipment of units, the previous period's surplus, $P(S,1-1)$, and the present period's surplus, $P(S,1)$, change the constraint to be an equality.

as in the last constraint, only the first Y' units are deployed at the front while the remainder $(Y'+1, \dots, Y)$ are deployed at the APDD. There is one constraint per period or a total of L constraints. (5:70)

Unit Linkage: This section describes certain ceilings and floors of the particular units deployed in relation to other units deployed. When considering its ceiling, more of a particular unit cannot be deployed without a corresponding increase in another type of unit. Conversely, a floor requires the deployment of a particular unit in order for another unit to increase the number that are deployed. Both of these terms keep certain units in a particular predetermined ratio. First a specific case of a ceiling will be discussed, followed by an example of a floor.

1. An example of a ceiling is "the deployment of combat support arms such as artillery, requiring a minimum number of combat units to be deployed before more follow-on artillery" are deployed. In order to keep the ratio between combat designated units and combat support units in at least a one-to-one ratio, the following expression is used:

$$\sum_{y=1}^Y \sum_{m=1}^3 \sum_{l=1}^L CI(y) * U(y,m,l) \geq 0.$$

Here, the multipliers $CI(y)$ are combat indicator variables to distinguish between combat, combat support, and other units, and are given the values 1, -1, and 0 respectively. The first two types are self explanatory. An example of an "other" unit would be the ALCE as it is considered neither a combat nor a combat

$$\sum_{i=1}^I \sum_{j=1,4} KARG(i,5) * X(i,j,5,1') + S1 + P(5,1'-1)$$

$$- \sum_{y=1}^{Y'} \sum_{m=1}^3 \sum_{l=1}^{1'-1} PL * TON(y,5) * U(y,m,1) - P(5,1) = 0.$$

and

$$\begin{aligned} & \sum_{i=1}^I KARG(i,5) * X(i,3,5,1') \\ & + \sum_{y=1}^Y \sum_{m=1}^3 \sum_{l=1}^{1'-My} [TRP(y) * PL1/[2 * DAPDFT] * U(y,m,1)] \geq S1 \end{aligned}$$

There are two constraints per period or a total of $2*L$ constraints. (5:68-9)

3. The constraint of supplies for the APOD will be discussed in this section. As in all the other cases, supply must equal or exceed the requirements of the deployed units. Supplies that come into the APOD are either consumed by local deployed units or shipped to the front by deployed truck units or intratheater aircraft missions. The supply constraint, then, is the sum of the supplies delivered to the APOD, plus the surplus supplies from the previous period, minus that which is consumed at the APOD, minus supplies transshipped to the front, and minus the excess supply for the period. The result is as follows:

$$\begin{aligned} & \sum_{i=1}^I KARG(i,5) * X(i,2,5,1') + P(5,1'-1) - S1 - P(5,1') \\ & - \sum_{y=Y'+1}^Y \sum_{l=1}^{1'-1} PL * TON(y,5) * U(y,2,1) = 0. \end{aligned}$$

The $S1$ term relates to the previous supply constraint at the front and is the amount that is sent forward to the front. Also,

$$\begin{aligned}
& \sum_{i=1}^I \sum_{j=1,3}^4 \text{KARG}(i,5) * X(i,j,5,1') \\
& + \sum_{y=1}^Y \sum_{m=1}^3 \sum_{l=1}^{l'-My} [\text{TRP}(y) * \text{PL}]/[2 * \text{DAPDFT}] * U(y,m,l) \\
& - \sum_{y=1}^{Y'} \sum_{m=1}^3 \sum_{l=1}^{l'-1} \text{PL} * \text{TON}(y,5) * U(y,m,l) \geq 0.
\end{aligned}$$

Another constraint must be added to the above expression to ensure that aircraft and trucks do not ship supplies that are not available at the APOD. By letting S1 equal the supplies transshipped during period 1, the parameter is limited to the capacity of the transshipment pipeline. This is similar to stating that the actual amount supplied cannot exceed the capability of moving supplies to the front from the APOD. Here, the variable TRP(y) functions as an indicator variable, in that if the unit considered is not a transportation unit, the value is zero. The expression for period 1' is then as follows:

$$\begin{aligned}
& \sum_{i=1}^I \text{KARG}(i,5) * X(i,3,5,1') \\
& + \sum_{y=1}^Y \sum_{m=1}^3 \sum_{l=1}^{l'-My} [\text{TRP}(y) * \text{PL}]/[2 * \text{DAPDFT}] * U(y,m,l) \geq S1
\end{aligned}$$

Furthermore, the value of S1 must be less than or equal to the amount of supplies left over at the APOD. This additional constraint will be discussed further in the constraints of supplies for the APOD. Therefore, the final constraints when considering slack and surplus variables are as follows:

deliveries (j=1), intratheater missions (j=3), and airborne deliveries (j=4). Mathematically, this is expressed as follows:

$$\sum_{i=1}^I \sum_{j=1,3}^4 KARG(i,5) * X(i,j,5,1)$$

When considering the transportation of supplies via truck, the time-distance relationship must be considered along with the unit's capability of moving supplies. Each unit has an input parameter of the number of ton-miles the unit can move per day [TRP(y)]. A combat unit though is normally given a parameter equal to zero. This means that the combat unit can only move supplies from its rear area to its front. The following expression determines the number of tons a unit can move in one period.

$$[TRP(y) * PL] / [2 * DAPDFT]$$

$$[\text{ton-miles/day}] * [\text{days/period}] / [\text{miles}] = [\text{tons/period}]$$

The two in the denominator reflects the round trip distance from the APOD to the front. Therefore, the total tonnage shipped via truck is as follows:

$$\sum_{y=1}^Y \sum_{m=1}^3 \sum_{l=1}^{I'-M_y} [TRP(y) * PL] / [2 * DAPDFT] * U(y,m,1)$$

By combining the expressions for the amount of supplies brought to the front and subtracting the consumption of the supplies by the units deployed to the front, the resulting expression is formed:

for the remainder of the period. Therefore, given the parameter $TON(y,5)$ as the daily consumption rate of a specific unit, the supply constraint for period l' is:

$$\sum_{i=1}^I \sum_{j=1}^{2,4} KARG(i,5) * X(i,j,5,l') + P(5,l'-1) - \sum_{y=1}^Y \sum_{m=1}^3 \sum_{l=1}^{l'-1} TON(y,5) * PL * U(y,m,l) - P(5,l') = 0.$$

Here $P(5,l')$ represents the surplus supplies shipped in the l' th period, while $P(5,l'-1)$ is the surplus carried forward from the previous period. Again, it is assumed here that each unit upon its deployment has sufficient supplies to sustain itself for the remainder of the period. There is one constraint for each period or L constraints. (5:67)

2. Here, the constraints to ensure adequate supplies reach the front are discussed. Every unit not based at the APOD is considered to consume its supplies at the front. In other words, all the units with the exception of the APOD-ALCE, the fighter squadrons and the transportation units consume their supplies at the front. In the file that contains the information on the different units, the first Y' units are deployed to the front, while the others are deployed to the APOD. Delivery to the front can take place by one of four means;

- a. direct delivery,
- b. airborne delivery,
- c. transportation via intratheater airlift, and
- d. transshipment via truck.

The last means of delivery, by truck, will be discussed separately. The expression of the total amount of supplies delivered by airlift is simply the sum of the number of direct

if the unit is not airborne capable. The expression then becomes,

$$\sum_{i=1}^I KARG(i,k) * X(i,4,k,1) - \sum_{y=1}^{Y'} AB(y) * TON(y,k) * U(y,1,1) + F(k,1-1) - F(k,1) = 0.$$

Once again, there is one constraint for each period and cargo type, or $L * K$ ($K=4$) constraints. (5:66)

Shipment of Supplies: Supplies are included in a separate classification ($k=5$), and are used to describe the aggregate needs to sustain a unit. Therefore, once units are delivered to the theater, there is a requirement to sustain the unit with a certain tonnage of supplies. Therefore, the input must equal or exceed the consumption of supplies by the deployed units. Three different sections will be discussed concerning this issue. The first section will insure that there are adequate supplies delivered to the theater to meet the needs of all the different units in the theater. Then, the second section will insure that the necessary supplies reach the units deployed to the front. Finally, the last section will insure adequate supplies are available for the units based at the AFOD. Each of these sections will be examined in turn.

1. The requirement for intertheater supplies is very similar to the shipment of units previously mentioned. There is one difference though; all the units that have "closed" in the previous periods are those receiving the supplies. The supplies are delivered only for the units deployed in the previous periods because it is assumed that each deployed unit has enough supplies

$$\sum_{i=1}^I \text{KARG}(i,k) * X(i,2,k,1) - \sum_{y=1}^Y \sum_{m=2}^3 \text{TON}(y,k) * U(y,m,1) \geq 0$$

However, because excess cargo [$P(k,1)$] or early cargo [$P(k,1-1)$] must also be considered, these variables should also be included in the constraint.

$$\sum_{i=1}^I \text{KARG}(i,k) * X(i,2,k,1) - \sum_{y=1}^Y \sum_{m=2}^3 \text{TON}(y,k) * U(y,m,1) + P(k,1-1) - P(k,1) = 0.$$

Finally, to ensure that cargo moved to the APOD reaches the front, an additional constraint is added:

$$\sum_{i=1}^I \text{KARG}(i,k) * X(i,3,k,1) - \sum_{y=1}^Y \text{TON}(y,k) * U(y,3,1) \geq 0$$

There are two constraints for each period and cargo size, or $L*K$ ($K=4$) constraints. (5:65-6)

2. The shipment of units to the front via direct delivery is very similar to the shipment of units to the APOD. Therefore, the expression to indicate the closure of units to the front by direct delivery is:

$$\sum_{i=1}^I \text{KARG}(i,k) * X(i,1,k,1) - \sum_{y=1}^{Y'} [1-AB(y)] * \text{TON}(y,k) * U(y,1,1) + P(k,1-1) - P(k,1) = 0.$$

As before, there is one constraint for each period and cargo type, or $L*K$ ($K=4$) constraints. (5:66)

3. The shipment of units to the front via airdrop is also similar to the previous constraint with one minor difference. To insure that only airborne capable units are delivered, an indicator variable $AB(y)$ is necessary. The value of $AB(y)$ is equal to one if the unit is airborne capable, or is equal to zero

arriving first to APOD and then transporting themselves to the front. If the ALCE units were unable to move themselves, the third summation would not exist. There is one constraint per period for both the APOD and the front, or $2*L$ constraints. (5:63-4)

Unit Limitation: The number of units shipped is obviously limited by the number of units available. Therefore, this constraint is simply expressed as the following:

$$\sum_{m=1}^3 \sum_{l=1}^L U(y,m,l) \leq \text{UNIT}(y).$$

There is one constraint for each type of unit, or a total of Y constraints. (5:64)

Shipment of Units: This section will discuss the three different ways that units can be shipped into the theater. These three ways are via the APOD, airborne delivery, and direct delivery.

1. The delivery completion time for the units to the APOD is examined in this section. When a unit is transported in its entirety, the unit is said to be "closed." All fighter aircraft squadrons, and transportation units are assumed to be delivered and assigned to the APOD. ALCE units are unique in that they are assigned to either the front or the APOD. Each unit is described in terms of personnel and outsize, oversize and bulk tonnage. Therefore, the unit cannot be "closed" until all the personnel and tonnage of each type of cargo for the unit is delivered. The mathematical expression of this is as follows:

pallet equivalents that an ALCE unit can download in one day. Since the period here spans many days, an arbitrary term of less than one is multiplied by the number of newly arriving ALCE units. The multiplier in this case is one-half. This will more closely approximate the additional assistance gained from the newly arriving ALCE units than if the periods were treated similar to a single day. Finally, because the number of aircraft "moving through the APOD must not be greater than the capabilities at the airport to unload them", the following constraints for period $1'$ are established:

For the APOD:

$$\sum_{i=1}^I \sum_{j=2,3} \sum_{k=1}^K \text{EAS}(i,k) * \text{MHE}(i) * X(i,j,k,1')$$

$$- \sum_{l=1}^{1'-1} \text{NPAL} * \text{PL} * \text{U}(\text{ALCE},2,1)$$

$$- .5 * \text{NPAL} * \text{PL} * \text{U}(\text{ALCE},2,1') \leq \text{Ha} * \text{PL}$$

For the front:

$$\sum_{i=1}^I \sum_{j=1,3} \sum_{k=1}^K \text{EAS}(i,k) * \text{MHE}(i) * X(i,j,k,1')$$

$$- \sum_{l=1}^{1'-1} \sum_{m=1,3} \text{NPAL} * \text{PL} * \text{U}(\text{ALCE-FRONT},m,1)$$

$$- \sum_{l=1}^{1'-Mx} \text{NPAL} * \text{PL} * \text{U}(\text{ALCE-FRONT},2,1)$$

$$- \sum_{m=1,3} .5 * \text{NPAL} * \text{PL} * \text{U}(\text{ALCE-FRONT},m,1') \leq \text{Hf} * \text{PL}$$

Concerning the constraints at the front, the second summation considers the ALCE units arriving either by direct or intratheater delivery. The third summation, considers ALCE units

A special situation to consider here is the case of fighters assigned to the APOD. If this is the circumstance, the fighters must be assured available ramp space at all times. Therefore, by combining this constraint to the previous constraint, the result is as follows:

$$\sum_{i=1}^I \sum_{j=2}^3 \sum_{k=1}^K X(i,j,k,1) * GT(i) / [PL * NPRK(i) * 24] \\ + \sum_{y=Y'+1}^Y U(y,2,1) * NTAC(y) / [PL * NPK(y)] \leq 1.$$

[units/period]*[AC/unit]*[24hrs]/([days/period]*[AC]*[24hrs/day])

The 24 in the numerator and the denominator cancel out, but the units are important to show the relationships between the variables. There is one constraint for each period considered, or a total of L constraints. (5:62-3)

2. This section reviews the constraints of material handling equipment (MHE) at the APOD and front. The time to off-load an aircraft is dependent upon the number of pallets the aircraft can hold and the type of cargo being off-loaded. The number of pallets an aircraft can hold (NPAL) is easily determined, while the relative ease with which these pallets are offloaded is not. Therefore, the variable EAS(i,k) measures the relative ease with which pallets for type i aircraft with type k cargo can be off-loaded. If all the aircraft and cargo can be removed with the same ease, the variables are all equal to one. The resources available to download the aircraft are the MHE available prior to the deployment (H_a for the APOD and H_f for the front), and the ALCE with its "ancillary port improvement packages" (APAL). The variable APAL measures the number of

considers the amount of ramp parking space available for the aircraft at the APOD. The second constraint of MHE will interface with the capability to unload an aircraft once it has arrived at the APOD or front.

1. This set of constraints considers the limited amount of ramp space available at the APOD. A linear relationship is assumed between the area required for different type of aircraft. Specifically, if the ramp is sufficient to park a total of 48 F-16's or 12 C-130's, the ramp is also sufficient to park 24 F-16's and 6 C-130's, or any convex combination thereof. This approximation is adequate for most "airfields except very small fields with end-point solutions to parking limitations." Here a large aircraft could block off a taxiway or runway from other aircraft. Normally though, this is not a problem because APOD's are normally not "chosen at minimally capable terminals." (5:61)

The parameter which considers ramp space is parking-space required per unit time for each type of aircraft. Here, it is assumed the aircraft is unloaded and serviced within the planned ground time (GT(i)). This parameter is constrained as follows:

$$\sum_{i=1}^I \sum_{j=2}^3 \sum_{k=1}^K X(i,j,k,1) * GT(i) / [PL * NPRK(i) * 24] \leq 1.$$

$$[AC/period]*[hrs]/([days/period]*[AC]*[hrs/day])$$

Here, j is summed only over the missions that transit the APOD. Also, all the units for the terms are cancelled out, leaving a unitless constraint. When the left hand side of the constraint is equal to one, the ramp space is completely saturated.

$$[LPL * UTE(i,j) * NUM(i) * SPD(i)] / [2 * DUSAPD].$$

$$[days/period * hours/day * aircraft * km/hour]/[km]=$$

$$aircraft/period$$

The number "2" in the denominator reflects the aspect of a round trip, the total distance flown. When considering only the aircraft that are involved in the deployment, and not the entire fleet of a particular aircraft, the above expression must be multiplied by the percentage of the aircraft that will be available in the deployment. Cooke does not include this last factor, but should, as without it, he is saying that all the flying time for a particular aircraft type is consumed by the aircraft in the deployment.

$$[LPL * UTE(i,j) * NUM(i) * AVAIL(i) * SPD(i)] / [2 * DUSAPD].$$

$$[days/period * hours/day * aircraft * \% * km/hour]/[km]=$$

$$aircraft/period$$

Therefore, the average number of a specific aircraft that will fly in a given period must be less than or equal to the above expression. By rearranging some of the terms and summing over the missions and cargo, the following constraint is developed:

$$\sum_{j=1}^J \sum_{k=1}^K [1/UTE(i,j)] * X(i,j,k,1')$$

$$\leq [LPL * NUM(i) * AVAIL(i) * SPD(i)] / [2 * DUSAPD]$$

Cooke uses a standardizing factor to convert aircraft of the same type, flying different missions to an equivalent number of sorties for the respective mission. The ratio used is as follows:

$$[Distance other] / [Distance from the US to AFOD]$$

This ratio shows that there are more sorties available for

Fine Power

$$\begin{aligned} & \sum_{y=1}^{Y'} \sum_{l=1}^{l'} \sum_{m=1,3} \text{CPI}(l) * \text{FP}(y) * \text{U}(y,m,l) \\ & + \sum_{y=1}^{Y'} \sum_{l=1}^{l'-My} \text{CPI}(l+My) * \text{FP}(y) * \text{U}(y,2,l) \\ & + \sum_{y=Y'+1}^Y \sum_{l=1}^{l'} \text{CPI}(l) * \text{FP}(y) * \text{U}(y,2,l) \geq \text{GFP}(l') \end{aligned}$$

One for each goal.

Aircraft Sorties

$$\begin{aligned} \text{Inter}(i) & * \sum_{j=1}^{2,4} \sum_{k=1}^K \text{X}(i,j,k,l) \\ + \text{Intra}(i) & * \sum_{k=1}^K \text{X}(i,3,k,l) \leq \text{NUM}(i) * \text{AVAIL}(i) * \text{PL} \end{aligned}$$

One for each aircraft type and period.

Aircraft UTE Rate

$$\begin{aligned} & \sum_{k=1}^K \sum_{j=1,4} [\text{1/UTE}(i,j)] * [\text{DUSFRT/DUSAPD}] * \text{X}(i,j,k,l') \\ & + \sum_{k=1}^K [\text{1/UTE}(i,2)] * \text{X}(i,2,k,l') \\ & + \sum_{k=1}^K [\text{1/UTE}(i,3)] * [\text{DAPDFT/DUSAPD}] * \text{X}(i,3,k,l') \\ & \leq [\text{PL} * \text{NUM}(i) * \text{AVAIL}(i) * \text{SPD}(i)] / [2 * \text{DUSAPD}] \end{aligned}$$

One for each aircraft type and period.

Airport Ramp Space

$$\begin{aligned} & \sum_{i=1}^I \sum_{j=2}^3 \sum_{k=1}^K \text{X}(i,j,k,l) * \text{GT}(i) / [\text{PL} * \text{NPRK}(i) * 24] \\ & + \sum_{y=Y'+1}^Y \text{U}(y,2,l) * \text{NTAC}(y) / [\text{PL} * \text{NPK}(y)] \leq 1. \end{aligned}$$

One for each period.

Airport MHE

For the APD:

$$\sum_{i=1}^I \sum_{j=2,3} \sum_{k=1}^K EAS(i,k) * MHE(i) * X(i,j,k,1')$$

$$- \sum_{l=1}^{1'-1} NPAL * PL * U(ALCE,2,1)$$

$$- .5 * NPAL * PL * U(ALCE,2,1') \leq H_a * PL$$

For the front:

$$\sum_{i=1}^I \sum_{j=1,3} \sum_{k=1}^K EAS(i,k) * MHE(i) * X(i,j,k,1')$$

$$- \sum_{l=1}^{1'-1} \sum_{m=1,3} NPAL * PL * U(ALCE-FRONT,m,1)$$

$$- \sum_{l=1}^{1'-My} NPAL * PL * U(ALCE-FRONT,2,1)$$

$$- \sum_{m=1,3} .5 * NPAL * PL * U(ALCE-FRONT,m,1') \leq H_f * PL$$

One for each period.

Unit Limitation

$$\sum_{m=1}^3 \sum_{l=1}^L U(\gamma,m,l) \leq UNIT(\gamma).$$

One for each deployable unit type.

Shipment of Units to the APDD

$$\sum_{i=1}^I \text{KARG}(i,k) * X(i,2,k,1) - \sum_{y=1}^Y \sum_{m=2}^3 \text{TON}(y,k) * U(y,m,1) + P(k,1-1) - P(k,1) = 0.$$

and

$$\sum_{i=1}^I \text{KARG}(i,k) * X(i,3,k,1) - \sum_{y=1}^Y \text{TON}(y,k) * U(y,3,1) \geq 0$$

Two for each period and cargo size.

Shipment of Units via Direct Delivery

$$\sum_{i=1}^I \text{KARG}(i,k) * X(i,1,k,1) - \sum_{y=1}^{Y'} [1-AB(y)] * \text{TON}(y,k) * U(y,1,1) + P(k,1-1) - P(k,1) = 0.$$

One for each period and cargo size.

Shipment of Units via Airdrop

$$\sum_{i=1}^I \text{KARG}(i,k) * X(i,4,k,1) - \sum_{y=1}^{Y'} AB(y) * \text{TON}(y,k) * U(y,1,1) + P(k,1-1) - P(k,1) = 0.$$

One for each period and cargo size.

Theater Supplies

$$\sum_{i=1}^I \sum_{j=1}^{2,4} \text{KARG}(i,5) * X(i,j,5,1') + P(5,1'-1) - \sum_{y=1}^Y \sum_{m=1}^3 \sum_{l=1}^{1'-1} \text{TON}(y,5) * PL * U(y,m,1) - P(5,1') = 0.$$

One for each period.

Front-Line Supplies

$$\sum_{i=1}^I \sum_{j=1,4} KARG(i,5) * X(i,j,5,1') + S1 + P(5,1'-1)$$

$$- \sum_{y=1}^{Y'} \sum_{m=1}^3 \sum_{l=1}^{1'-1} PL * TON(y,5) * U(y,m,1) - P(5,1) = 0.$$

and

$$\sum_{i=1}^I KARG(i,5) * X(i,3,5,1')$$

$$+ \sum_{y=1}^Y \sum_{m=1}^3 \sum_{l=1}^{1'-My} [TRP(y) * PL] / [2 * DAPDFT] * U(y,m,1) \geq S1$$

One for each period.

APOD Supplies

$$\sum_{i=1}^I KARG(i,5) * X(i,2,5,1') + P(5,1'-1) - S1 - P(5,1')$$

$$- \sum_{y=Y'+1}^Y \sum_{l=1}^{1'-1} PL * TON(y,5) * U(y,2,1) = 0.$$

One for each period.

Link of Combat and Combat Support

$$\sum_{y=1}^Y \sum_{m=1}^3 \sum_{l=1}^L CI(y) * U(y,m,1) \geq 0.$$

One.

Link of Units and HQ's Units

$$cmbt - 3HQ \leq 2.$$

One for each HQ's Unit.

Rigger Capability

$$\sum_{i=1}^I \sum_{k=1}^K \text{KARG}(i,5) * X(i,4,5,1) - P(5,1-1) + P(5,1) = RC * PL$$

One for each period.

The upper limit for the total number of aircraft variables is $I * J * K * L$, while for the unit variables it is $Y * M$. However, because many of the combinations are nonexistent, the number of actual variables necessary is considerably less than the upper limit. The number of equations from the constraints, in order, is # Goals + $2(I*L)$ + $3L$ + Y + $4(L*K)$ + $4L$ + 1 + #HQs + L , or # Goals + # HQs + $2(I*L)$ + $4(L*K)$ + $8L$ + Y + 1 .

IV The General Model

Introduction

DEPLOY is an interactive goal programming model for the rapid deployment of armed forces. The program PAGP (Partitioning Algorithm for Goal Programming) is used for the allocation routine.(3:378-86) This chapter will discuss goal programming (GP) in general along with its relationship to linear programming (LP). As a result, this chapter will serve as an introduction to those not familiar with GP and act as a review to those who are well acquainted with GP. Then in the second section, the PAGP program will be examined as to why it was chosen and the efficiencies associated with it. PAGP was acquired from the Association for Computing Machinery (ACM) and, because it is copyrighted, cannot be used for commercial use without permission.

Goal Programming

Linear Programming (LP) is a very powerful and useful tool, but its "major criticism ... is that only one objective is permitted." (12:353) When only one objective is used, the "goals must be measured on a common scale." (12:358) Often though, decision makers are confronted with more than one objective such as maximizing total profits or minimizing total costs. A few of the many objectives may be improving the workers' morale, paying dividends to the stockholders, and meeting the consumers' needs. All of these objectives would be very difficult to put on a common scale. This is certainly true when considering the rapid deployment of armed forces. Here, the objective is to maximize delivered combat power without jeopardizing the sustainability of

the deployed forces. The problem with using LP is that combat power cannot be expressed on a common scale, but rather is measured in relationship to the amount of anti-tank power, front line trace and firepower a specific unit projects. Goal Programming (GP) can manage multiple objectives and for this reason is used for the general allocation routine.

In discussing GP, the set-up process and the assignment process will be explained, followed by the mathematical expression. Each objective must be formulated with a specified numerical goal in the set-up process. This is similar to the objective function in LP except the desired level or outcome of the objective must be specified. Next, the assignment of values to the decision variables is such that the weighted sum of the deviation variables from the objective function's goal is minimized. (8:172) Mathematically this is expressed as,

$$\min f = \sum_i [P(k) * W(i,k) * d(i)^+ + P(s) * W(i,s) * d(i)^-]$$

st:

$$\sum_{j=1}^n M(i,j) * X(j) + d(i)^- - d(i)^+ = g(i) \quad i=1, \dots, p$$

$$\sum_{j=1}^n A(i,j) * X(j) \leq b(i) \quad i=p+1, \dots, p+m$$

$$X(j), d(i)^-, d(i)^+ \geq 0 \quad j=1, \dots, n; \quad i=1, \dots, p \quad (12:371)$$

The first summation is the weighted sum of deviations from the objective function. Here, the $W(i,k)$ is the assigned weight of the deviation variable $d(i)$ for priority k , while the P 's are the priority factors of the specific priority. By assigning weights to the deviation variables, it is possible to prioritize the

goals within the same level. These variables will be discussed more extensively in the following paragraphs. The second summation is the goal objectives with " $g(i)$ " being the specified level of goal " i ". Finally, the last summation is the normal constraints similar to those found in an LP problem.

The priority factors are also called preemptive-priority factors. This indicates that "any goal at preemptive-priority k (designated $P(k)$) will always be preferred to (i.e., preempt) any goal at a lower priority $k+1, \dots, K$, regardless of any scalar multiplier $W(i,k)^+$ or $W(i,k)^-$ associated with these lower priorities." (4:57) To understand the preemptive aspect, consider, for example, a possible decision process of selecting a house to buy. Suppose the buyer's first priority is that the house be within a 10-mile radius from his place of work. All homes that lie outside the 10-mile radius are not considered. Next, suppose the buyer wants to consider only houses under \$120,000. As a result, even though houses less than \$120,000 may exist outside the 10-mile radius, they are not considered and "are excluded (or preempted) from consideration by the first priority." (4:57) Therefore, "the preemptive-priority concept is used in the decision analysis as an iterative screening process." (9:380-81)

The final factors to be discussed are the deviation variables. These variables are defined as either positive or negative deviation variables. Further, they provide a means to measure both the underachievement and overachievement of a goal. By minimizing either one or both, the goal is defined as either a

ceiling, floor or equality. The following chart specifies which variable to minimize for a specific goal.

GP Form: $f(i) + d(i)^- - d(i)^+ = g(i)$

$f(i) \leq g(i)$: ceiling	minimize $d(i)^+$,
$f(i) = g(i)$: equality	minimize $d(i)^+ \& d(i)^-$, or
$f(i) \geq g(i)$: floor	minimize $d(i)^-$ (9:377)

The deviation variables, therefore, are used to measure the amount that the goals differ from what is desired.

In this section, the difficulty associated with LP was discussed along with how GP compensates by being able to manage multiple objectives. Then, the capability of GP was related to the need of DEPLOY, managing multiple objective functions. In addition, the setup and assignment processes were examined along with the mathematical expression of GP. Finally, the functions of the priority factors and deviation variables were discussed.

PAGE

The PAGP routine was chosen among other routines for several reasons, including its ability to efficiently allocate resources. First, the program had proved itself as an efficient allocation routine in a nuclear exchange model (BRIK) by Capts. Bunnell and Tackacs. (4:66,69) PAGP could also be employed as a subroutine in DEPLOY and therefore increase the portability of the model by not being bound to a necessary library routine. In addition to this, PAGP has been shown to be more efficient than one of the other popular GP packages done by Lee. (3:381-84) The key to PAGP's efficiencies is that the goal constraints are partitioned, some variables are eliminated, and a special termination rule is employed. Each of these efficiencies will be discussed in the following paragraphs.

The partitioning of goal constraints is the first efficiency of PAGP to be examined. This partitioning "is accomplished by observing that, for any goal constraint i , one and only one of the three things may occur.

- (1) only $d(i)^-$ appears in the objective function,
- (2) only $d(i)^+$ appears in the objective function,
- (3) both $d(i)^-$ and $d(i)^+$ appear in the objective function." (3:379)

In case (1) the partition would assign goal constraint i to the priority factor associated with $d(i)^-$; in case (2) constraint i would be assigned to the priority factor associated with $d(i)^+$; while in case (3) the partition would determine the higher order priority factor (in terms of the ordinal ranking) associated with either $d(i)^-$ or $d(i)^+$ and constraint i would be assigned to that priority". (3:379) When real constraints are involved, they are considered before any of the goals and are treated first. Then when the real constraints have been satisfied, the constraints that have deviation variables in priority one are considered. New priorities continue to enter one by one until either all the priorities are considered or the special stopping rules are initiated. By partitioning the goal constraints, the problem is divided into subproblems which are solved in turn, resulting in improved efficiency.

The next efficiency of PAGP to be discussed is the elimination procedure. "The motivation behind the elimination procedure comes from the theory of LP." When a standard LP problem is optimized, " $x(j)$ cannot enter the basis to form an alternate optimal solution" where $z(j) - c(j) > 0$. A corollary to this theorem is "in the standard GP problem, if the optimal

tableau for subproblem $S(k)$ has been found, then any nonbasic variable $t(s)$ (where $t(s)$ can be a decision variable or a deviation variable) which has at least one positive relative cost [$z(j)-c(j) > 0$] can be eliminated from entering the basis in subproblems $S(k+1), \dots, S(p)$." As a result of this procedure, fewer nonbasic variables are considered for entry into the basis, making the algorithm more efficient. (3:380)

The final efficiency to be investigated is the special stopping rule. When the program determines at a subproblem that there are no nonbasic variables to enter the basis, the stopping rule is invoked. This rule is initiated regardless of the number of lower priorities remaining to be considered. Therefore, when there is no possibility of improving, the problem terminates. This, like the other aspects of PAGP discussed, causes this program to be more efficient.

Conclusion

This chapter has described the general model of DEPLOY as a GP model that incorporates the program PAGP for the allocation routine. The basics of GP were reviewed along with its advantage over normal LP and how it satisfies the requirement of multiple objectives generated from DEPLOY. Furthermore, the efficiencies of PAGP were investigated as to why the routine was chosen to support DEPLOY. This concludes the examination of the general model, and in the next chapter the model characteristics of DEPLOY will be discussed.

V. Model Characteristics

This chapter examines two characteristics of DEPLOY. The first section discusses the model restrictions along with the scenario limitations concerning the application and usefulness of the model. In the second section, the input parameters necessary for DEPLOY will be reviewed.

Limitations

E. S. Quade is quoted as saying, "All of the assumptions of a model must be made explicit ... so his (the modeler's) errors will be more evident".(11:219) Though assumptions do not "necessarily reflect errors, they do represent the modeler's perception of the real world".(4:26) As a result, it is paramount that before any model is used, the assumptions and limitations be known to the user. Therefore, the limitations of DEPLOY are plainly stated, not to reveal error, but rather to give the user necessary information. This section is divided into two areas, the first area reviews the model restrictions while the second area discusses the scenario limitations.

Model Restrictions: DEPLOY's application and usefulness are restricted by the following factors:

1. Constant availability of Aircraft. The availability of aircraft is constant throughout the time periods considered in the model. That is, an aircraft cannot become more or less available from one period to the next. The only exception to this is when considering the first period the aircraft is available. Before the first period of availability, the aircraft is not considered as a possible resource to deploy units. At the first period of availability, the number of aircraft considered

as resources is a fixed constant ratio of the number of aircraft in the inventory. This fixed constant ratio is user-defined.

2. No individual scheduling between aircraft and deployed units. Aircraft in the model move cargo identified by the size and period in which it is shipped. There is no attempt to identify particular cargo with a specific deployed unit. As a result, an aircraft may be transporting several units at once, or just one unit. There is no distinction made between these two situations. This is employed in an attempt to minimize the number of different variables in the model. The process is valid because in order for a unit to "close", a specific number of passengers and different sizes of cargo must be transported. The number of each is determined by the unit "closing", therefore there is no need to specify the type of unit with a particular aircraft.

3. Non-integer solutions. The number of resources utilized and the number of units deployed are not restricted to integers. Therefore, the solutions will most likely include non-integer solutions.

4. No synergistic effect between units. The measure of effectiveness of all deployed units is independent of the number and type of other units deployed. That is, two mechanized battalions have the same effectiveness whether there are ten or no artillery battalions deployed.

5. Linear relationship of effectiveness of deployed units. There is a linear relationship between the total effectiveness of deployed units and the number of units deployed. What is meant

here is that the total effectiveness of ten deployed mechanized battalions is ten times the effectiveness of one mechanized battalion. This is realistic if, in the case here, there are a few number of units to deploy.

6. Only one APOD considered. There is only one APOD that aircraft can fly into to deliver troops and cargo. The decision to limit the model to one APOD was made in order to keep the matrix dimensions small without losing the realism. By adding another APOD, the constraint matrix would increase considerably.(7:79)

Scenario Limitations: The possible scenarios in a model are an important factor in any analysis. If the scenario of the model is not similar to the situation on hand, then the results of the model will not accurately measure the real outcomes. Therefore, the scenarios of DEPLOY are explicitly stated.

1. No attrition considered. Regardless of the mission or the aircraft type, no attrition is considered. This characteristic of DEPLOY was decided because, to adequately address attrition, the units would have to be matched up with specific aircraft as attrition is a function of the aircraft type and the mission being flown.

2. Airborne delivery is limited by riggers. A limiting factor of the amount of cargo delivered via air-delivery is the maximum army rigger output. It is assumed that there is no constraint on available parachutes.

3. The time span considered is four five-day periods. The time span is a set of four linked periods of five days. By reducing the number of time periods considered, the number of

VII. CONCLUSION

DEPLOY is a unique interactive user-friendly computer program that optimizes the combat power delivered to a theater during a specified time period. Further, the units deployed are kept within acceptable levels of sustainability. This final chapter is divided into two major sections. The first section will discuss a summary of the modeling process. Next, the major accomplishments of this thesis will be presented and, finally, recommendations for further research will be given.

Model Summary

This thesis was initiated after a recommendation from the thesis "Optimizing Force Development and Force Structure for the Rapid Deployment Force" completed in March 1984 by Army Captain James Cooke. Cooke demonstrated for the first time a particular methodology which determines how the interactions of unit weight, combat attributes, logistics needs, and airlift resources can be jointly optimized. (7,4) The recommendation from Cooke's thesis was to take the mathematical form of the model and develop it into a user-friendly computer package that could be used to analyze different force mixes. DEPLOY accomplished this recommendation and more. The major accomplishments of DEPLOY are discussed in the next section.

Major Accomplishments

There are three major accomplishments completed in DEPLOY. First, the mathematical model demonstrated by Cooke is developed into an interactive user-friendly computer package and, in addition to this, the user is able to perform analysis on different force mixes and different AFOD sets. Second, DEPLOY

An analysis of the output reveals many interesting results. First, the C-5 and C-141 are very heavily relied upon for all types of missions and for all sizes of cargo. Furthermore, when there is no direct delivery capability, airborne delivery is used extensively. This points out the requirement for a direct delivery type aircraft, such as the C-17. Finally, the importance of deploying fighter squadrons and ALCEs is shown in that the minimum number of both are deployed.

Summary

This chapter performed the necessary verification and demonstration of DEPLOY. The verification ensured the correctness of the model with different data sets using MFOS for an allocation routine. The demonstration showed that DEPLOY does indeed work. Here also, an example was discussed along with the meaning of the output from DEPLOY.

```

178  ALCE          APCD  2      1.3306142
187  ALCE-FRONT    INTRA 3      .6680929

```

 OTHE GOAL ACHIEVEMENTS ARE

PRIORITY	GOAL NUMBER	OVER-ACHIEVEMENT	UNDER-ACHIEVEMENT
2	78	.0000	2.6598
3	79	139.2600	.0000
4	80	.0000	36.2161
5	81	.0000	1.9997

The output of PAGP gives more than just the variable values. First, the subproblem or priority number on which the optimization ended is specified. This number will always be less than or equal to the number of priorities considered in the problem. One of the efficient coding techniques of PAGP discussed in Chapter IV is the use of special stopping rules. Here, the program is stopped if non-basic variables are about to enter the basis. As a result of this stopping rule, the ending subproblem may be less than the number of priorities in the problem. Second, the number of constraints, including goals, considered in the final problem is given. Next, the variable number, name and value is specified. In order to simplify the output, only the non-zero aircraft unit variables are given. The meaning of the name follows the order of the subscripts of the variable discussed in Chapter III. For aircraft variables, the type of aircraft, mission or destination, cargo type, and period of flight is specified. For unit variables, the name of the unit, mode of delivery or destination, and period of deployment is given. Finally, the attainment of the goals are specified in terms of over-achievement and under-achievement.

Table 6-6. Simulated PAGP Output.

THE OPTIMIZATION ENDED ON SUBPROBLEM 5
THERE WERE 126 CONSTRAINTS IN THE FINAL OPTIMAL TABLEAU.

THE OPTIMAL SOLUTION FOR THE DECISION VARIABLES X(J)

1	C-5	APOD	OUT	1	.6396083
2	C-5	APOD	OUT	2	3.6028938
3	C-5	APOD	OUT	3	3.5383696
4	C-5	APOD	OUT	4	6.7929937
5	C-5	APOD	OVER	1	4.5360120
6	C-5	APOD	OVER	2	55.9677715
9	C-5	APOD	BULK	1	1.9556843
17	C-5	APOD	PAX	1	1.1106032
21	C-5	INTRA	OUT	1	.6361074
23	C-5	INTRA	OUT	3	3.5383696
24	C-5	INTRA	OUT	4	6.7929937
27	C-5	INTRA	OVER	3	15.8438586
28	C-5	INTRA	OVER	4	26.9288448
33	C-5	INTRA	BULK	3	7.3438638
35	C-5	INTRA	BULK	4	20.8414566
42	C-5	AIREN	OUT	2	2.5010077
43	C-5	AIREN	OUT	3	.8682231
45	C-5	AIREN	OVER	1	37.8306337
46	C-5	AIREN	OVER	2	2.2263776
49	C-5	AIREN	BULK	1	17.7928981
51	C-5	AIREN	BULK	2	12.3571019
57	C-5	AIREN	PAX	1	8.1289054
58	C-5	AIREN	PAX	2	6.4454446
59	C-5	AIREN	PAX	3	.6690075
62	C-141	APOD	OVER	2	6.7172956
63	C-141	APOD	OVER	3	198.7123456
64	C-141	APOD	OVER	4	52.6629820
93	C-141	AIREN	OVER	1	246.0000000
94	C-141	AIREN	OVER	2	239.5513962
109	C-130	INTRA	OVER	1	19.2356735
111	C-130	INTRA	OVER	3	90.0000000
112	C-130	INTRA	OVER	4	90.0000000
113	C-130	INTRA	BULK	1	11.1528047
128	747C	APOD	BULK	2	7.4425366
130	747C	APOD	BULK	3	7.8594519
132	747C	APOD	BULK	4	18.5257392
136	747P	APOD	PAX	4	10.9819335
137	AREN	BTLN	DIRECT	1	5.3113129
138	AREN	BTLN	DIRECT	2	3.6886871
142	AREN	BTLN HQ	DIRECT	2	2.2269247
143	AREN	BTLN HQ	DIRECT	3	.7730753
149	AIR	ASLT BTLN	INTRA	1	.2720196
151	AIR	ASLT BTLN	INTRA	3	.7395374
152	AIR	ASLT BTLN	INTRA	4	2.9047293
160	MECH	BTLN	INTRA	4	.0000125
170	F-16	SQDN	APOD	2	2.2268286
171	F-16	SQDN	APOD	3	.7731714
177	ALCE		APOD	1	.0012929

Output Results: In this section the output of the allocation routine is given along with an explanation of the output.

Here, the process of acquiring the output data is explained. DEPLOY generated the appropriate constraints and goals from the input mentioned above. Then, the necessary information was passed to PAGP, the allocation routine, to satisfy the constraints and maximize the goals. At this point, a very unusual situation occurred. Though PAGP had been checked with a small problem in the identical format for the deployment problem, an examination of the output proved the results invalid. The output was invalid simply because the constraints were not satisfied and, therefore, the answers were infeasible. In an attempt to show that the problem could be solved, the Multi-Purpose Optimization System (MPOS) version 4.0, an LP package, was used. The method of Sequential Goal Programming (SGP) was utilized to solve the GP problem with MPOS. Here, the first goal became the objective function to maximize while lower goals were discarded. Then, the lessor of the goal or the attainment of the objective function became a new constraint, while, the second goal entered as the objective function to maximize. This iterative process continued until all the goals were considered. The results of the final output are listed in Table 6-6.

Table 6-3. APOD Data

1. DISTANCES						
FROM US:		7200.				
FROM CORPS:		200.				
2. AIRCRAFT	INTER	INTRA	OUT	OVER	BULK	
C-5:	1.5	.2	1.0	1.0	1.0	
C-141	1.5	.2	.0	1.0	1.0	
C-130	.0	.2	.0	1.0	1.0	
747P	1.5	.0	.0	1.0	1.0	
747C	1.5	.0	.0	.0	.0	
3. COMBAT VALUES OF UNITS CLOSING						
PERIOD 1		2.5				
PERIOD 2		1.8				
PERIOD 3		1.3				
PERIOD 4		1.1				
4. MISCELLANEOUS DATA						
MHE AT APOD		500				
MHE AT FRT		500				
ALCE CAP		600				
RIGGER CAP		430.0				

Table 6-4. Deployable Units Data

8 5 2 0															
TYPE	NUMB	OUT	OVER	BULK	PAX	SPL	CP	AT	FT	AC	PK	TP	TV	AB	CI
AREN BTLN	9	0	1400	268	1400	149	4	19	4	0	0	0	20	1	1
AREN BTLN HQ	3	73	0	0	305	0	0	0	0	0	0	0	20	1	-1
AIR ASLT BTLN	9	152	990	574	1960	446	6	29	4	0	0	0	18	0	1
MECH BTLN	9	2055	3865	151	2052	440	8	40	6	0	0	0	14	0	1
155MM ART BTLN	3	139	1243	91	710	212	3	3	0	0	0	0	12	0	-1
F-16 SQDN	3	0	249	155	472	102	8	36	0	18	48	0	0	0	0
MDM TRUCK CMFY	3	1098	405	270	250	55	0	0	0	0	0	182	40	0	0
ALCE	2	176	2213	244	400	73	0	0	0	0	0	0	0	0	0

Table 6-5. Goals for DEPLOY

GOAL # 2 IS 25.0 OF FRONT LINE TRACE BY PERIOD 1
 GOAL # 3 IS 60.0 OF ANTI-TANK POWER BY PERIOD 2
 GOAL # 4 IS 180.0 OF FIREPOWER BY PERIOD 3
 GOAL # 5 IS 2.0 OF MECH BTLN BY PERIOD 4

Input Parameters: The source and rationale for the different input parameters are discussed in this section. The majority of these parameters which indicate that DEPLOY works properly were the same as those used by Cooke. The airport limitations and unit weight were obtained from reference 2, along with the aircraft limitations and capabilities but were "checked and modified by HQ USAF/SAGM." As a result of this modification, the C-5 UTE Rate was changed from the 5.5 suggested by the Army, to 12.5 as suggested by the Air Force. The measure of front line trace was "taken from doctrinal publications," while the anti-tank capability was calculated "from TOE data for unit owned TOW and DRAGON weapon systems, ... using a value of 1 for TOW systems and 0.5 for DRAGON systems." The firepower for the different units was not as easily obtained. Here, the firepower "was developed from the relative firepower of units in a force-on-force war game at the Army War College." The actual data files used are given in Tables 6-2 through 6-4, while the goals are shown in Table 6-5. The goals start at the second priority, because the first priority is reserved for the inequality constraints. (7:81)

Table 6-2. Aircraft Data

TYPE	NMBR	AVL	DRCT	APOD	INTRA	ABRN	OUT	OVER	BULK	PAX	SPD	GTM	MHE	FP	PRK
C-5	60	.8	.0	12.5	9.0	0.0	65	25	60	25	80	25	320	430	3.3 36 1 5
C-141	180	.8	.0	12.5	10.0	12.0	0	0	21	14	25	14	110	410	2.3 13 1 12
C-130	300	.8	.0	.0	4.0	.0	0	0	14	6	14	6	64	270	1.5 6 1 16
747C	30	.8	.0	10.0	.0	.0	0	0	60	0	90	0	0	430	3.6 36 2 5
747P	20	.8	.0	10.0	.0	.0	0	0	0	0	0	0	364	430	2.8 36 2 5

Sensitivity. There are three sensitivity options in DEPLOY. These options allow the user to make specific changes to the aircraft, APDD, and deployable units data set. The changes are made one at a time, from a menu of different possibilities. Furthermore, the particular data set can be viewed after any change. To verify the editors, each option was exercised and the data viewed. In all cases, the data was correctly changed and placed in the program. Each of the sensitivity options were verified using this method. An illustration of the sensitivity portion is given in the user's guide in Appendix A.

PAGP. This is the program used to allocate the resources according to the methodology of Goal Programming. In its basic form, PAGP was designed to read inputs interactively. Therefore, since DEPLOY and PAGP are run separately, there was a requirement to pass information via files. Considering that PAGP was verified by Bunnell and Takacs in BRIK (4:77), the only verification necessary was to ensure that the proper information was properly placed. To confirm this, several write and print statements were incorporated into PAGP to check the passed data. By testing different data sets, the input parameters were visually examined as the program performed its allocation. Again, in all cases the data was correctly input. Therefore, PAGP was verified for small problems.

Demonstration

This section demonstrates the output of DEPLOY with reasonable data. There are two main areas discussed in this section, the input parameters and the allocation output.

variable is specified. If the variable is cargo aircraft, the aircraft type is given along with the type of mission or destination, cargo type and period of flight. If the variable is a deployed unit, the name is given along with its mode and time period of travel. The mode of travel of a unit is discussed in Chapter III under the variable and subscript sections. Further, if the variable is a slack or surplus variable, no name is given. An example of a constraint printout is given in Table 6-1. By using this feature, several different data sets were entered and the resulting constraints examined. In all cases, the constraint matrix correctly reflected the proper results. Therefore, the matrix formulation was verified.

Table 6-1. Constraint Output

THEATER SUPPLIES						
AIJ(	33	12)=	80.00000	C-5	APOD	SUPP 2
AIJ(	33	52)=	80.00000	C-5	AIREN	SUPP 2
AIJ(	33	68)=	25.00000	C-141	APOD	SUPP 2
AIJ(	33	100)=	25.00000	C-141	AIREN	SUPP 2
AIJ(	33	129)=	90.00000	747C	APOD	SUPP 2
AIJ(	33	137)=	-745.00000	AREN	BTLN	DIRECT 1
AIJ(	33	141)=	.00000	AREN	BTLN HQ	DIRECT 1
AIJ(	33	145)=	-2230.00000	AIR	ASLT BTLN	APOD 1
AIJ(	33	149)=	-2230.00000	AIR	ASLT BTLN	INTRA 1
AIJ(	33	153)=	-2200.00000	MECH	BTLN	APOD 1
AIJ(	33	157)=	-2200.00000	MECH	BTLN	INTRA 1
AIJ(	33	161)=	-1060.00000	155MM	ART BTLN	APOD 1
AIJ(	33	165)=	-1060.00000	155MM	ART BTLN	INTRA 1
AIJ(	33	169)=	-510.00000	F-16	SQDN	APOD 1
AIJ(	33	173)=	-275.00000	MDM	TRUCK CMFY	APOD 1
AIJ(	33	177)=	-365.00000	ALCE		APOD 1
AIJ(	33	181)=	-365.00000	ALCE-FRONT		APOD 1
AIJ(	33	185)=	-365.00000	ALCE-FRONT		INTRA 1
AIJ(	33	190)=	-1.00000			
AIJ(	33	189)=	1.00000			
RHS(	33)=		.00000			

VI. Verification and Demonstration

This chapter discusses the necessary verification and demonstration of DEPLOY. The verification portion of the chapter will ensure the correctness of the model through a scrutiny of the various parameters. The demonstration will show, with reasonable data, that the model accomplishes its designed purpose.

Verification

This section verifies DEPLOY by considering three critical areas. First, the correctness of the matrix formulation will be examined to ensure that it properly reflects the goals and constraints discussed in Chapter III. Next, the sensitivity portion, which changes input parameters will be tested to ensure that the desired changes are in fact made and not misplaced. Finally, the necessary input parameters for the allocation routine, PAGP, will be inspected for their proper placement.

Matrix Formulation. The matrix is comprised of the data set entered by the user and the mathematical equations discussed in Chapter III. Therefore, the model assimilates the data given by the user and develops the necessary constraints and goals. This is a noteworthy feature of DEPLOY, which helps the user examine the matrix formulation. The user has the option of observing the constraints being generated at the terminal and/or reading a hard copy of the matrix by routing the file "OFILE" to the printer. The echo of the constraint is extremely easy to read and understand. Following the title of the constraint set, the row, column and value of the variable is given. Then, the name of the

addressed first in relationship to the application and then to the usefulness of DEPLOY. Next, the input requirements for the model were examined. In the next chapter a demonstration and verification of the model will be reviewed.

EAS(I,K)	The relative ease with which cargo K can be down-loaded for air-craft I.
CPI(L)	The combat power indices for period L.
NHA	The amount of MHE in pallet equivalents prepositioned at the APOD.
NHF	The amount of MHE in pallet equivalents prepositioned at the front.
NPAL	The number of pallet equivalents the ALCE can download.
RC	The riggers' capacity to prepare airdrop loads per day.

Table 5-3
Input Requirements for the Deployable Units

UNAME(Y)	The name of the unit.
NUINT(Y)	The number of units available for deployment.
NTEN(Y,K)	The equipment tonnage of size K for unit Y.
NFP(Y)	The firepower exhibited by unit Y.
NAT(Y)	The antitank force of unit Y.
NFT(Y)	The front line trace capability of unit Y.
NTAC(Y)	The number of fighter aircraft assigned to unit Y.
NPK(Y)	The number of fighters that can park at the APOD.
NTP(Y)	The ton-miles/day the unit can transport supplies.
NTV(Y)	The miles/day the unit can transport itself.
IAB(Y)	The airborne capability indicator.
NCI(Y)	The combat unit type of unit Y.

Table 5-4
Input Requirements for the Goals

ITYPE	The type of goal.
DEJ	The desired level of the goal.
IDP	The desired period to accomplish.

Conclusion

Two aspects of the model have been discussed in this chapter. The restrictions and scenario limitations were

Input Requirements

As in most models, a complete description of the entities of the model is necessary. The entities for DEPLOY are divided into four main areas; aircraft, APOD, deployable units, and goals. Each area requires specific information, and is stated in Tables 5-1 through 5-4. With this information, the different constraints and goals discussed in Chapter III are constructed. Once these constraints and goals are built, the information is fed into PAGP to optimally allocate resources for the established constraints and goals.

Table 5-1
Input Requirements for the Aircraft Inventory

ACNAME(I)	The name of the aircraft.
NAC(I)	The number of aircraft in the inventory.
AVAIL(I)	The availability of the aircraft.
UTE(I,J)	The UTE rate for a particular mission.
KARG(I,K)	The cargo carrying capability for different sizes of cargo.
KTS(I)	The cruise TAS of the aircraft.
GT(I)	The enroute ground time of the aircraft.
MHE(I)	The maximum number of pallets the aircraft can carry.
NFF(I)	The first period the aircraft is available.
NPRK(I)	The number of aircraft that can park at the APOD.

Table 5-2
Input Requirements for the APOD Complex

DUSAPD	The distance from the US to the APOD.
DAPDFT	The distance from the APOD to the front.
INTER(I)	The round trip time for an inter-theater mission for aircraft I.
INTRA(I)	The round trip time for an intra-theater mission for aircraft I.

variables and constraints are significantly reduced.

4. All units are immediately available and are located in the US. All units to be deployed are ready for deployment at the beginning of the first period. Also, the cargo and troops are considered to be in a central location.

5. A constant daily amount of supplies is necessary. In order to sustain a unit once it has deployed, only a predetermined constant amount of supplies is necessary. There is no provision for a changing supply requirement for deployed units. Furthermore, it is assumed that each unit has sufficient supplies to sustain itself until the following period.

6. Intratheater specific aircraft are not deployed at the APDD. Aircraft such as the C-130 that specialize in intratheater airlift are deployed at an area different from the APDD. If this were not the case, "the ramp and serving requirement would restrict the capability of the APDD to handle strategic airlift in the theater".(7:81)

7. The ratio of combat to combat support units is set to at least 1:1. This ratio is explained in Chapter IV under the section labeled Unit Linkage. Furthermore, this ratio can be easily changed by altering the code in subroutine CONSTR under the title Combat and Combat Support.

8. Airborne deliveries originate from the U.S. Units delivered via airdrop arrive directly from the U.S. There is no capacity to load the airborne cargo at the APDD. Also, airborne units can only be delivered via airdrop. Therefore, if there are no airdrop capable aircraft, no airborne units would be deployed.

allows the user to perform two possible types of analysis: optimal accomplishment of certain Army objectives and also determination of the minimum force structure is necessary to accomplish the goals of the Army. Finally, DEPLOY has made the model more flexible than that demonstrated by Cooke. Each of these accomplishments will be discussed in detail.

Computer Package: DEPLOY permits the user to easily define the desired force structure and AFOD data. The force structure is comprised of aircraft and Army and Air Force units that may participate in the deployment. Each aircraft and unit has specific characteristics specified by the user to define the problem. Data containing the force structure and AFOD data may be entered interactively or from a user-defined file. Furthermore, DEPLOY allows the user to make certain changes to the entered data, thereby saving the user the time necessary to re-enter the complete data set to make minor changes.

Types of Analysis: DEPLOY performs two types of analysis. The first type of analysis determines the degree to which the Army goals can be met with a given fleet of aircraft and deployable units. The Army goals specify levels of frontline trace, anti-tank power, and fire power. In addition, an Army goal may include the transport of a certain type of unit to the front. For instance, a goal may be to position two mechanized battalions to the front by the tenth day of battle. Each type of goal considered in the analysis is determined by the user along with the priority, level of attainment, and period of completion.

The second type of analysis determines the minimum force structure necessary to accomplish the given Army goals. Here, the Army goals become hard constraints while the aircraft become the goals to minimize. When performing this type of analysis, the user has the option to treat all aircraft types equal or to weight the aircraft according to cost or any other means. By allowing the user the option of two types of analysis, DEPLOY is more versatile in meeting the needs of the analyst.

Flexibility: By having fewer restrictions, DEPLOY is more flexible in reflecting the actual deployment of armed forces. The model used by Cooke to demonstrate his methodology had two restrictions that are not binding in DEPLOY. The two restrictions are that intratheater airlift was "limited to moving bulk cargo supplies" and ground travel time was considered equal for all deployed units. (4:80) As a result of the first restriction, the deployed units to the front had to be delivered directly to the front or delivered to the APOD and then required to transport themselves to the front. If a unit was unable to transport itself or not airborne capable and there were no direct delivery aircraft available, the unit could not be deployed. An example of a unit that is most likely unable to transport itself and is not airborne capable is an ALCE unit deployed to the front to help facilitate the incoming aircraft. DEPLOY removes this restriction and allows units to be transported to the front via intratheater airlift. The second restriction analyzed shows that all units may not be able to transport themselves at the same speed. DEPLOY allows the user to enter the distance per day that a specific unit can move and

then calculates the time required to travel from the APDD to the front. This allows the model to more accurately reflect the individual units. DEPLOY then, is more flexible in its reflection of the real deployment of troops than the model used by Cooke.

Recommendations

Research is an iterative process and, therefore, extended study would yield continued insight and practical application. DEPLOY is no exception, so that the following recommendations are made for further research:

1. Attrition of aircraft and troops should be addressed to determine the effects that it has on deployment and sustaining the units deployed.
2. A full-scale model should be analyzed with DEPLOY to determine what can be accomplished with the present day inventory of aircraft.
3. Analysis should be completed to examine Right Hand Side (RHS) ranging along with other types of sensitivity analysis.
4. PAGP should be connected so that it can be used in conjunction with DEPLOY.

APPENDIX A

User Guide

This guide will familiarize the user with many of the features of DEPLOY. It will proceed through DEPLOY according to the various model function sequences.

DEPLOY was developed to be "user-friendly." The definition of "user-friendly" is taken to mean that anyone with the necessary data and a limited background in either computer or mathematical programming could use the model. Since DEPLOY displays prompts and menus on the screen to guide the user through the model's functions, the user needs access to a computer equipped with a monitor.

Many of the prompts which appear on the screen are questions that require answers. Realizing that a user could input an illogical response or incorrect data, the model incorporates many internal checks to guard against these possibilities. An example of a check can be found at any yes or no ($1=y/0=n$) question. If the user responds with something other than a 1 or 0, an error statement is issued and the user is asked to enter in a proper response. Also, the model discourages the user from inputting inappropriate data. For example, probabilities are accepted only if they are between zero and one, inclusive. Again, if the data is out of range, an error statement is given and the user is told to enter in an appropriate probability.

This user guide is divided into four main sections. The first section will discuss the process of conducting new analysis. The defining of the goals will be reviewed in the

second section. Then, in the third section, the sensitivity/capability of DEPLOY will be discussed. Finally, the allocation output will be explained.

New Analysis

DEPLOY permits the user to input data interactively or through user-designated files. The input data is divided into three subsections: aircraft data, APOD data, and deployable units data. This section will describe the type of data needed by DEPLOY for new analysis.

Data is entered in the following order: aircraft, APOD and deployable units data. Since DEPLOY follows the same basic operation of input for all three data types, the following explanation applies to all three data sets. The first decision the user must make is whether the data is to be entered interactively or from a file. If there has not been any previous analysis which created a usable data base stored in a user-designated file, the data must be entered interactively. If the data is entered interactively, DEPLOY will guide the user with a series of questions. These questions insure that the required data is input in the correct format. After the data is entered, either interactively or via an external file, the user has the option to view the data set.

It is possible to build a data file independently of DEPLOY, but it is easier to input the data interactively and allow the model to build the external file. If the data base is input from an external file, the user must insure the file exists prior to operating DEPLOY. If a data file is built, its name must be less than seven characters long. Since the file must be formatted, it

is easier to allow DEPLOY to create the file.

The rest of this section describes the input requirements for the three types of data bases used in DEPLOY. Each section will list and briefly describe the required parameters of DEPLOY.

Aircraft Input: The data for the aircraft may be entered interactively or by an external file. An example of an aircraft data file is in Table A-1. The following is a list of required aircraft data:

Aircraft name	--	Maximum of 6 characters.
Number of aircraft	--	Integer, format limit = 999.
Aircraft availability	--	The percentage of aircraft in the fleet that may participate in the deployment. Real, between 0 and 1, inclusive.
Aircraft UTE rate	--	Real, between 0 and 24 hours/day, inclusive.
Cargo capacity	--	The tonnage capacity of the aircraft for out, over, and bulk sized cargo along with the number of passengers that may accompany the different cargo loads. Also, the number of passenger capacity without any cargo load. Integer, format limit = 999.
Aircraft speed	--	True Air Speed (TAS) of the aircraft in knots. Integer, format limit = 999.
Aircraft ground time	--	The ground time in hours to off-load, up-load, and refuel the aircraft at the APOD or front. Real, format limit = 99.
Pallet capacity	--	The maximum number of pallets the aircraft can carry. Integer, format limit = 999.
First period to deploy	--	The first period the aircraft will be available in the deployment. Integer, format limit = 99.

Aircraft ramp space -- The maximum number of aircraft that can park at the APOD with no other types of aircraft present. Integer, format limit = 999.

Table A-1. Example of Aircraft Data File

5

TYPE	NMBR	AVL	DRCT	APOD	INTRA	ABRN	OUT	OVER	BULK	PAX	SPD	GTM	MHE	FP	PRK
C-5	60	.8	.0	12.5	9.0	9.8	65 25	60 25	80 25	320	430	3.3	36	1	5
C-141	180	.8	.0	12.5	10.0	12.0	0 0 21	14 25	14 110	410	2.3	13	1	12	
C-130	300	.8	.0	.0	4.0	.0	0 0 14	6 14	6 64	270	1.5	6	1	16	
747C	30	.8	.0	10.0	.0	.0	0 0 60	0 90	0 0	430	3.6	36	2	5	
747P	20	.8	.0	10.0	.0	.0	0 0 0	0 0	0 0	364	430	2.8	36	2	5

APOD Input: As in the aircraft input section, data may be entered interactively or by a user-designated file. An example of an APOD data file is given in Table A-2. The following list contains required APOD and miscellaneous parameters.

Distances -- The distance from the U.S. to the APOD and from the APOD to the front in km. Real, format limit = 99,999.

Intertheater turntimes -- The time required in days for each aircraft, from the start of one intertheater mission until the start of another intertheater mission immediately following. Real, format limit = 10.

Intratheater turntimes -- The same as intertheater turntimes, but for intratheater missions. Real, format limit = 10.

Off-loading cargo ease -- The relative ease with which a specific size of cargo for a specific aircraft can be off-loaded. Real between 0 and 1, inclusive. The default value is 1.

Combat power indices -- The time-dependent portion of firepower. Real, format limit 99.

Prepositioned MHE -- The amount of MHE measured in Pallet Equivalents (PE) prepositioned at the APOD and the front airfield. Integer, format limit = 99,999.

ALCE capability -- The number of PE's an ALCE unit can off-load in one day. Integer, format limit = 99,999.

Rigger capability -- The number of airdrop loads the Army rigger unit is able to prepare in one day. Real, format limit = 99,999.

Table A-2. Example of APOD Data File

```

1. DISTANCES
  FROM US:      7200.
  FROM CORPS:   200.
2. AIRCRAFT
  C-5:          1.5   .2   1.0   1.0   1.0
  C-141         1.5   .2   .0   1.0   1.0
  C-130         .0   .2   .0   1.0   1.0
  747P         1.5   .0   .0   1.0   1.0
  747C         1.5   .0   .0   .0   .0
3. COMBAT VALUES OF UNITS CLOSING
  PERIOD 1      2.5
  PERIOD 2      1.8
  PERIOD 3      1.3
  PERIOD 4      1.1
4. MISCELLANEOUS DATA
  MHE AT APOD   500
  MHE AT FRT    500
  ALCE CAP      600
  RIGGER CAP    430.0

```

Deployable Units Input: As in the previous two cases, the data may be entered either interactively or via an external file. An example of a Unit Data file is given in Table A-3. The following list contains the required input data concerning the units that may be deployed.

Unit name -- Maximum of 14 characters.

Number of Units -- Integer, format limit = 99.

Equipment of Units -- The tonnage of equipment in different cargo sizes (out, over, and bulk). Integer, format limit = 99,999.

Personnel in Unit -- The number of personnel assigned to the unit. Integer, format limit = 99,999.

Unit consumption rate -- The consumption rate of supplies by the unit for one day. Integer, format limit = 9,999.

Firepower of Unit -- The amount of firepower the unit displays. Real, format limit = 99.

Anti-tank power -- The amount of anti-tank power displayed by the unit. Real, format limit = 99.

Front line trace -- The amount of defense frontage the unit can maintain. Real, format limit = 99.

Fighter Aircraft -- The number of fighter aircraft assigned to the unit. Real, format limit = 99.

Ramp space used -- The maximum number of fighter aircraft only that can park at the APOD. Integer, format limit = 99.

Transport capability -- The thousands of ton-miles/day the unit can move supplies from the APOD to the front. Integer, format limit = 99.

Unit speed -- The distance a unit can move itself in one day without any assistance from another unit. Integer, format limit = 99.

Table A-3. Example of Unit Data File

8 5 2 0

TYPE	NUMB	OUT	OVER	BULK	PAX	SPL	FP	AT	FT	AC	PK	TP	TV	AB	CI
ARBN BTLN	9	0	1400	268	1400	149	4	19	4	0	0	0	20	1	1
ARBN BTLN HQ	3	73	0	0	305	0	0	0	0	0	0	0	20	1	-1
AIR ASLT BTLN	9	152	990	574	1960	446	6	28	4	0	0	0	18	0	1
MECH BTLN	9	2055	3885	151	2052	440	8	40	6	0	0	0	14	0	1
155MM ART BTLN	3	139	1243	91	710	212	3	3	0	0	0	0	12	0	-1
F-16 SQDN	3	0	249	155	472	102	8	36	0	18	48	0	0	0	0
MDM TRUCK CMPY	3	1098	405	270	250	55	0	0	0	0	0	182	40	0	0
ALCE	2	176	2213	244	400	73	0	0	0	0	0	0	0	0	0

DEFINING THE GOALS

DEPLOY performs two types of analysis. The first type of analysis determines the degree to which the Army goals can be met with a given fleet of aircraft and deployable units. The Army goals specify levels of frontline trace, anti-tank power, and firepower. In addition, an Army goal may include the transport of a certain type of unit to the front. For instance, a goal may be to position two mechanized battalions to the front by the tenth day of battle. The second type of analysis determines the minimum force structure necessary to accomplish the given Army goals. Here, the Army goals become hard constraints while the aircraft become the goals to minimize. The menu of the choice of analysis is given in Table A-4. When performing the second type of analysis, the user has the option of treating all aircraft equally or weighting the aircraft according to cost or any other means determined by the user.

Table A-4. Menu of the Different Types of Analysis

WHAT QUESTION DO YOU WANT ANSWERED? SELECT ONE.

-
1. FORCE STRUCTURE- USE THE LEAST NUMBER OR LEAST COSTLY AIRCRAFT TO ACCOMPLISH THE ARMY MISSION.
 2. ARMY ACCOMPLISHMENT- GIVEN A FORCE STRUCTURE, ACCOMPLISH THE ARMY GOALS AS BEST AS POSSIBLE.
-

Once the type of analysis has been determined, the different Army goals must be defined. The user has the option of specifying levels and completion times of front line trace, anti-tank power, and firepower along with a specific type of unit. An example of the menu to define the goals is given in Table A-5.

The first priority goal is reserved for certain constraints, therefore, the user may define goals from 2 to 11, or 10 different goals. The following list contains the information necessary to define the goals:

Type of goal	-- The specified goal from the list of possible goals.
Level of the goal	-- The amount of the goal desired. Real.
Period of completion	-- The desired completion time period of the goal. Integer.

Table A-5. Menu to Define the Goals

SELECT THE GOAL TYPE FOR PRIORITY 2

```
-- 1 FRONT LINE TRACE
-- 2 ANTI-TANK POWER
-- 3 FIREPOWER
-- 4 ARBN BTLN
-- 5 ARBN BTLN HQ
-- 6 AIR ASLT BTLN
-- 7 MECH BTLN
-- 8 155MM ART BTLN
** IF FINISHED TYPE 0
```

After the goals have been defined, the user may view the goals that were defined. Table A-6 is an example of defined goals.

Table A-6. Example of Defined Goals.

```
GOAL # 2 IS 25.0 OF FRONT LINE TRACE BY PERIOD 1
GOAL # 3 IS 60.0 OF ANTI-TANK POWER BY PERIOD 2
GOAL # 4 IS 180.0 OF FIREPOWER BY PERIOD 2
GOAL # 5 IS 2.0 OF MECH BTLN BY PERIOD 2
```

SENSITIVITY ANALYSIS

This section describes the three major sub-areas of a sensitivity analysis on changes in aircraft data, APD data and deployed units data. To determine the data set to be altered,

the user selects the desired option from the main menu of sensitivity analysis. The main menu for the sensitivity analysis is given in Table A-7. Once the user selects the desired option, he will be asked if the data set to be altered has been input either interactively or via an external file. If the data has not been entered, the user is directed to enter the data set. Once assured that the data is in DEPLOY, a submenu is displayed from which the user directs the appropriate changes. The submenus are given in Tables A-8 through A-10.

Table A-7. Sensitivity Analysis Main Menu

SENSITIVITY MENU

1. AIRCRAFT DATA:
 - DELETE AN AIRCRAFT.
 - CHANGE NUMBER, AVAILABILITY, GROUND TIME.
 2. AFOD DATA:
 - CHANGE DISTANCES, TURNTIMES, COMBAT VALUES.
 3. ARMY DATA:
 - DELETE A UNIT.
 - CHANGE NUMBER AVAILABLE.
 - CHANGE FIREPOWER, ANTI-TANK, FRONT LINE TRACE VALUES.
-

Table A-8. Aircraft Submenu

AIRCRAFT CHANGE SUBMENU

1. DELETE AN AIRCRAFT.
2. CHANGE THE NUMBER OF AIRCRAFT AVAILABLE.
3. CHANGE THE AVAILABILITY OF AN AIRCRAFT.
4. CHANGE THE GROUND TIME FOR AN AIRCRAFT.

REPLY 0 IF NO MORE CHANGES DESIRED

Table A-9. APOD Submenu

APOD CHANGE SUBMENU

1. CHANGE THE DISTANCE FROM THE US TO THE APOD.
2. CHANGE THE DISTANCE FROM THE APOD TO THE FRONT.
3. CHANGE THE COMBAT VALUES FOR THE PERIODS.
4. CHANGE THE MHE PREPOSITIONED AT THE APOD.
5. CHANGE THE MHE PREPOSITIONED AT THE FRONT.
6. CHANGE THE CAPACITY OF THE ALCE.
7. CHANGE THE CAPACITY OF THE RIGGERS.

REPLY 0 IF NO MORE CHANGES DESIRED

Table A-10. Deployable Units Submenu

ARMY CHANGE SUBMENU

1. DELETE A UNIT.
2. CHANGE THE NUMBER OF UNITS AVAILABLE.
3. CHANGE THE FIREPOWER FOR A UNIT.
4. CHANGE THE ANTI-TANK POWER FOR A UNIT.
5. CHANGE THE FRONT LINE TRACE FOR A UNIT.

REPLY 0 IF NO MORE CHANGES DESIRED

OUTPUT

The final section of this chapter will examine the output, or the allocation routine for DEPLOY. Because of the common block size limitation of ASD's CDC-750, DEPLOY and the allocation routine are two separate programs. Therefore, files are used to pass the needed information from DEPLOY to FAGP. There are three such files, TFILE, IFILE, and IDFILE. After these files are read into FAGP, and the goals are optimized, the results are written to the file GPANS. Each of these files will be discussed separately.

IEILE: this represents the "top" file, as it contains the necessary initial information for PAGP. The different input parameters defined in TFILE are as follows:

1. The number of priorities considered.
2. The number of variables.
3. The number of equality (real) constraints.
4. The total number of constraints (real and inequalities).
5. The number of named variables.
6. The number of constraints for each priority.
7. The deviational variable to be minimized.
8. The weight of the deviational variable.
9. The number of non-zero variables in each constraint and goal.

Once these parameters are read into PAGP, the coefficients of the variables are inputted.

IEILE: This corresponds to the "input" file, as it contains the values of the non-zero coefficients in the constraints and goals. It was decided early in the project to read in the non-zero terms because approximately 94% of the matrix was zero. Therefore, to correctly determine where in the matrix the non-zero coefficients should be placed, the row and column were specified. The format of the file contains two integers followed by a real number. The integers specify the row and column respectively, while the real number is the coefficient value to be placed in the matrix. As a check, the row value read in was compared to the row on which PAGP was presently operating. If the two row values were different, an error message was issued and the program aborted. Finally, by IEILE containing only the non-zero coefficients, the file creation time was reduced to one-fifth the normal creation time, and the storage space was decreased to one twelfth the normal size had all the zero and non-zero terms been included in the file. The smaller storage

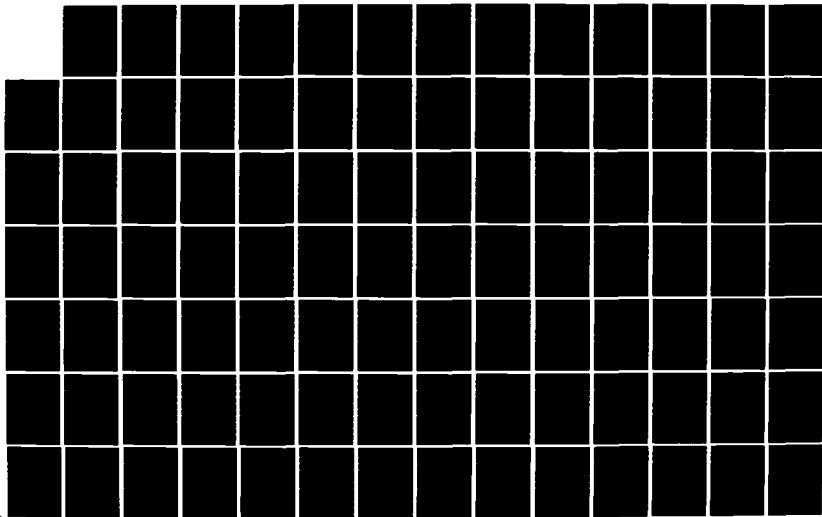
AD-A152 007

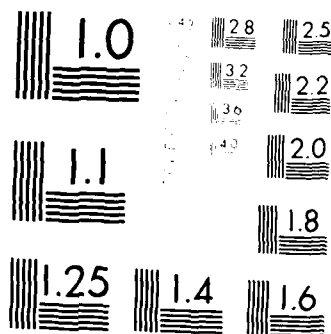
DEPLOY: AN INTERACTIVE GOAL PROGRAMMING MODEL FOR THE
RAPID DEPLOYMENT OF... (U) AIR FORCE INST OF TECH
WRIGHT-PATTERSON AFB OH SCHOOL OF ENGI... D O TATE
06 DEC 84 AFIT/GOR/OS/84D-14 F/G 15/7

2/3

UNCLASSIFIED

NL





MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A

space allowed the file to be saved for future analysis.

IDEILE: This represents the "identification" file, as it identifies the variables with a specific name and meaning. By inputting the names of the variables, the output is much easier to read, as the values of the variables are matched with the name of the variable.

GEANS: The final file to be discussed is the output file which means "GP answers." Here, the subproblem or the last priority with which the optimization concluded is specified along with the number of constraints that were considered. After this, the non-zero decision variables and their values are specified. The last portion contains the achievement of the different priorities in relation to the desired achievement. An explanation of the output is given in Chapter VI along with an example of GPANS in Table 6-6.

APPENDIX B

DEPLOY Subroutine Listing

ACFINC -- Called by CHANGE and makes the desired changes in the aircraft data file.

ACINF -- Reads aircraft data from an external user-defined formatted file.

ACINS -- Allows the user to enter aircraft data interactively and save the data in a user-defined formatted file.

APDINC -- Called by CHANGE and makes the desired changes in the APOD data file.

APDINF -- Reads APOD and miscellaneous data from an external user-defined formatted file.

APDINS -- Allows the user to enter APOD-related data interactively and save the data in a user-defined formatted file.

ARMINC -- Called by CHANGE and makes the desired changes in the deployable units data file.

ARMINF -- Reads deployable units data from an external user-defined formatted file.

ARMINS -- Allows the user to enter deployable units data interactively and save the data in a user-defined formatted file.

ELDFMX -- Builds the pointer matrix from which each variable is assigned to an aircraft, unit, or a slack or surplus variable.

CONSTR -- Builds the inequality constraints from the inputted data files.

CHANGE -- Displays the possible options to alter input data, and then calls the necessary subroutine to alter the data.

FMAT -- Formats the files for PAGP.

GOALS -- Builds the goals determined in MGOAL.

HEADER -- Prints the name of the program when the program is started.

LIMITS -- Answers the following three questions:

1. What is the largest size of cargo that can be delivered directly to the front?
2. Are there any aircraft capable of airdrop?
3. Are there any airborne units?

MGOAL -- Displays the different goal options and solicits the goals from the user.

POSTC -- Called after all the changes to the data have been made in CHANGE and asks if all the necessary data has been entered. If all the data has not been entered, the subroutine directs the user to enter the appropriate data or data files.

PRNT -- Called from CHANGE and prints on the screen the data file currently being worked on.

REALC -- Builds the equality constraints from the inputted data files.

SORTIE -- Builds the sortie constraint for the aircraft.

WRIT -- Called by CHANGE and writes the desired data file to an external file.

APPENDIX C

DEPLOY Variable Listing

Variable	Definition
AIJ(NEQ,IP)	The matrix that contains the constraints.
AVAIL(I)	The percentage of aircraft of type I that is available for the deployment.
CPI(L)	The Combat Power Index associated with period L.
DIST(N)	The distance from the US to the APOD for N=1 and from the APOD to the Front for N=2.
EAS(I,K)	The relative ease with which cargo of type K can be offloaded from aircraft type I.
FNAME	The file name.
GLS(N,J)	The matrix which contains the information about the different goals.
GT(I)	The ground time of aircraft type I.
IAB(I)	An indicator variable for airborne units.
IDFILE	The file that is created to identify the variable number with a name.
IYP	The number of units that are deployed to the front.
KARG(I,K,N)	The tonnage capacity (N=1) or passenger capacity (N=2) of type I aircraft for type K cargo.
KG	The matrix that contains the names of the different cargo types.
KTS(I)	The TAS of type I aircraft.
LL(I,N)	The lower limit of missions (N=1) or cargo type (N=2) of type I aircraft.
LMT(N)	An indicator for the largest size of cargo that can be transported via direct delivery (N=1), or an indicator variable for airborne capable aircraft (N=2) or units (N=3).

LP	The period length in days.
LPR(N)	The pointer array for the slack and surplus variables for the Rigger Constraint.
LPS(N,K,L)	A pointer matrix for the slack and surplus variables.
LPSL(L)	A pointer matrix for the slack and surplus variables.
LPU(I,M,L)	The pointer matrix for the different deployable units.
LPX(I,J,K,L)	The pointer matrix for the different aircraft types.
LU(I,N)	The upper limit of mission type (N=1) or cargo type (N=2) of type I aircraft.
MD(M)	The matrix that contains the names of the different modes of transportation.
MHE(I)	The pallet capacity of type I aircraft.
MS(J)	The matrix that contains the names of the different missions.
NAC(I)	The number of type I aircraft in the inventory.
NAME(I)	The matrix that contains the name of the different aircraft types.
NAT(I)	The amount of anti-tank power that unit type I projects.
NC(I)	The number of constraints in goal I.
NCI(I)	The combat indicator to distinguish between combat and combat support units.
NFP(I)	The amount of fire power that unit type I projects.
NCP	The number of real (equality) constraints.
NDV(NEQ)	The deviational variable to be minimized for constraint number NEQ.
NEQ	An index variable for the equations.
NFP(I)	The amount of fire power that unit type I projects.

NFT(I)	The amount of defensive frontage that type I unit can maintain.
NHA(N)	The amount of MHE prepositioned at the APOD (N=1) and at the front (N=2).
NHQ(N)	The position of the headquarter units in the unit array.
NID	The number of named variables in the file IDFILE.
NMAC	The number of different aircraft types considered in the model.
NP	The number of periods considered in the model.
NPAL	The number of pallet equivalents the ALCE unit can offload per day.
NPK(I)	The number of type I aircraft required to saturate the APOD.
NPRIT	The number of priorities in the model.
NPRK(Y)	The maximum number of fighter aircraft assigned to unit Y that can park at the APOD.
NT(NEQ)	The number of nonzero terms in equation NEQ.
NTAC(Y)	The number of fighter type aircraft assigned to unit type Y.
NTCN(I,K)	The number of tons of cargo type K per unit I.
NTP(I)	The number of ton-miles per day that unit type I can transport supplies.
NTV(I)	The number of miles that unit type I can move per day.
NUNITS	The number of units considered in the model.
NUNT(I)	The number of type I units.
NWGT(NEQ)	The weight of the deviational variable for constraint NEQ.
OFIL	The output file that contains the constraints in readable form.
RC	The capability of the riggers.
RHS(NEQ)	The right hand side of constraint NEQ.

TINTER(I)	The turntime for aircraft type I on an intertheater mission.
TINTRA(I)	The turntime for aircraft type I on an intratheater mission.
UNAME(I)	The matrix that contains the names of the different units.
UTE(I,J)	The UTE rate for type I aircraft on mission type J.

APPENDIX D

[illegible]

```

PRINT *, 'DO YOU CARE TO CHANGE THE?'
PRINT *, '      1. NUMBER OF PERIODS CONSIDERED.'
PRINT *, '      2. LENGTH OF THE PERIOD CONSIDERED.'
PRINT *, '      3. MAKE NO CHANGES.'
PRINT *
PRINT *, 'SELECT ONE OPTION.'
52 READ *, NTIME
IF (NTIME.LT.1.OR.NTIME.GT.3) THEN
    PRINT *, 'NOT A VALID RESPONSE! SELECT (1,2,3).'
    GO TO 52
END IF
IF (NTIME.EQ.1) THEN
    PRINT 12
    PRINT *, 'ENTER THE NUMBER OF PERIODS DESIRED.'
    READ *, NP
END IF
IF (NTIME.EQ.2) THEN
    PRINT 12
    PRINT *, 'ENTER THE LENGTH OF THE PERIOD IN DAYS.'
    READ *, LP
END IF
IF (NTIME.NE.3) GO TO 50
C
C++ FIND OUT WHAT KIND OF ANALYSIS WANTS TO BE DONE
C
01 PRINT 12
PRINT *, 'WHAT TYPE OF ANALYSIS IS THIS? SELECT ONE OPTION.'
PRINT *, '      1) NEW ANALYSIS.'
PRINT *, '      2) CONTINUATION OF AN EXISTING ANALYSIS.'
PRINT *, '      3) QUIT.'
2 READ *, NRES
IF (NRES.LT.1.OR.NRES.GT.3) THEN
    PRINT *, 'NOT A VALID OPTION, PLEASE ENTER (1,2,3).'
    GO TO 2
END IF
IF (NRES.EQ.3) STOP
IF (NRES.EQ.2) THEN
    CALL CHANGE
    GO TO 98
END IF
C
C++ FIND OUT HOW THE DATA WILL BE ENTERED FOR THE AIRCRAFT FILE
C
PRINT 12
PRINT *, 'HOW WILL YOU ENTER THE DATA FOR THE AIRCRAFT FILE?'
PRINT *, 'SELECT ONE OPTION.'
PRINT *, '      1) INTERACTIVELY.'
PRINT *, '      2) FROM A USER DEFINED FILE.'
3 READ *, NFILE
IF (NFILE.LT.1.OR.NFILE.GT.2) THEN
    PRINT *, 'NOT A VALID OPTION, PLEASE ENTER (1,2).'
    GO TO 3
END IF
IF (NFILE.EQ.1) THEN
    CALL ACIVE (NP)

```

```

        ELSE
            CALL APOINF
        END IF
    C
C** FIND OUT HOW THE DATA WILL BE ENTERED FOR THE APOC FILE
C
    PRINT 12
    PRINT *, 'HOW WILL YOU ENTER THE DATA FOR THE APOC FILE?'
    PRINT *, ' SELECT ONE OPTION.'
    PRINT *, '      1) INTERACTIVELY.'
    PRINT *, '      2) FROM A USER DEFINED FILE.'
13  READ *,NFILE
    IF(NFILE.LT.1.OR.NRES.GT.2) THEN
        PRINT *, 'NOT A VALID OPTION, PLEASE ENTER (1,2). '
        GO TO 13
    END IF
    IF(NFILE.EQ.1) THEN
        CALL APOINS(NMAC,NP)
    ELSE
        CALL APOINF
    END IF
C
C** FIND OUT HOW THE DATA WILL BE ENTERED FOR THE APOD FILE
C
    PRINT 13
    PRINT *, 'HOW WILL YOU ENTER THE DATA FOR THE ARMY UNIT FILE?'
    PRINT *, ' SELECT ONE OPTION.'
    PRINT *, '      1) INTERACTIVELY.'
    PRINT *, '      2) FROM A USER DEFINED FILE.'
23  READ *,NFILE
    IF(NFILE.LT.1.OR.NRES.GT.2) THEN
        PRINT *, 'NOT A VALID OPTION, PLEASE ENTER (1,2). '
        GO TO 23
    END IF
    IF(NFILE.EQ.1) THEN
        CALL ARMIN3
    ELSE
        CALL ARMINF
    END IF
C
C** CHECK TO SEE IF THERE ARE CHANGES TO BE MAKE TO THE DATA
C
33  PRINT 12
    PRINT *, 'DO YOU WISH TO MAKE ANY CHANGES TO THE ENTERED DATA?'
    PRINT *, ' ENTER Y=NO=Y.'
25  READ *,NCHECK
    IF(NCHECK.LT.1.OR.NCHECK.GT.1) THEN
        PRINT *, 'NOT A VALID RESPONSE' / 'Y=NO=Y.'
        GO TO 25
    END IF
    IF(NCHECK.EQ.1) GO TO 33
C
C** END OF THE INPUT, NOW THE SET UP BEGINS
C
    PRINT 12

```

```

C
C++ BUILD THE LPA & FU MATRIX- 52
C
      CALL BLOPMXNUNIT3
C
C++ BUILD THE REAL CONSTRAINTS- 53
C
      CALL REALC
C
C++ BUILD THE PRIORITY #1 CONSTRAINTS- 54
C
      CALL CONSTR(ATTR)
C
C++ ASK IF THIS IS AN FORCE STRUCTURE ANALYSIS- WHAT IS THE MINIMUM
C++ NUMBER OF AIRCRAFT REQUIRED TO PERFORM THE ARMY REQUIREMENT, OR
C++ IS IT AN ARMY DEPLOYMENT ANALYSIS- GIVEN A CERTAIN NUMBER OF
C++ AIRCRAFT, HOW MUCH OF THE DESIRED OBJECTIVES CAN BE MET.
C
      NQUES=5
      PRINT 12
      PRINT +, 'WHAT QUESTION DO YOU WANT ANSWERED? SELECT ONE.'
      PRINT +, '-----'
      PRINT +, '1. FORCE STRUCTURE- USE THE LEAST NUMBER AND LEAST
      PRINT +, 'COSTLY AIRCRAFT TO ACCOMPLISH THE ARMY MISSION'
      PRINT +, '2. ARMY ACCOMPLISHMENT- GIVEN A FORCE STRUCTURE,
      PRINT +, 'ACCOMPLISH THE ARMY GOALS AS BEST AS POSSIBLE.'
30  READ +,NQUES
      IF (NQUES.LT.1.OR.NQUES.GT.2) THEN
          PRINT +, 'NOT A VALID REPLY! ENTER (1 OR 2).'
          GO TO 30
      END IF
      IF (NQUES.EQ.1) THEN
          PRINT 12
          PRINT +, 'HOW DO YOU WANT TO CONSIDERED THE DIFFERENT AIRCRAFT
IT  PRINT +, '-----'
1-----
          PRINT +, '1. ALL EQUAL (DEFAULT)'
          PRINT +, '2. ACCORDING TO COSTS (USER INPUT)'
          PRINT +, 'CHOOSE ONE OPTION.'
40  READ +,NOCST
          IF (NOCST.LT.1.OR.NOCST.GT.2) THEN
              PRINT +, 'NOT A VALID REPLY! ENTER (1,2)'
              GO TO 40
          END IF
C
C++ MAKE THE NUMBER OF EACH AIRCRAFT EQUAL TO ZERO
C
      DO 43 I=1,NMAC
          MAC(I)=0
43  CONTINUE
C
C++ 43- PUT THE COSTS ARE IF COST IS TO BE CONSIDERED

```



```

1. PRINT *, 'SELECT ONE OF THE AIRCRAFT TO DELETE.'
   PRINT *, '-----'
   DO 10 I=1,NMAC
     PRINT 5, I, NAME(I)
10 CONTINUE
   PRINT *
   PRINT *, '0 IF NONE.'
24 READ *, NRES
   IF (NRES.LT.0 .OR. NRES.GT.NMAC) THEN
     PRINT *, 'NOT A VALID RESPONSE ENTER AGAIN.'
     GO TO 24
   END IF
   IF (NRES.EQ.0) RETURN

```

*** MAKE ADJUSTMENTS TO THE DATA

```

1. NAME=NAME+1
   DO 15 I=NRES,NMAC
     NAME(I) = NAME(I)+1
     NAC(I) = NAC(I)+1
     FUEL(I) = FUEL(I)+1
     DO 16 J=1,4
       RATE(I,J) = RATE(I,J)+1
15 CONTINUE
   DO 17 I=1,8
     DO 18 J=1,2
       HRS(I,J,K) = HRS(I+1,J,K)
18 CONTINUE
17 CONTINUE
   FTS(1) = FTS(1)+1
   ST(1) = ST(1)+1
   MAC(1) = MAC(1)+1
   NAC(1) = NAC(1)+1
   FUEL(1) = FUEL(1)+1
20 CONTINUE
   RETURN

```

*** INCREASE THE NUMBER OF AIRCRAFT AVAILABLE FOR THE DEPLOYMENT

```

1. PRINT *, 'SELECT THE AIRCRAFT YOU DESIRE TO INCREASE THE NUMBER.'
   PRINT *, '-----'
   DO 10 I=1,NMAC
     PRINT 5, I, NAME(I)
10 CONTINUE
   PRINT *
   PRINT *, '0 IF NONE.'
24 READ *, NRES
   IF (NRES.LT.0 .OR. NRES.GT.NMAC) THEN
     PRINT *, 'NOT A VALID RESPONSE ENTER AGAIN.'
     GO TO 24
   END IF
   IF (NRES.EQ.0) RETURN
   PRINT *
   PRINT *

```

```

      END IF
      DO 44 I=1,NMHI
        WRITE(15,3) NAME(I),NAC(I),AVAIL(I),UTE(I,1),UTE(I,2),UTE(I,3),
C       UTE(I,4),KARG(I,1,1),KARG(I,1,2),KARG(I,2,1),KARG(I,2,2),
C       KARG(I,3,1),KARG(I,3,2),KARG(I,4,2),KTS(I),STS(I),MHE(I),
C       NFR(I),NFRK(I)
        IF(NSAVE.EQ.1) THEN
          WRITE(15,3) NAME(I),NAC(I),AVAIL(I),UTE(I,1),UTE(I,2),UTE(I,3),
C       UTE(I,4),KARG(I,1,1),KARG(I,1,2),KARG(I,2,1),KARG(I,2,2),
C       KARG(I,3,1),KARG(I,3,2),KARG(I,4,2),KTS(I),STS(I),MHE(I),
C       NFR(I),NFRK(I)
        END IF
        IF (INVEQ.1) THEN
          PRINT 6,NAME(I),NAC(I),AVAIL(I),UTE(I,1),UTE(I,2),UTE(I,3),
C       UTE(I,4),KARG(I,1,1),KARG(I,1,2),KARG(I,2,1),KARG(I,2,2),
C       KARG(I,3,1),KARG(I,3,2),KARG(I,4,2),KTS(I),STS(I),MHE(I),
C       NFR(I),NFRK(I)
        END IF
44    CONTINUE
      IF(NSAVE.EQ.1) CLOSE(15)
      IF(INVEQ.1) THEN
        PRINT *
        PRINT *, 'INPUT ANY CHARACTER TO CONTINUE.'
        READ(2,*)NAME
      END IF
C
C** FORMAT STATEMENTS
C
2    FORMAT(A6)
4    FORMAT(A2)
6    FORMAT(' *****')
8    FORMAT(4I,14,5F3.1,1X,3(I3,1X,I2),14,14,F4.1,2(I3),14)
C
C
      RETURN
      END
C*****
C** 5
      SUBROUTINE ACFINC(ICH5)
C
C** THIS WILL MAKE CHANGES AS DESIRED FROM THE CHANGE SUBROUTINE
C** FOR THE AIRCRAFT DATA
C
      CHARACTER NAME(10)*5,NAME(11)*14,NAME(13)*14,FNAME,MS(4)*5,
C       KG(5)*4,MD(3)*5
      COMMON /CF NAC,AC(10),AVAIL(10),UTE(10,4),KARG(10,5,2),KTS(10),
C       STS(10),MHE(10),NFR(10),NFRK(10)
      COMMON /CF NAME,NAME,NAME,MS,KG,MD
C
C** GO TO THE CORRECT STATION FOR THE CHANGE
C
      PRINT *
      GO TO (20,30,40,50) ICH5
C
C** DELETE 47 AIRCRAFT FROM THE DATA LIST

```



```

READ *,UTE(1,1)
IF(UTE(1,1).LT.0.OR.UTE(1,1).GT.24) THEN
  PRINT *,'NOT A VALID UTE RATE. (0 < UTE < 24) .'
  GO TO 45
END IF
46 PRINT *,'      INTERTHEATER TO THE APOD.'
READ *,UTE(1,2)
IF(UTE(1,2).LT.0.OR.UTE(1,2).GT.24) THEN
  PRINT *,'NOT A VALID UTE RATE. (0 < UTE < 24) .'
  GO TO 46
END IF
47 PRINT *,'      INTRATHEATER TO THE FRONT.'
READ *,UTE(1,3)
IF(UTE(1,3).LT.0.OR.UTE(1,3).GT.24) THEN
  PRINT *,'NOT A VALID UTE RATE. (0 < UTE < 24) .'
  GO TO 47
END IF
48 PRINT *,'      AIRDROP TO THE FRONT.'
READ *,UTE(1,4)
IF(UTE(1,4).LT.0.OR.UTE(1,4).GT.24) THEN
  PRINT *,'NOT A VALID UTE RATE. (0 < UTE < 24) .'
  GO TO 48
END IF
PRINT *
PRINT *,'INPUT FIRST THE TOYRAGE (REAL) AND THEN THE NUMBER OF PA
CINTEGERS.'
PRINT *,'FOR EACH TYPE OF CARB ALONG WITH #PA .'
PRINT *,'      OUTSIDE:
READ *,KARG(1,1,1),KARG(1,1,2)
PRINT *,'      OVERSIDE:
READ *,KARG(1,2,1),KARG(1,2,2)
PRINT *,'      BULK-SIDE:
READ *,KARG(1,3,1),KARG(1,3,2)
C
C++ BULK SAME AS SUPPLIES
C
      KARG(1,5,1)=KARG(1,3,1)
      KARG(1,5,2)=KARG(1,3,2)
PRINT *
PRINT *,'INPUT THE CAPACITY OF PAK ONLY (INTEGER) ON THE (NAME)
READ *,KARG(1,4,2)
PRINT *
PRINT *,'INPUT THE AVERAGE CRUISE TIME (INTEGER) FOR THE (NAME)
READ *,KTS(1)
PRINT *
PRINT *,'INPUT THE GROUND TIME TO UNLOAD AND REFUEL (REAL)
READ *,GT(1)
PRINT *
PRINT *,'INPUT THE NUMBER OF PALLETS (INTEGER) ON A (NAME)
READ *,MHE(1)
PRINT *
PRINT *,'WHAT IS THE FIRST PERIOD THE AC WILL BE AVAILABLE?
29 READ *,NEP(1)
      IF(NEP(1).GT.MP) THEN
        PRINT *,'ARE YOU SURE THAT IT IS NOT UNTIL THE

```

```

      DOOR CONTINUE.
      PRINT *, 'DO YOU WANT TO CONTINUE? (1=Y, 0=N)'
10  READ *, NRES
      IF (NRES.GT.1.OR.NRES.LT.0) THEN
          PRINT *, 'NOT A VALID REPLY, PLEASE ENTER (1=Y/0=N)'
          GO TO 10
      END IF
      IF (NRES.EQ.0) STOP
C
C** ASK IF DESIRED TO SAVE DATA IN A FILE
C
      PRINT &
      PRINT *, 'DO YOU WANT THE DATA SAVED IN AN EXTERNAL FILE? (1=Y, 0=N)'
12  READ *, NSAVE
      IF (NSAVE.GT.1.OR.NSAVE.LT.0) THEN
          PRINT *, 'NOT A VALID REPLY, PLEASE ENTER (1=Y/0=N)'
          GO TO 12
      END IF
      IF (NSAVE.EQ.1) THEN
          PRINT *, 'ENTER THE NAME OF THE FILE YOU WANT THE DATA SAVED (6
CCHARACTERS MAX)'
          READ 2, FNAME
          OPEN(15, FILE=FNAME)
          REWIND(15)
          END IF
C
C** START THE INPUT PROCESS
C
      I=0
      PRINT &
      PRINT *, 'I AM READY TO ACCEPT YOUR INPUTS, AND WILL PROMPT YOU FOR
C THE DATA.'
11  PRINT *
      PRINT *, '-----'
      PRINT *, '---          ENTERING AIRCRAFT DATA          ---'
      PRINT *, '-----'
      PRINT *
      PRINT *, 'INPUT THE AIRCRAFT NAME. * IF FINISHED INPUT 0.'
      READ 2, FIRST
      IF (FIRST.EQ.'0') GO TO 20
      I=I+1
      NAME(I)=FIRST
      PRINT *
      PRINT *, 'INPUT THE NUMBER OF ', NAME(I), ' IN THE INVENTORY.'
      READ *, NAC(I)
      PRINT *
      PRINT *, 'INPUT THE AVAILABILITY (REAL) OF THE ', NAME(I), ' (0..1)'
01  READ *, AVAIL(I)
      IF (AVAIL(I).LT.0.OR.AVAIL(I).GT.1) THEN
          PRINT *, 'NOT A VALID REPLY (0 < AVAIL < 1)'
          GO TO 31
      END IF
      PRINT *
      PRINT *, 'INPUT THE UTE RATE (REAL) FOR THE FOLLOWING MISSION:'
43  PRINT *, '      DIRECT DELIVERY.'

```

```

5      CONTINUE
      CLOSE (10)
      IF (MDU.EQ.1) THEN
          PRINT *
          PRINT *, 'ENTER ANY CHARACTER TO CONTINUE.'
          READ 2, FNAME
      END IF

C
C++  FORMAT STATEMENTS
C
2      FORMAT(A6)
4      FORMAT(A77)
3      FORMAT(A6,I4,5F5.1,1X,3(I3,1X,I2),I4,I4,F4.1,2(I3),I4)
5      FORMAT(////////////////////)
C
C
      RETURN
      END

C*****
C++  S3
      SUBROUTINE ACINS(NP)
C
C++  CREATES THE AIRCRAFT FILE FOR THE ANALYSIS
C
      CHARACTER NAME(10)*6,UNAME(11)*14,GNAME(3)*14,MS(4)*5,KG(5)*4,
C      MD(3)*5
      CHARACTER FIRST*6,FNAME*6,TITLE*77
      COMMON/ACF/NMAC,NAC(10),AVAIL(10),UTE(10,4),KARG(10,5,2),KTS(10),
C      GT(10),MHE(10),NFP(10),NPRK(10)
      COMMON/CHR/NAME,UNAME,GNAME,MS,KG,MD

C
C
      PRINT *
      PRINT *, 'THE FOLLOWING DATA IS NEEDED FOR EACH AIRCRAFT.'
      PRINT *, '-----'
      PRINT *, 'AIRCRAFT NAME          ---          MAX 6 CHARACTERS'
      PRINT *, 'NUMBER OF AIRCRAFT      ---          < 1,000'
      PRINT *, 'AVAILABILITY OF THE AC     ---          BETWEEN (0...10)'
      PRINT *, 'DAILY UTE RATE FOR THE FOLLOWING MISSIONS:'
      PRINT *, '    DIRECT DELIVERY          ---          < 24.0'
      PRINT *, '    APOD DELIVERY              ---          < 24.0'
      PRINT *, '    INTRATHEATER                ---          < 24.0'
      PRINT *, '    AIRDROP                     ---          < 24.0'
      PRINT *, 'TONNAGE OF THE FOLLOWING CARGO ALONG WITH PAX:'
      PRINT *, '    OUTSIZE                     ---          < 1,000/100'
      PRINT *, '    OVERSIZE                    ---          < 1,000/100'
      PRINT *, '    BULK SIZE                   ---          < 1,000/100'
      PRINT *, 'CAPACITY OF PAX ONLY CARGO---          < 1,000'
      PRINT *, 'TAS OF THE AIRCRAFT(KTS)      ---          < 1,000'
      PRINT *, 'GROUND TIME TO UNLOAD(HRS)---          < 100.0'
      PRINT *, 'MAX NUMBER OF PALLETS         ---          < 1,000'
      PRINT *, 'FIRST PERIOD AC IS AVAILABLE---          < # PERIODS'
      PRINT *, 'NUMBER OF AC PARK AT APOD ---          < 1,000'
      PRINT *
      PRINT *, 'IF YOU DO NOT HAVE ALL THE ABOVE DATA, YOU MAY TERMINATE'

```

```

C
C++ READS IN AN EXTERNAL FILE THAT CONTAINS
C++ INFORMATION ABOUT THE STRATEGIC AIRCRAFT
C
      CHARACTER NAME(10)*5,HDG*77,FNAME*6,UNAME(11)*14,GNAME(3)*14,
C      MS(4)*5,KG(5)*4,MD(3)*5
      COMMON/ACF/NMAC,NAC(10),AVAIL(10),UTE(10,4),KARG(10,5,2),KTS(10),
C      GT(10),MHE(10),NFP(10),NPRK(10)
      COMMON/CHR/NAME,UNAME,GNAME,MS,KG,MD
C
C++ FIND THE FILE NAME WITH THE DATA
C
      PRINT 9
      PRINT *, 'ENTER THE FILE NAME (6 CHAR MAX) WITH THE AIRCRAFT DATA.'
      READ 2,FNAME
      OPEN(10,FILE=FNAME)
      REWIND(10)
C
C++ ASK IF THE FILE IS TO BE VIEWED
C
      PRINT 9
      PRINT *, 'DO YOU WANT TO SEE THE INPUT FILE? (1=Y/0=N)'
21  READ *,IVU
      IF(IVU.GT.1.OR.IVU.LT.0) THEN
          PRINT *, 'NOT A VALID REPLY, PLEASE ENTER (1=Y/0=N).'
          GO TO 21
      END IF
C
C++ READ IN THE DATA FILE
C
99  READ (10,*)NMAC
      READ (10,4)HDG
      WRITE (13,*)NMAC
      WRITE (13,4)HDG
      IF(IVU.EQ.1) THEN
          PRINT 9
          PRINT *,NMAC
          PRINT 4,HDG
      END IF
5   DO 6 I=1,NMAC
      READ(10,8)NAME(I),NAC(I),AVAIL(I),UTE(I,1),UTE(I,2),UTE(I,3),
C      UTE(I,4),KARG(I,1,1),KARG(I,1,2),KARG(I,2,1),KARG(I,2,2),
C      KARG(I,3,1),KARG(I,3,2),KARG(I,4,2),KTS(I),GT(I),MHE(I),
C      NFP(I),NPRK(I)
      KARG(I,5,1)=KARG(I,3,1)
      KARG(I,5,2)=KARG(I,3,2)
      WRITE(13,8)NAME(I),NAC(I),AVAIL(I),UTE(I,1),UTE(I,2),UTE(I,3),
C      UTE(I,4),KARG(I,1,1),KARG(I,1,2),KARG(I,2,1),KARG(I,2,2),
C      KARG(I,3,1),KARG(I,3,2),KARG(I,4,2),KTS(I),GT(I),MHE(I),
C      NFP(I),NPRK(I)
      IF(IVU.EQ.0) GO TO 6
      PRINT 8,NAME(I),NAC(I),AVAIL(I),UTE(I,1),UTE(I,2),UTE(I,3),
C      UTE(I,4),KARG(I,1,1),KARG(I,1,2),KARG(I,2,1),KARG(I,2,2),
C      KARG(I,3,1),KARG(I,3,2),KARG(I,4,2),KTS(I),GT(I),MHE(I),
C      NFP(I),NPRK(I)

```

```

        WRITE(13,81) I, IP, LPR(1)
        IF(LANS.EQ.0) GO TO 80
        PRINT 81, I, IP, LPR(1)
80      CONTINUE
81      FORMAT('LPR(', I2, ')=' , 2I6)
        WRITE(13,*) 'IP=' , IP
        CLOSE(17)
        RETURN
        END
C*****
C 810
        SUBROUTINE LIMITS
C
C      THIS DETERMINES THE LIMITS OF
C      LMT(1): LARGEST TYPE OF CARGO THAT A DIRECT DELIVERY AC
C              CAN DELIVER
C      LMT(2): 1 IF THERE ARE ANY AIRDROP CAPABLE AC, ELSE=0
C      LMT(3): 1 IF THERE ARE ANY AIRBORNE UNITS, ELSE=0
C
        COMMON/ACF/NMAC,NAC(10),AVAIL(10),UTE(10,4),KARG(10,5,2),KTS(10),
C      GT(10),MHE(10),NFP(10),NPRK(10)
        COMMON/ARM/NUNITS,NUNT(10),NTON(11,5),NFP(10),NAT(10),NFT(10),
C      NTAC(10),NPK(10),NTP(11),NTV(11),IAB(10),NCI(10),IYP,NHQ(2)
        COMMON/TAB/NP,LP,IP,NCR,NEQ,NPRIT,LL(10,2),LU(10,2),LPX(10,4,5,4),
C      LPU(10,3,4),AIJ(150,300),RHS(150),LPS(3,5,4),LPSL(4),LPR(10),
C      NT(150),LMT(3),NOC(10),NWGT(150),NDV(150),NID
C
        ID=0
        DO 10 I=1,NMAC
            IF(LL(I,1).GT.1) GO TO 12
            IF(ID.EQ.0) THEN
                ID=1
                LMT(1)=LL(I,2)
                GO TO 12
            END IF
            IF(LL(I,2).LT.LL(ID,2)) THEN
                ID=1
                LMT(1)=LL(I,2)
            END IF
12      IF(LU(I,1).EQ.4) LMT(2)=1
10      CONTINUE
C
        DO 20 I=1,NUNITS
            IF(IAB(I).EQ.0) GO TO 20
            LMT(3)=1
20      CONTINUE
C
        WRITE(13,22) LMT(1), LMT(2), LMT(3)
        FORMAT('LMT(1)=' , 3I3)
C
        RETURN
        END
C*****
C** 82
        SUBROUTINE ACINF

```

```

        DO 60 N=1,NP
            IP=IP+1
            LPU(1,M,N)=IP
            WRITE(13,62) LPU(1,M,N),UNAME(1),MD(M),N
            WRITE(17,62) LPU(1,M,N),UNAME(1),MD(M),N
            IF(LANS.EQ.0) GO TO 60
            PRINT 62,LPU(1,M,N),UNAME(1),MD(M),N
60      CONTINUE
53      CONTINUE
56      CONTINUE
62      FORMAT(14,3X,A14,2X,A5,I3)
        MID=IP
C
C++ BUILD THE LPS-MATRIX
C
C++ FOR THE SHIPMENTS- DIRECT,APOD,ABN
C++ FOR THE SUPPLIES(K=5)-THTR,FRONT,APOD
        DO 64 I=1,3
C++ FOR OUT, OVER, BULK, PAX, & SUPPLIES
        DO 65 K=1,5
C
C++ DO NOT INCLUDE (1..4) IF NO DIRECT DELIVERY NOR AIRDROP CAPABILITY
C
        IF(X.NE.5) THEN
            IF(OLMT(1).LT.1.AND.1.EQ.1) GO TO 65
            IF(OLMT(2).E1.0.AND.1.EQ.3) GO TO 65
        END IF
C++ FOR EACH PERIOD
        DO 66 N=1,NP
            IP=IP+1
            LPS(1,K,N)=IP
            WRITE(13,67) I,K,N,IP,LPS(1,K,N)
            IF(LANS.EQ.0) GO TO 66
            PRINT 67,I,K,N,IP,LPS(1,K,N)
66      CONTINUE
63      CONTINUE
64      CONTINUE
67      FORMAT('LPS(',I2,',')=',2I6)
C
C++ BUILD THE LPSL-MATRIX
C
        DO 70 I=1,NP
            IP=IP+1
            LPSL(1)=IP
            WRITE(13,71) I,IP,LPSL(1)
            IF(LANS.EQ.0) GO TO 70
            PRINT 71,I,IP,LPSL(1)
70      CONTINUE
71      FORMAT('LPSL(',I2,',')=',2I6)
C
C++ BUILD THE LPR-MATRIX, FOR RIGGER SLACK & SURPLUS
C
        DO 80 I=1,NP
            IP=IP+1
            LPR(1)=IP

```

```

      IP=0
C++ FOR EACH TYPE OF AIRCRAFT
      DO 42 I=1,NMAC
C++ FOR EACH TYPE OF MISSION DELIVERED TO:
C++ 1=DROT, 2=APOD, 3=INTRA 4=ABRN
      DO 43 J=LL(I,1),LU(I,1)
C++ FOR EACH TYPE OF CARGO:
C++ 1=OUT, 2=OVER, 3=BULK,SUPPLIES, 4=PAK
      DO 44 K=LL(I,2),LU(I,2)
C++ FOR EACH AVAILABLE PERIOD
      DO 45 L=NFR(I),NP
        IP=IP+1
        LPX(I,J,K,L)=IP
        WRITE(13,17)LPX(I,J,K,L),NAME(I),MS(J),KG(K),L
        WRITE(17,17)LPX(I,J,K,L),NAME(I),MS(J),KG(K),L
        IF(LANG.EQ.0) GO TO 46
        PRINT 17,LPX(I,J,K,L),NAME(I),MS(J),KG(K),L
46      IF(K.EQ.3) THEN
          IP=IP+1
          LPX(I,J,5,L)=IP
        ELSE
          GO TO 45
        END IF
        WRITE(13,17)LPX(I,J,5,L),NAME(I),MS(J),KG(5),L
        WRITE(17,17)LPX(I,J,5,L),NAME(I),MS(J),KG(5),L
        IF(LANG.EQ.0) GO TO 45
        PRINT 17,LPX(I,J,5,L),NAME(I),MS(J),KG(5),L
17      FORMAT(14,3X,A6,2X,A5,2X,A4,13)
45      CONTINUE
44      CONTINUE
43      CONTINUE
42      CONTINUE
C
C++ BUILD THE LFD-MATRIX
C
C++ FOR EACH UNIT
C
      DO 56 I=1,NUNITS+1
C
C++ MODE OF DELIVERY: 1=FRONT, 2= APOD-FRONT WALK, AND 3=APOD-FLY
C
      DO 53 M=1,3
C
C++ ELIMINATE UNITS THAT REMAIN AT THE APOD FOR M=1,3
C++ DIRECT DELIVERY (M=1) IF NO DIRECT DELIVERY TYPE AC
C++ AIRDROP (M=1) IF NO AEN UNITS, OR AIRDROP TYPE AC
C++ ALSO- AEN UNITS ARE DEPLOYED ONLY BY AIRDROP
C
      IF(LGT.1X.F.AND.M.NE.3.AND.1.LE.NUNITS) GO TO 58
      IF(M.NE.1.AND.1AB(I).EQ.1) GO TO 59
      IF(LMT(1).LT.1.AND.1AB(1).EQ.0.AND.M.EQ.1) GO TO 58
      IF(LMT(2).EQ.0.AND.M.EQ.1) GO TO 58
      IF(LMT(3).EQ.0.AND.M.EQ.1) GO TO 58
      IF(1AB(1).EQ.1.AND.M.NE.1) GO TO 53
C++ FOR EACH PERIOD

```

```

22      CONTINUE
C**    FIND THE UPPER LIMIT
      IF (UTEX(I,4).NE.0.0) GO TO 24
      IF (UTEX(I,3).NE.0.0) THEN
          LUX(I,1)=3
          GO TO 24
      END IF
      IF (UTEX(I,2).NE.0.0) THEN
          LUX(I,1)=2
          GO TO 24
      END IF
      LUX(I,1)=1
24      CONTINUE
      IF (LL(I,1).LE.LUX(I,1)) GO TO 26
      PRINT *, 'ERROR IN LL AND LU (I,1) '
C**    FOR 2=KARGO
C**    FIND THE LOWER LIMIT
26      LL(I,2)=1
      LUX(I,2)=4
      IF (KARG(I,1,1).NE.0.0) GO TO 29
      IF (KARG(I,2,1).NE.0.0) THEN
          LL(I,2)=2
          GO TO 29
      END IF
      IF (KARG(I,3,1).NE.0.0) THEN
          LL(I,2)=3
          GO TO 29
      END IF
          LL(I,2)=4
          LUX(I,2)=4
          GO TO 30
29      CONTINUE
C**    FIND THE UPPER LIMIT
      IF (KARG(I,4,2).NE.0.0) GO TO 30
      IF (KARG(I,3,1).NE.0.0) THEN
          LUX(I,2)=3
          GO TO 30
      END IF
      IF (KARG(I,2,1).NE.0.0) THEN
          LUX(I,2)=2
          GO TO 30
      END IF
          LUX(I,2)=1
30      CONTINUE
C**    CHECK THE LIMITS
      IF (LL(I,2).LE.LUX(I,2)) GO TO 20
      PRINT *, 'ERROR IN LL AND LU (I,2) '
20      CONTINUE
C
C**    FIND THE LIMITS
C
      CALL LIMITS
C
C**    BUILD THE LPX-MATRIX
C

```

```

C
MS(1)='DIRECT'
MS(2)='APOD'
MS(3)='INTRA'
MS(4)='AIRBN'
KG(1)='OUT'
KG(2)='OVER'
KG(3)='BULK'
KG(4)='PAK'
KG(5)='SUPP'
MD(1)='DIRECT'
MD(2)='APOD'
MD(3)='INTRA'
RETURN
END

C*****
C** S-BLD
SUBROUTINE BLDPMX
C
C** THIS BUILDS THE POINTER MATRIX
C
CHARACTER NAME(10)*6,UNAME(11)*14,GNAME(3)*14
CHARACTER MS(4)*5,KG(5)*4,MD(3)*5
COMMON/ACF/IMAC,NACK(10),AVAIL(10),UTE(10,4),KARG(10,5,2),KTB(10,4),
C GT(10),MHE(10),NPR(10),NPRK(10)
COMMON/APD/DIST(2),TINTER(10),TINTRK(10),EAS(10,4),OPK(4),
C NHAT(2),NPAL,RC,GLS(9,3)
COMMON/ARM/NUNITS,NUNT(10),NTGN(11,5),NPR(10),NAT(10),NFT(10),
C NTAC(10),NPK(10),NTP(11),NTUX(11),IAB(10),NCI(10),IYP,NHQ(2)
COMMON/TAB/NP,LP,IP,NCR,NEQ,NPRIT,LL(10,2),LUX(10,2),LPK(10,4,5,4),
C LPK(10,3,4),AIX(150,300),RHS(150),LPS(3,5,4),LPSL(4),LPR(10),
C NT(150),LMT(3),NCK(10),NWGT(150),NDV(150),NID
COMMON/CHR/NAME,UNAME,GNAME,MS,KG,MD
C** ASK IF POINTER IS TO BE ECHOED
+ PRINT *, 'ECHO THE POINTER? (Y=1,N=0)'
+ READ *,LANG
LANG=0
C** FIND THE DIFFERENT AIRCRAFT LOWER AND UPPER LIMITS
DO 20 I=1,NMAC
C** FOR 1=MISSION
LL(I,1)=1
LUX(I,1)=4
C** FIND THE LOWER LIMIT
IF (UTE(I,1).NE.0.0) GO TO 22
IF (UTE(I,2).NE.0.0) THEN
LL(I,1)=2
GO TO 22
END IF
IF (UTE(I,3).NE.0.0) THEN
LL(I,1)=3
GO TO 22
END IF
LL(I,1)=4
LUX(I,1)=4
GO TO 24

```

```

24      CONTINUE
      RH5(I)=0.0
      NTK(I)=0
      NDV(I)=3
      NWGT(I)=1
22      CONTINUE
C++    ZEROIZE THE KARG AND TON MATRICES
      DO 30 I=1,10
        DO 32 J=1,5
          NTON(I,J)=0
          DO 33 K=1,2
            KARG(I,J,K)=0
33          CONTINUE
32          CONTINUE
        DO 34 J=1,4
          DO 36 M=1,3
            LPU(I,M,J)=0
36          CONTINUE
          UTE(I,J)=0.0
          DO 38 K=1,4
            DO 40 L=1,4
              LPU(I,J,K,L)=0
40            CONTINUE
38          CONTINUE
34          CONTINUE
          NACK(I)=0
          KTS(I)=0
          MHE(I)=0
          NFR(I)=0
          NFRK(I)=0
          AVAIL(I)=0.0
          GT(I)=0.0
          TINTER(I)=0.0
          TINTRA(I)=0.0
          NUNT(I)=0
          NTAC(I)=0
          NPK(I)=0
          NFP(I)=0
          NAT(I)=0
          NFT(I)=0
          NTP(I)=0
          NTV(I)=0
          NCI(I)=0
          IAB(I)=0
          NC(I)=1
          NHQ(1)=0
          NHQ(2)=0
30      CONTINUE
      DO 42 I=1,9
        DO 44 J=1,3
          GLS(I,J)=0.0
44        CONTINUE
42      CONTINUE
C
C++    ESTABLISH THE NAMES OF THE DIFFERENT MISSIONS, KARGO, AND MODES

```

```

C      PRINT *, BUILD THE FILES FOR PAGE=1,2=0
      READ *, LANS
      IF(LANS.EQ.0) GO TO 99
C** BUILD THE FILES FOR PAGE INPUT- 311
      CALL FMAT
      12  FORMAT(////////////////////)
      99  END
C-----
C*****
C** S0
      SUBROUTINE INITIAL
C
C** INITIALIZES THE PARAMETERS AND MATRICES
C
      CHARACTER NAME(10)*6, UNAME(11)*14, GNAME(3)*14
      CHARACTER MS(4)*5, KG(5)*4, MD(3)*5
      COMMON/ACF/NMAC, NAC(10), AVAIL(10), UTE(10,4), KARG(10,5,2), KTB(10),
C      GT(10), MHE(10), NFP(10), NPRK(10)
      COMMON/APD/DIST(2), TINTER(10), TINTRA(10), EAS(10,4), CPI(4),
C      NHA(2), NPAL, RC, GLS(9,3)
      COMMON/ARM/NUNITS, NUNT(10), NTGN(11,5), NFP(10), NAT(10), NET(10),
C      NTHC(10), NPRK(10), NTP(11), NTMX(11), IAB(10), NCX(10), IYP, NHQ(2)
      COMMON/TAB/NP, LP, IP, NCR, NEQ, NPRIT, LL(10,2), LU(10,3), LPR(10,4,5,4),
C      LPL(10,3,4), AIJ(150,300), RHS(150), LPS(3,5,4), LPSL(4), LPRK(10),
C      NT(150), LMT(3), NC(10), NWGT(150), NDV(150), NID
      COMMON/CHR/NAME, UNAME, GNAME, MS, KG, MD
C-----
C** INITIALIZE THE TERMS
C** NORMAL PARAMETERS
      NP=4
      LP=5
      NPRIT=1
      DO 10 I=1,10
        DO 12 K=1,4
          EAS(I,K)=1.0
        12  CONTINUE
      10  CONTINUE
C** ZEROIZE THE INTEGERS
      NMAC=0
      NUNITS=0
      IP=0
      NEQ=0
      NCR=0
      IYP=0
      NHA(1)=0
      NHA(2)=0
      NPAL=0
      DO 18 I=1,3
        LMT(I)=0
      18  CONTINUE
C** ZEROIZE THE AIJ AND RHS-MATRIX
      DO 22 I=1,150
        DO 24 J=1,300
          AIJ(I,J)=0.0
        24  CONTINUE
      22  CONTINUE

```

```

        PRINT *, 'ENTER THE DESIRED NUMBER OF ', NAME(NRES)
        READ *, NMAC(NRES)
        RETURN
C
C** CHANGE THE AVAILABILITY OF AN AIRCRAFT
C
40  PRINT *, 'SELECT THE AIRCRAFT YOU DESIRE TO CHANGE THE AVAILABILITY
    1 OF.'
    PRINT *, '-----'
    1-----
    DO 42 I=1,NMAC
        PRINT 7,I,NAME(I),AVAIL(I)
42  CONTINUE
    PRINT *
    PRINT *, '0 IF NONE.'
44  READ *, NRES
    IF(NRES.LT.0.OR.NRES.GT.NMAC) THEN
        PRINT *, 'NOT A VALID RESPONSE! ENTER AGAIN.'
        GO TO 44
    END IF
    IF(NRES.EQ.0) RETURN
    PRINT *
    PRINT *
    PRINT *, 'ENTER THE DESIRED AVAILABILITY (0..1) OF THE ', NAME(NRES)
46  READ *, RES
    IF(RES.LT.0.OR.RES.GT.1) THEN
        PRINT *, 'NOT A VALID RESPONSE! ENTER AGAIN (0..1).'
        GO TO 46
    END IF
    AVAIL(NRES)=RES
    RETURN
C
C** CHANGE THE GROUND TIME OF AN AIRCRAFT
C
50  PRINT *, 'SELECT THE AIRCRAFT YOU DESIRE TO CHANGE THE GROUND TIME'
    PRINT *, '-----'
    DO 52 I=1,NMAC
        PRINT 7,I,NAME(I),GT(I)
52  CONTINUE
    PRINT *
    PRINT *, '0 IF NONE.'
54  READ *, NRES
    IF(NRES.LT.0.OR.NRES.GT.NMAC) THEN
        PRINT *, 'NOT A VALID RESPONSE! ENTER AGAIN.'
        GO TO 54
    END IF
    IF(NRES.EQ.0) RETURN
    PRINT *
    PRINT *
    PRINT *, 'ENTER THE DESIRED GROUND TIME OF THE ', NAME(NRES)
    READ *, GT(NRES)
    RETURN
C
C** FORMAT STATEMENT
C

```

```

5  FORMAT(' ',I2,'/ ',A6)
6  FORMAT(' ',I2,'/ ',A6,' - ',I2)
7  FORMAT(' ',I2,'/ ',A6,' - ',F3.1)
8  FORMAT('////////////////')
C
C
      END
C*****
C**  S4
      SUBROUTINE APDINF
C
C**  READS IN AN EXTERNAL FILE THAT CONTAINS
C**  INFORMATION ABOUT THE STRATEGIC AIRCRAFT
C
      CHARACTER HDG(4)*77,SUBH(20)*14,FNAME*6
      CHARACTER NAME(10)*6,UNAME(11)*14,GNAME(3)*14
      CHARACTER MS(4)*5,KG(5)*4,MD(3)*5
      COMMON/ACF/NMAC,NAC(10),AVAIL(10),UTE(10,4),KARG(10,5,2),KTS(10),
C      GT(10),MHE(10),NFP(10),NPRK(10)
      COMMON/APD/DIST(2),TINTER(10),TINTRA(10),EAS(10,4),CPI(4),
C      NHA(2),NPAL,RC,GLS(9,3)
      COMMON/ARM/NUNITS,NUNT(10),NTON(11,5),NFP(10),NAT(10),NFT(10),
C      NTAC(10),NFK(10),NTP(10),NTUX(10),IAB(10),NCI(10),IYP,NHG(2)
      COMMON/TAB/NP,LP,IP,NCR,NEQ,NPRIT,LL(10,2),LJ(10,2),LPX(10,4,5,4),
C      LPU(10,3,4),AIJ(150,300),RHS(150),LPS(3,5,4),LPSL(4),LPR(10),
C      NT(150),LMT(3),NC(10),NWGT(150),NDV(150),NID
      COMMON/CHR/NAME,UNAME,GNAME,MS,KG,MD
C
C**  FIND THE FILE NAME WITH THE DATA
C
      PRINT 9
      PRINT *, 'ENTER THE FILE NAME (6 CHAR MAX) WITH THE APD DATA.'
      READ 2,FNAME
      OPEN(11,FILE=FNAME)
      REWIND(11)
C
C**  ASK IF THE FILE IS TO BE VIEWED
C
      PRINT 9
      PRINT *, 'DO YOU WANT TO SEE THE INPUT FILE? (1=Y/0=N)'
21  READ *,IWD
      IF(IWD.GT.1.OR.IWD.LT.0) THEN
          PRINT *, 'NOT A VALID REPLY, PLEASE ENTER (1=Y/0=N).'
          GO TO 21
      END IF
C
C**  READ IN THE DATA FILE BEGINNING WITH THE FIRST MAJOR HEADING
C**  AND PRINT IF TO BE VIEWED
C
99  READ (11,4)HDG(1)
      WRITE (13,4)HDG(1)
      IF (IWD.EQ.1) THEN
          PRINT 9
          PRINT 4,HDG(1)
      END IF

```

```

C
C++ READ IN THE DISTANCES
C
    DO 12 I=1,2
        READ(11,6)SUBH(I),DIST(I)
        WRITE(13,6)SUBH(I),DIST(I)
        IF(IVU.EQ.1) PRINT 6,SUBH(I),DIST(I)
12    CONTINUE
C
C++ READ IN THE INTER AND INTRA TURN-TIMES
C
    READ(11,4)HDG(2)
    WRITE(13,4)HDG(2)
    IF(IVU.EQ.1) PRINT 4,HDG(2)
    DO 14 I=1,NMAC
        READ(11,8)SUBH(I+2),TINTER(I),TINTRA(I),
C          EAS(I,1),EAS(I,2),EAS(I,3)
        WRITE(13,8)SUBH(I+2),TINTER(I),TINTRA(I),
C          EAS(I,1),EAS(I,2),EAS(I,3)
        IF(IVU.EQ.1) PRINT 8,SUBH(I+2),TINTER(I),TINTRA(I),
C          EAS(I,1),EAS(I,2),EAS(I,3)
14    CONTINUE
C
C++ READ IN THE COMBAT POWER INDEXES FOR UNITS CLOSING AT THE FRONT
C
    READ(11,4)HDG(3)
    WRITE(13,4)HDG(3)
    IF(IVU.EQ.1) PRINT 4,HDG(3)
    J=NMAC+2
    DO 16 I=1,NP
        READ(11,17)SUBH(I+J),CPI(I)
        WRITE(13,17)SUBH(I+J),CPI(I)
        IF(IVU.EQ.1) PRINT 17,SUBH(I+J),CPI(I)
16    CONTINUE
C
C++ NUMBER OF MHE PREPOSITIONED
C
    READ(11,4)HDG(4)
    WRITE(13,4)HDG(4)
    IF(IVU.EQ.1) PRINT 4,HDG(4)
    J=NMAC+NP+2
    DO 18 I=1,2
        READ(11,19)SUBH(I+J),NHA(I)
        WRITE(13,19)SUBH(I+J),NHA(I)
        IF(IVU.EQ.1) PRINT 19,SUBH(I+J),NHA(I)
18    CONTINUE
C
C++ THE NUMBER OF PALLETS AN ALCE UNIT CAN DOWNLOAD
C
    READ(11,19)SUBH(19),NPAL
    WRITE(13,19)SUBH(19),NPAL
    IF(IVU.EQ.1) PRINT 19,SUBH(19),NPAL
C
C++ RIGGER CAPABILITY PER DAY DETERMINED
C

```

```

      READ(11,17)SUBH(20),RC
      WRITE(13,17)SUBH(20),RC
      IF(IVU.EQ.1) THEN
        PRINT 17,SUBH(20),RC
        PRINT *
        PRINT *, 'ENTER ANY CHARACTER TO CONTINUE.'
        READ 2,FNAME
      END IF

C
C**  FORMAT STATEMENTS
C
2     FORMAT(A6)
4     FORMAT(A77)
6     FORMAT(A14,F6.0)
8     FORMAT(A14,5F6.1)
9     FORMAT(////////////////////)
17    FORMAT(A14,F6.1)
19    FORMAT(A14,I6)
C
C
      RETURN
      END

C+++++*****
C** S
      SUBROUTINE APDINS(NMAC,NP)
C
C**  PROMPTS INFORMATION FROM THE USER CONCERNING THE APOD
C**  AND MISCELLANEOUS DATA
C
      CHARACTER HDG(4)*77,SUBH(20)*14,FNAME*6
      CHARACTER NAME(10)*6,UNAME(11)*14,GNAME(3)*14
      CHARACTER MS(4)*5,KG(5)*4,MD(3)*5
      COMMON/APD/DIST(2),TINTER(10),TINTRA(10),EAS(10,4),CFI(4),
C      NHA(2),NPAL,RC,GLS(9,3)
      COMMON/CHR/NAME,UNAME,GNAME,MS,KG,MD
C
C**  PRINT OUT THE INFORMATION NEEDED FOR THIS FILE
C
      PRINT 5
      PRINT *, '      THE FOLLOWING DATA IS NEEDED FOR THIS FILE:'
      PRINT *, '-----'
      PRINT *, 'DISTANCES IN KM FROM'
      PRINT *, '      US TO APOD          ---   < 100,000'
      PRINT *, '      APOD TO FRONT        ---   < 100,000'
      PRINT *, 'COMPLETION TIME (ROUND TRIP) FOR EACH AC ON:'
      PRINT *, '      INTERTHEATER MISSIONS    ---   < 10.0'
      PRINT *, '      INTRATHEATER MISSIONS        ---   < 10.0'
      PRINT *, 'COMBAT VALUES FOR CLOSING UNITS FOR EACH PERIOD'
      PRINT *, '      ---   < 100.0'
      PRINT *, 'THE AMOUNT OF MHE PREPOSITIONED AT:'
      PRINT *, '      APOD IN PALLET EQUIVALENTS (PE)    ---   < 100,000'
      PRINT *, '      FRONT IN PE                          ---   < 100,000'
      PRINT *, 'FALCE CAPABILITY/DAY IN PE              ---   < 100,000'
      PRINT *, 'TRIGGER CAPABILITY TO PREPARE AIRDROP    ---   < 100,000'
      PRINT *

```

```

      PRINT *, 'IF YOU DO NOT HAVE ALL THE ABOVE DATA, YOU MAY TERMINATE
OR CONTINUE.'
      PRINT *, 'DO YOU WANT TO CONTINUE? (1=Y,0=N)'
11    READ *,NRES
      IF(NRES.GT.1.OR.NRES.LT.0) THEN
        PRINT *, 'NOT A VALID REPLY, PLEASE ENTER (1=Y/0=N).'
        GO TO 10
      END IF
      IF(NRES.EQ.0) STOP
C
C** ASK IF DESIRED TO SAVE DATA IN A FILE
C
      PRINT 5
      PRINT *, 'DO YOU WANT THE DATA SAVED IN AN EXTERNAL FILE?(1=Y,0=N)'
12    READ *,NSAVE
      IF(NSAVE.GT.1.OR.NSAVE.LT.0) THEN
        PRINT *, 'NOT A VALID REPLY, PLEASE ENTER (1=Y/0=N).'
        GO TO 12
      END IF
      IF(NSAVE.EQ.1) THEN
        PRINT *, 'ENTER THE NAME OF THE FILE YOU WANT THE DATA SAVED (8
CCHARACTERS MAX)'
        READ 2,FNAME
        OPEN(11,FILE=FNAME)
        REWIND(11)
      END IF
C
C** START THE INPUT PROCESS
C
      I=0
      PRINT 5
      PRINT *, 'I AM READY TO ACCEPT YOUR INPUTS, AND WILL PROMPT YOU FOR
C THE DATA.'
      PRINT *
      PRINT *, '-----'
      PRINT *, '--          ENTERING THE APOD DATA          --'
      PRINT *, '-----'
      PRINT *
      PRINT *, 'ENTER THE DISTANCE (KM,INTEGER) FROM US TO THE APOD.'
      READ *,DIST(1)
      PRINT *
      PRINT *, 'ENTER THE DISTANCE (INTEGER) FROM APOD TO FRONT.'
      READ *,DIST(2)
      PRINT *
      PRINT *, 'ENTER THE TIME (DAYS) BETWEEN MISSIONS (REAL) FOR:'
      DO 15 I=1,NMAC
        PRINT *, 'INTERTHEATER FOR THE ',NAME(I)
        READ *,TINTER(I)
        PRINT *, 'INTRATHEATER FOR THE ',NAME(I)
        READ *,TINTRAT(I)
15    CONTINUE
C
C** INPUT THE EASE WITH WHICH DIFFERENT TYPES OF
C** CARGO AND AIRCRAFT CAN BE UNLOADED
C

```

```

PRINT *
PRINT *, 'WANT TO CONSIDER THAT DIFFERENT CARGO FOR DIFFERENT
PRINT *, ' AIRCRAFT MAY BE EASIER TO OFFLOAD? (1=Y/0=N). '
15 READ *, NEAS
IF (NEAS.LT.0.OR.NEAS.GT.1) THEN
    PRINT *, 'NOT A VALID ANSWER! PLEASE ENTER (1=Y/0=N). '
    GO TO 16
END IF
IF (NEAS.EQ.0) GO TO 11
C
PRINT *
PRINT *, '***** NOTE: '
PRINT *, ' 1. THE NUMBER MUST BE BETWEEN 0 AND 1. '
PRINT *, ' 2. A NUMBER MUST BE INPUT REGARDLESS IF THE AC IS CA
1 PABLE OF '
PRINT *, ' CARRYING THE CARGO SIZE OR NOT. '
PRINT *
C
DO 13 I=1,NMAC
    PRINT *, 'FOR THE ', NAME(I), ' WHAT IS THE RELATIVE EASE FOR '
    PRINT *, ' OUT, OVER AND BULK CARGO? '
14 READ *, (EAS(I,J), J=1,3)
    DO 18 J=1,3
        IF (EAS(I,J).LT.0.OR.EAS(I,J).GT.1) THEN
            PRINT *, 'NOT A VALID RESPONSE! PLEASE ENTER (0...1). '
            GO TO 14
        END IF
18 CONTINUE
13 CONTINUE
15 CONTINUE
11 PRINT *
PRINT *, 'ENTER THE COMBAT VALUE INDEXES (REAL) FOR: '
DO 20 I=1,NP
    PRINT *, ' PERIOD # ', I
    READ *, CPI(I)
20 CONTINUE
PRINT *
PRINT *, 'ENTER THE MHE (FE) PREPOSITIONED (INTEGER) AT THE: '
PRINT *, ' AP00:
READ *, NHA(1)
PRINT *, ' FRONT: '
READ *, NHA(2)
PRINT *
PRINT *, 'ENTER THE NUMBER OF PE/CH/ (INTEGER) OF THE ALCE: '
READ *, NPAL
PRINT *
PRINT *, 'ENTER THE NUMBER OF LOADS/CH/ (REAL) OF THE RIGGERS: '
READ *, PC
PRINT *
C
C** ASK IF THE FILE IS TO BE VIEWED
C
PRINT *, 'DO YOU WANT TO SEE THE INPUT FILE? (1=Y/0=N) '
21 READ *, IVU
IF (IVU.GT.1.OR.IVU.LT.0) THEN

```

```

        PRINT *, 'NOT A VALID REPLY, PLEASE ENTER (1=Y/0=NO).
        GO TO 21
    END IF
    IF(NSAVE.EQ.0.AND.IVU.EQ.0) RETURN
C
C++ SET UP THE HEADINGS FOR THE VIEW AND/OR THE FILE
C
    HDG(1)='1. DISTANCES'
    HDG(2)='2. AIRCRAFT INTER INTRA OUT OVER BULK'
    HDG(3)='3. COMBAT VALUES OF UNITS CLOSING'
    HDG(4)='4. MISCELLANEOUS DATA'
C
C++ SET UP THE SUBHEADINGS UNLESS NSAVE=0 AND IVU=0
C
    SUBH(1)=' FROM US:'
    SUBH(2)=' FROM CORPS:'
    SUBH(3)=' PERIOD 1'
    SUBH(4)=' PERIOD 2'
    SUBH(5)=' PERIOD 3'
    SUBH(6)=' PERIOD 4'
    SUBH(17)=' MHE AT APOD'
    SUBH(18)=' MHE AT FRNT'
    SUBH(19)=' ALCE CAP'
    SUBH(20)=' RIGGER CAP'
C
C++ WRITE THE FILE TO OUTPUT
C
    WRITE (13,4)HDG(1)
    WRITE(13,6)SUBH(1),DIST(1)
    WRITE(13,6)SUBH(2),DIST(2)
    WRITE(13,4)HDG(2)
    DO 44 I=1,NMAC
        WRITE(13,9)NAME(I),TINTER(I),TINTRA(I),
C          EAS(I,1),EAS(I,2),EAS(I,3)
44    CONTINUE
    WRITE(13,4)HDG(3)
    DO 46 I=1,NF
        WRITE(13,17)SUBH(I-3),I,CP(I,1)
46    CONTINUE
    WRITE(13,4)HDG(4)
    DO 48 I=1,2
        WRITE(13,19)SUBH(I+16),NHAC(I)
48    CONTINUE
    WRITE(13,19)SUBH(19),NPAI
    WRITE(13,7)SUBH(20),RC
C
C++ WRITE THE FILE IF NSAVE=1
C
    IF(NSAVE.EQ.1) THEN
        WRITE (11,4)HDG(1)
        WRITE(11,6)SUBH(1),DIST(1)
        WRITE(11,6)SUBH(2),DIST(2)
        WRITE(11,4)HDG(2)
        DO 24 I=1,NMAC
            WRITE(11,9)NAME(I),TINTER(I),TINTRA(I),

```

```

C          EAS(I,1),EAS(I,2),EAS(I,3)
24      CONTINUE
        WRITE(11,4)HDG(3)
        DO 25 I=1,NP
            WRITE(11,17)SUBH(I+2),I,CPI(I)
25      CONTINUE
        WRITE(11,4)HDG(4)
        DO 26 I=1,2
            WRITE(11,19)SUBH(I+16),NHA(I)
26      CONTINUE
        WRITE(11,19)SUBH(19),NPAL
        WRITE(11,7)SUBH(20),RC
        END IF
C
C++ PRINT OUT THE FILE IF IVU = 1
C
        IF(IVU.EQ.1) THEN
            PRINT 5
            PRINT 4,HDG(1)
            PRINT 6,SUBH(1),DIST(1)
            PRINT 6,SUBH(2),DIST(2)
            PRINT 4,HDG(2)
            DO 34 I=1,NMAC
                PRINT 8,NAME(I),TINTER(I),TINTRA(I),
C          EAS(I,1),EAS(I,2),EAS(I,3)
34      CONTINUE
            PRINT 4,HDG(3)
            DO 35 I=1,NP
                PRINT 17,SUBH(I+2),I,CPI(I)
35      CONTINUE
            PRINT 4,HDG(4)
            DO 38 I=1,2
                PRINT 19,SUBH(I+16),NHA(I)
38      CONTINUE
            PRINT 19,SUBH(19),NPAL
            PRINT 7,SUBH(20),RC
            PRINT *
            PRINT *, 'INPUT ANY CHARACTER TO CONTINUE.'
            READ 2,FNAME
        END IF
C
C++ FORMAT STATEMENTS
C
2      FORMAT(A5)
4      FORMAT(A77)
5      FORMAT(////)
6      FORMAT(A14,F5.0)
7      FORMAT(A14,F5.1)
8      FORMAT(3X,A5,5X,5F6.1)
17     FORMAT(A10,I2,2X,F5.1)
19     FORMAT(A14,I6)
C
C
        RETURN
        END

```

```

C*****
C++ 3:
SUBROUTINE ARMINF
C
C++ READS IN AN EXTERNAL FILE THAT CONTAINS
C++ INFORMATION ABOUT THE ARMY UNITS
C
CHARACTER NAME(10)*6,HDG*77,FNAME*6,UNAME(11)*14,GNAME(3)*14
CHARACTER MS(4)*5,KG(5)*4,MD(3)*5
COMMON/ARM/NUNITS,NUNT(10),NTON(11,5),NFP(10),NAT(10),NFT(10),
C NTAC(10),NPK(10),NTP(11),NTV(11),IAB(10),NCI(10),IYP,NHQ(2)
COMMON/CHR/NAME,UNAME,GNAME,MS,KG,MD
C
C++ FIND THE FILE NAME WITH THE DATA
C
PRINT 8
PRINT *, 'ENTER THE FILE NAME (6 CHAR MAX) WITH THE ARMY UNITS.'
READ 2,FNAME
OPEN(12,FILE=FNAME)
REWIND(12)
C
C++ ASK IF THE FILE IS TO BE VIEWED
C
PRINT 8
PRINT *, 'DO YOU WANT TO SEE THE INPUT FILE? (1=Y/0=N)'
21 READ *,IVU
IF(IVU.GT.1.OR.IVU.LT.0) THEN
PRINT *, 'NOT A VALID REPLY, PLEASE ENTER (1=Y/0=N)'
GO TO 21
END IF
C
C++ READ IN THE DATA FILE AND PRINT IF IVU=1
C++ THE FIRST IYP UNITS ARE DEPLOYED TO THE FRONT
C++ WHILE THE LAST IYP+1 UNITS ARE AT THE APOD
C
22 READ(12,*)NUNITS,IYP,NHQ(1),NHQ(2)
WRITE(13,*)NUNITS,IYP,NHQ(1),NHQ(2)
READ(12,4)HDG
WRITE(13,4)HDG
IF(IVU.EQ.1) THEN
PRINT 8
PRINT *,NUNITS,IYP,NHQ(1),NHQ(2)
PRINT 4,HDG
END IF
DO 53 I=1,NUNITS
READ(12,8)UNAME(I),NUNT(I),NTON(I,1),NTON(I,2),NTON(I,3),
C NTON(I,4),NTON(I,5),NFP(I),NAT(I),NFT(I),NTAC(I),NPK(I),
C NTP(I),NTV(I),IAB(I),NCI(I)
WRITE(13,8)UNAME(I),NUNT(I),NTON(I,1),NTON(I,2),NTON(I,3),
C NTON(I,4),NTON(I,5),NFP(I),NAT(I),NFT(I),NTAC(I),NPK(I),
C NTP(I),NTV(I),IAB(I),NCI(I)
IF(IVU.EQ.0) GO TO 53
PRINT 8,UNAME(I),NUNT(I),NTON(I,1),NTON(I,2),NTON(I,3),
C NTON(I,4),NTON(I,5),NFP(I),NAT(I),NFT(I),NTAC(I),NPK(I),
C NTP(I),NTV(I),IAB(I),NCI(I)

```

```

53  CONTINUE
    CLOSE (12)
    J=NUNITS+1
    UNAME(J)='ALCE-FRONT'
    DO 60 I=1,5
        NTON(J,I)=NTON(NUNITS,I)
60  CONTINUE
    NTP(J)=NTP(NUNITS)
    NTV(J)=NTV(NUNITS)
    IF (IVU.EQ.1) THEN
        PRINT *
        PRINT *, 'INPUT ANY CHARACTER TO CONTINUE.'
        READ 2,FNAME
    END IF

C
C++  FORMAT STATEMENTS
C
2   FORMAT(A6)
4   FORMAT(A77)
6   FORMAT(A14,I2,I4,I6,I5,I13,I4,I3,I3)
8   FORMAT(//////////)
    CLOSE (12)

C
C
    RETURN
    END

C*****
C++  S7
    SUBROUTINE ARMIN3
C
C++  CREATES THE ARMY UNIT FILE FOR THE ANALYSIS
C
    CHARACTER HDG*77,FNAME*6,FIRST*14
    CHARACTER NAME(10)*6,UNAME(11)*14,GNAME(3)*14
    CHARACTER MS(4)*5,KG(5)*4,MD(3)*5
    COMMON/ARM/NUNITS,NUNT(10),NTON(11,5),NFP(10),NAT(10),NFT(10),
C    NTAC(10),NFK(10),NTP(11),NTV(11),IAB(10),NCI(10),IYP,NHQ(2)
    COMMON/CHR/NAME,UNAME,GNAME,MS,KG,MD

C
C
    PRINT 8
    PRINT *, '    THE FOLLOWING DATA IS NEEDED FOR EACH ARMY UNIT.'
    PRINT *, '-----'
    PRINT *
    PRINT *, 'UNIT NAME/10'
    PRINT *, 'NUMBER OF UNITS AVAILABLE'
    PRINT *, 'AMOUNT OF OUTSIDE TONNAGE'
    PRINT *, 'OVERSIZE TONNAGE'
    PRINT *, 'BULK SIZE TONNAGE'
    PRINT *, 'NUMBER OF PERSONNEL'
    PRINT *, 'SUPPLY CONSUMPTION RATE/DAY'
    PRINT *, 'FIREPOWER'
    PRINT *, 'ANTI-TANK POWER'
    PRINT *, 'FRONT LINE TRACE'
    PRINT *, 'NUMBER OF FIGHTER AC'
    PRINT *, '--- 14 CHAR MAX'
    PRINT *, '--- < 100'
    PRINT *, '--- < 100000'
    PRINT *, '--- < 100000'
    PRINT *, '--- < 100000'
    PRINT *, '--- < 100000'
    PRINT *, '--- < 100'
    PRINT *, '--- < 100'
    PRINT *, '--- < 100'
    PRINT *, '--- < 100'

```



```

PRINT +, 'INPUT THE UNIT'S NAME (14 CHAR MAX). IF FINISHED INPUT 0.'
READ 3, FIRST
IF (FIRST.EQ. '0') GO TO 20
I=I+1
UNAME(I)=FIRST
PRINT *
PRINT +, 'INPUT THE NUMBER OF UNITS (INTEGER) ABLE TO DEPLOY.'
READ +, NUNT(I)
PRINT *
PRINT +, 'INPUT THE TONNAGE (INTEGER) FOR THE FOLLOWING SIZE:'
PRINT +, '      OUT:'
READ +, NTON(I,1)
PRINT +, '      OVER:'
READ +, NTON(I,2)
PRINT +, '      BULK:'
READ +, NTON(I,3)
PRINT *
PRINT +, 'INPUT THE NUMBER OF PERSONNEL (INTEGER) IN THE ', UNAME(I)
READ +, NTON(I,4)
PRINT *
PRINT +, 'INPUT THE SUPPLY CONSUMPTION RATE/DAY (INTEGER):'
READ +, NTON(I,5)
PRINT *
PRINT +, 'INPUT THE POWER ASSOCIATED WITH THE UNIT (INTEGER) FOR:'
PRINT +, '      FIREPOWER:'
READ +, NFP(I)
PRINT +, '      ANTI-TANK:'
READ +, NAT(I)
PRINT +, '      FRONT LINE TRACE:'
READ +, NFT(I)
PRINT *
PRINT +, 'INPUT THE # OF FIGHTER AC (INTEGER) IN THE ', UNAME(I)
READ +, NTA(I)
IF (NTA(I).NE.0) THEN
    PRINT +, 'INPUT THE # OF FIGHTERS THAT CAN PARK AT THE APOD:'
    READ +, NPK(I)
    END IF
PRINT *
PRINT +, 'INPUT THE TON-MILE/DAY (INTEGER) THE UNIT CAN TRANSPORT
1 SUPPLIES:'
READ +, NTP(I)
PRINT *
PRINT +, 'INPUT THE MILES/DAY (INTEGER) THE UNIT CAN MOVE ITSELF:'
READ +, NTM(I)
PRINT *
PRINT +, 'IS THE UNIT AIRBORNE CAPABLE? (1=Y, 0=N).'
58 READ +, IAB(I)
IF (IAB(I).GT.1.OR.IAB(I).LT.0) THEN
    PRINT +, 'NOT A VALID REPLY, PLEASE ENTER (1=Y/0=N).'
    GO TO 58
END IF
PRINT *
PRINT +, 'INDICATE WHICH TYPE OF UNIT IT IS: ANSWER AS FOLLOWS:'
PRINT +, '-----'
PRINT +, '      1=COMBAT UNIT'

```

```

PRINT *, '          -1=COMBAT SUPPORT UNIT (IVE. HQ. OF ARTILLERY)
PRINT *, '          0=OTHER (IVE. ALCE, TRUCK COMP, OR FIGHTER ETC.)
68 READ *,INQ1(1)
IF INQ1(1).GT.1.OR.INQ1(1).LT.-1) THEN
PRINT *, 'NOT A VALID REPLY, PLEASE ENTER (1=Y/0=N). '
GO TO 68
END IF
C
C++ CHECK IF THE MEMORY FOR DATA CLOSE TO BEING FULL
C
IF (I.EQ.9) PRINT *, 'YOU ONLY HAVE MEMORY FOR ONE MORE UNIT.'
IF (I.EQ.10) THEN
PRINT *
PRINT *, '*****
PRINT *, '++      YOU ARE OUT OF MEMORY FOR ARMY DATA      ++
PRINT *, '*****
GO TO 20
END IF
C
C++ GO BACK TO GET MORE DATA
C
PRINT *
GO TO 11
C
C++ END OF THE INPUT
C
20 PRINT *
NUNITS=1
PRINT *, 'HOW MANY HQ UNITS ARE THERE IN THE DATA FILE? (0,1,2). '
19 READ *,INQ
IF INQ.LT.0.OR.INQ.GT.2) THEN
PRINT *, 'NOT A VALID ANSWER! ENTER ONE (0,1,2). '
GO TO 19
END IF
IF INQ.EQ.0) THEN
GO TO 18 I=1 AND
PRINT *, 'ENTER THE POSITION OF THE HQ UNIT.'
17 READ *,INHQ(1)
IF INHQ(1).LT.2.OR.INHQ(1).GT.NUNITS) THEN
PRINT *, 'NOT A VALID POSITION! RE-ENTER.'
GO TO 17
END IF
16 CONTINUE
END IF
I=NUNITS+1
UNAME(I)=ALCE-FRONT
DO 15 I=1,5
NTON(I)=NUNITS+1
15 CONTINUE
NTR(I)=NUNITS
NTX(I)=NUNITS
PRINT *, 'HOW MANY UNITS ARE DEPLOYED TO THE FRONT? '
20 READ *,IXP
IF (IXP.GT.NUNITS) THEN
PRINT *, 'NOT A VALID RESPONSE (IXP > NUNITS) . ENTER AGAIN.'

```

```

        GO TO 33
    END IF
    IF (NSHWE.EQ.1) WRITE(15,*) (NUNITS,IYP
    PRINT 8
    PRINT *, 'DO YOU WANT TO SEE THE INPUT FILE CREATED? (1=Y/0=N)'
24  READ *,IYU
    IF (IYU.GT.1.OR.IYU.LT.0) THEN
        PRINT *, 'NOT A VALID REPLY, PLEASE ENTER (1=Y/0=N)'
        GO TO 24
    END IF
0
0** WRITE OUT DATA TO THE FILE AND PRINT IF DESIRED
0
    HDG=TYPE          NUMB OUT OVER BULK PAW SPLY CP AT FT AC/PK
1  IF (IYAB.EQ.1)
        WRITE(13,*) (NUNITS,IYP,NHQ(1),NHQ(2))
        WRITE(13,4) HDG
    IF (IYU.EQ.1) THEN
        PRINT 8
        PRINT *, (NUNITS,IYP,NHQ(1),NHQ(2))
        PRINT 4,HDG
    END IF
    IF (NSHWE.EQ.1) THEN
        WRITE (12,*) (NUNITS,IYP,NHQ(1),NHQ(2))
        WRITE (12,4) HDG
    END IF
    DO 53 I=1,NUNITS
        WRITE(13,6) (NAME(I),NUNTI(I),NTON(I,1),NTON(I,2),NTON(I,3),
1          NTON(I,4),NTON(I,5),NPK(I),NAT(I),NET(I),NTAC(I),NPK(I),
2          NTR(I),NTW(I),IAB(I),NCI(I))
        IF (NSHWE.EQ.0) GO TO 54
        WRITE(12,6) (NAME(I),NUNTI(I),NTON(I,1),NTON(I,2),NTON(I,3),
3          NTON(I,4),NTON(I,5),NPK(I),NAT(I),NET(I),NTAC(I),NPK(I),
4          NTR(I),NTW(I),IAB(I),NCI(I))
54      IF (IYU.EQ.0) GO TO 53
        PRINT 8, (NAME(I),NUNTI(I),NTON(I,1),NTON(I,2),NTON(I,3),
6          NTON(I,4),NTON(I,5),NPK(I),NAT(I),NET(I),NTAC(I),NPK(I),
7          NTR(I),NTW(I),IAB(I),NCI(I))
53  CONTINUE
    IF (NSHWE.EQ.1) CLOSE (13)
    IF (IYU.EQ.1) THEN
        PRINT *
        PRINT *, 'INPUT ANY CHARACTER TO CONTINUE.'
        READ 3,PAYME
    END IF
0
0** FORMAT STATEMENTS
0
1  FORMAT (4F10.4)
2  FORMAT (17F10.4)
3  FORMAT (4I4,12,4I5,15,5I3,14,3I3)
4  FORMAT (17F10.4)
0
0
0  RETURN

```

```

      CHARACTER NAME(10)*5,UNAME(11)*14,GNAME(3)*14,FNAME*5,
      C   ADDR(4)*77,BUSR(20)*14,TITLE*77,BLANK*1
      CHARACTER MS(4)*5,RS(5)*4,MD(3)*5
      COMMON/ACF,NMAC,NMC(10),AVAL(1,10),UTE(1,3,4),KARG(10,5,2),KTR(10),
      C   GT(10),MHE(10),NFR(10),NFRN(10)
      COMMON/APD/DIST(20),TINTER(10),TINTR(10),EAS(10,4),CP(4),
      C   NRC(2),NRCAL,RC,SL(3,2,3)
      COMMON/ARM/UNITS,NUIT(10),NTCN(11,5),NFRY(10),NAT(10),NET(10),
      C   NTHCN(10),NRK(10),NTR(11),NTW(11),IAS(10),NDI(10),I/P,NRC(2)
      COMMON/THS/IF,LP,IP,NCR,NED,NFRIT,LL(1,3,2),LUX(1,2),LPR(1,3,4,5,4),
      C   LPS(1,3,4),AIO(150,300),RHS(150),LPS(3,5,4),LPSL(4),LPR(10),
      C   NTK(150),LMT(3),NCK(10),NMGTK(150),NOMX(150),NIO
      COMMON/CHR,NAME,UNAME,GNAME,MS,RS,MD
      C
      C++ GO TO THE CORRECT TYPE OF FILE
      C++ 10= AIRCRAFT DATA 20= APDD DATA 30= ARMY DATA
      C
      PRINT 7
      GO TO (10,20,30) IFILE
      C
      C++ AIRCRAFT DATA
      C
      10 PRINT *, 'ENTER THE FILE NAME DESIRED FOR THE AIRCRAFT DATA (5 CHAR
      C   1 MAX) '
      READ 2,FNAME
      OPEN 15,FILE=FNAME
      REWIND 15
      TITLE=TYPE UNMR PUL DROT APDD INTRA AERN OUT OVER BULK PAK
      15PD GTM MHE FR PRA
      WRITE 15,* NMAC
      WRITE 15,4 TITLE
      DO 14 I=1,NMAC
      WRITE 15,5 NAME(1),UNAME(1),AVAL(1,1),UTE(1,1),UTE(1,2),
      C   UTE(1,3),UTE(1,4),KARG(1,1,1),KARG(1,1,2),KARG(1,1,3),
      C   KARG(1,2,2),KARG(1,3,1),KARG(1,3,2),KARG(1,4,2),KTR(1),
      C   GT(1),MHE(1),NFR(1),NFRN(1)
      14 CONTINUE
      CLOSE 15
      RETURN
      C
      C++ APDD DATA AND HEADINGS
      C
      20 PRINT *, 'ENTER THE FILE NAME DESIRED FOR THE APDD DATA (5 CHAR
      C   1 MAX) '
      READ 2,FNAME
      OPEN 15,FILE=FNAME
      REWIND 15
      C
      C++ SET UP THE HEADINGS FOR APDD FILE
      C
      R03(1)= 1. DISTANCES
      R03(2)= 2. AIRCRAFT INTER INTRA OUT OVER BULK
      R03(3)= 3. COMBAT VALUES OF UNITS CLOSING
      R03(4)= 4. MISCELLANEOUS DATA
      BUSR(1)= ' FROM US:'

```

```

      J=NP+1
      DO 15 I=1,2
        PRINT 19,SUBH(J)+J,NPA+1
15    CONTINUE
C
C++ THE NUMBER OF PALLETS AN ALCE UNIT CAN DOWNLOAD
C
      J=NP+5
      PRINT 19,SUBH(J),NPAL
C
C++ RISER CAPABILITY PER DAY DETERMINED
C
      PRINT 17,SUBH(J)+1,RC
      PRINT +
      PRINT +
      PRINT +, 'INPUT ANY CHARACTER TO CONTINUE.'
      READ 2,BLANK
      PRINT 0
      RETURN
C
C++ ARMY PRINT OUT
C
30  TITLE= TYPE           NUMB  OUT  OVER  BULK  PAX   SPL OF AT FT AC
190  TR  TR  AB  CI
      PRINT +,NLNITS,1,IP,NRQ(1),NRQ(2)
      PRINT 4,TITLE
      DO 53 I=1,NLNITS
        PRINT 9,NAME(I),NUNT(I),NTON(I,1),NTON(I,2),NTON(I,3),
C          NTON(I,4),NTON(I,5),NPA(I),NAT(I),NET(I),NTAC(I),NPK(I),
C          NTF(I),NTVC(I),IAB(I),NCI(I)
53  CONTINUE
      PRINT +
      PRINT +
      RETURN
C
C++ FORMAT STATEMENTS
C
2    FORMAT(A1)
4    FORMAT(A77)
5    FORMAT(A6,14,5F5.1,1X,3(13,1X,12),14,14,F4.1,2(13),14)
6    FORMAT(A14,F6.0)
7    FORMAT(// '////////////////////')
8    FORMAT(A14,5F5.1)
9    FORMAT(A14,12,4I5,13,5I3,14,3I3)
17   FORMAT(A14,F6.1)
19   FORMAT(A14,13)
C
C
C      END
C+++++
C++ S-MRIT
      SUBROUTINE WRIT( IFILE)
C
C++ WRITES TO THE FILE DESIRED BY THE USER
C

```

```

C++ AIRCRAFT DATA
C
10 TITLE= TYPE NUMBER AML DROT APDZ INTRA ABRN OUT OVER BULK PRA
1940 GTM MHE FR PRA
PRINT *,NMAC
PRINT 4,TITLE
DO 14 I=1,NMAC
PRINT 8,NAME(I),MAC(I),AAAIL(I),UTEX(I,1),UTEX(I,2),UTEX(I,3),
C UTEX(I,4),KARG(I,1,1),KARG(I,1,2),KARG(I,2,1),KARG(I,2,2),
C KARG(I,3,1),KARG(I,3,2),KARG(I,4,2),KTS(I),GT(I),MHE(I),
C NFR(I),NPR(I)
14 CONTINUE
RETURN
C
C++ APDZ DATA AND HEADINGS
C
20 HDG(1)=1. DISTANCES
HDG(2)=2. AIRCRAFT INTER INTRA OUT OVER BULK
HDG(3)=3. COMBAT VALUES OF UNITS CLOSING
HDG(4)=4. MISCELLANEOUS DATA
SUBH(1)= FROM US
SUBH(2)= FROM CORPS
DO 22 I=3,NP+2
SUBH(I)= PERIOD
22 CONTINUE
SUBH(NP+3)= MHE AT APDZ
SUBH(NP+4)= MHE AT FRNT
SUBH(NP+5)= ALCE CAP
SUBH(NP+6)= RIGGER CAP
C
C++ PRINT OUT THE DISTANCES
C
PRINT 4,HDG(1)
DO 23 I=1,2
PRINT 8,SUBH(I),DIST(I)
23 CONTINUE
C
C++ PRINT OUT THE INTER AND INTRA TURN-TIMES
C
PRINT 4,HDG(2)
DO 24 I=1,NMAC
PRINT 8,NAME(I),TINTER(I),TINTRA(I),
C EAS(I,1),EAS(I,2),EAS(I,3)
24 CONTINUE
C
C++ PRINT OUT THE COMBAT POWER INDEXES FOR UNITS CLOSING AT THE FRONT
C
PRINT 4,HDG(3)
DO 25 I=1,4
PRINT 17,SUBH(I+2),CFI(I)
25 CONTINUE
C
C++ NUMBER OF MHE PREPOSITIONED
C
PRINT 4,HDG(4)

```

```

        END IF
C
C++ IF DONE THEN ASK IF THE DATA IS TO BE SAVED IN A FILE
C
        IF(MSUB.EQ.0) THEN
            PRINT 9
            PRINT *, 'SAVE THE NEW INFORMATION IN A FILE? (1=Y/0=N)'
36      READ *, ISAV
            IF(ISAV.LT.0.OR.ISAV.GT.1) THEN
                PRINT *, 'NOT A VALID REPLY! ENTER (1=Y/0=N).'
                GO TO 36
            END IF
            IF(ISAV.EQ.1) CALL WRIT(6)
            GO TO 60
        END IF
        CALL ARMINC(MSUB)
        NRES=1
        GO TO 30
C
C++ FORMAT STATEMENTS
C
2      FORMAT(A6)
4      FORMAT(A77)
5      FORMAT(A14,F6.0)
6      FORMAT(A14,5F6.1)
17     FORMAT(A14,F6.1)
18     FORMAT(A14,I6)
9      FORMAT(//)
        END
C*****
C++
        SUBROUTINE PRNT(IFILE)
C
C++ PRINTS OUT THE FILE DESIRED BY THE USER
C
        CHARACTER NAME(10)*6,UNAME(11)*14,GNAME(3)*14,
C      HDG(4)*77,SUBH(20)*14,TITLE*77,BLANK*1
        CHARACTER MS(4)*5,KG(5)*4,MD(3)*5
        COMMON/ACR/NHAC,NAC(10),AVAIL(10),UTE(10,4),KARG(10,5,2),KTS(10),
C      STX(10),PHE(10),NFR(10),NPRK(10)
        COMMON/APD/DIST(2),TINTER(10),TINTRA(10),EAS(10,4),CPI(4),
C      NHA(3),NPAL,RC,GLB(9,3)
        COMMON/ARM/NUNITS,NUNT(10),NTON(11,5),NFR(10),NAT(10),NFT(10),
C      NTHC(10),NPK(10),NTP(11),NTVC(11),IAB(10),NCI(10),IYP,NHQ(2)
        COMMON/TYPE/IF,LP,IF,NCR,NEC,NPRIT,LL(10,2),LU(10,2),LPK(10,4,5,4),
C      LPU(10,3,4),AIJ(150,300),RHS(150),LPS(3,5,4),LPBL(4),LPR(10),
C      NT(150),LMT(3),ND(10),NWGT(150),NDV(150),NID
        COMMON/CHS/NAME,UNAME,GNAME,MS,KG,MD
C
C++ GO TO THE CORRECT TYPE OF FILE
C++ 10= AIRCRAFT DATA 20= APD DATA 30= ARMY DATA
C
        PRINT 7
        GO TO (10,20,30) IFILE
C

```

```

PRINT *
PRINT *, 'REPLY 0 IF NO MORE CHANGES DESIRED'
24 READ *,MSUB
IF(MSUB.LT.0.OR.MSUB.GT.7) THEN
    PRINT *, 'NOT A VALID REPLY! ENTER (0...7). '
    GO TO 24
END IF
:
C++ IF DONE THEN ASK IF THE DATA IS TO BE SAVED IN A FILE
C
IF(MSUB.EQ.0) THEN
    PRINT *
    PRINT *, 'SAVE THE NEW INFORMATION IN A FILE? (1=Y/0=N). '
25 READ *,ISAV
IF(ISAV.LT.0.OR.ISAV.GT.1) THEN
    PRINT *, 'NOT A VALID REPLY! ENTER (1=Y/0=N). '
    GO TO 25
END IF
IF(ISAV.EQ.1) CALL WRIT(2)
GO TO 60
END IF
CALL ADDING MSUB
NRES=1
GO TO 23
C
C++ IF INFORMATION IS INTERNAL ASK IF A VIEW OF THE DATA IS DESIRED
C
30 IF(NRES.EQ.0) CALL ARMINF
IF(NRES.EQ.1) THEN
    PRINT *
    PRINT *, 'DO YOU WANT TO VIEW THE ARMY UNIT DATA? (1=Y/0=N). '
33 READ *,IVU
IF(IVU.LT.0.OR.IVU.GT.1) THEN
    PRINT *, 'NOT A VALID ANSWER! ENTER (1=Y/0=N). '
    GO TO 33
END IF
IF(IVU.EQ.1) CALL PRNT(3)
END IF
C
C++ SUBMENU FOR ARMY UNIT DATA CHANGE
C
PRINT *
PRINT *, '
PRINT *, '----- ARMY CHANGE SUBMENU'
PRINT *, '-----'
PRINT *, '1. DELETE A UNIT.'
PRINT *, '2. CHANGE THE NUMBER OF UNITS AVAILABLE.'
PRINT *, '3. CHANGE THE FIREPOWER FOR A UNIT.'
PRINT *, '4. CHANGE THE ANTI-TANK POWER FOR A UNIT.'
PRINT *, '5. CHANGE THE FRONT LINE TRACE FOR A UNIT.'
PRINT *
PRINT *, 'REPLY 0 IF NO MORE CHANGES DESIRED'
32 READ *,MSUB
IF(MSUB.LT.0.OR.MSUB.GT.5) THEN
    PRINT *, 'NOT A VALID REPLY! ENTER (0...7). '
    GO TO 32

```

```

C** IF DONE THEN ASK IF THE DATA IS TO BE SAVED IN A FILE
C
    IF (MSUB.EQ.0) THEN
        PRINT 9
        PRINT *, 'SAVE THE NEW INFORMATION IN A FILE? (1=Y/0=N). '
16      READ *, ISAV
        IF (ISAV.LT.0.OR.ISAV.GT.1) THEN
            PRINT *, 'NOT A VALID REPLY! ENTER (1=Y/0=N). '
            GO TO 16
        END IF
        IF (ISAV.EQ.1) CALL WRIT(1)
        GO TO 60
    END IF
    CALL ACFINDCMSUB
    NRES=1
    GO TO 10
C
C** THE AIRCRAFT DATA MUST BE ENTERED BEFORE THIS POINT
C** IF INFORMATION IS INTERNAL ASK IF A VIEW OF THE DATA IS DESIRED
C
20  IF (NRES.EQ.0) THEN
    PRINT *, 'HAS THE AIRCRAFT DATA BEEN ENTERED (YET)? (1=Y/0=N). '
21  READ *, INAC
    IF (INAC.LT.0.OR.INAC.GT.1) THEN
        PRINT *, 'NOT A VALID REPLY! ENTER (1=Y/0=N). '
        GO TO 21
    END IF
    IF (INAC.EQ.0) THEN
        CALL ACINF
        CALL APDINF
    END IF
    END IF
22  IF (NRES.EQ.1) THEN
    PRINT 9
    PRINT *, 'DO YOU WANT TO VIEW THE APD DATA? (1=Y/0=N). '
23  READ *, IVU
    IF (IVU.LT.0.OR.IVU.GT.1) THEN
        PRINT *, 'NOT A VALID ANSWER! ENTER (1=Y/0=N). '
        GO TO 23
    END IF
    IF (IVU.EQ.1) CALL PRNT(2)
    END IF
C
C** SUBMENU FOR APD DATA CHANGE
C
    PRINT 9
    PRINT *, '
    PRINT *, '
    PRINT *, '-----
    PRINT *, '      1. CHANGE THE DISTANCE FROM THE US TO THE APD. '
    PRINT *, '      2. CHANGE THE DISTANCE FROM THE APD TO THE FRONT. '
    PRINT *, '      3. CHANGE THE COMBAT VALUES FOR THE PERIODS. '
    PRINT *, '      4. CHANGE THE MHE PREPOSITIONED AT THE APD. '
    PRINT *, '      5. CHANGE THE MHE PREPOSITIONED AT THE FRONT. '
    PRINT *, '      6. CHANGE THE CAPACITY OF THE ALCE. '
    PRINT *, '      7. CHANGE THE CAPACITY OF THE RIGGERS. '

```

```

        END IF
        IF(MENU.EQ.0) THEN
            CALL POSTO
            RETURN
        END IF
        PRINT 9
C
C++ PRINT OUT THE OPTIONS AND INSTRUCTIONS
C
52 PRINT *, '***** NOTE: '
   PRINT *, 'HAS THE DATA TO BE CHANGED BEEN READ OR INPUT INTO THE PR
PROGRAM'
   PRINT *, 'BY A FILE OR INTERACTIVE MEANS? (1=Y/0=N)'
54 READ *, NRES
   IF(NRES.LT.0.OR.NRES.GT.1) THEN
       PRINT *, 'NOT A VALID REPLY! ENTER (1=Y/0=N)'
       GO TO 54
   END IF
   PRINT 9
C
C++ GO TO THE PROPER SUBMENU
C++ 10- AIRCRAFT 20- APOD 30- ARMY
C
   GO TO (10,20,30) MENU
C
C++ IF INFORMATION IS INTERNAL ASK IF A VIEW OF THE DATA IS DESIRED
C
10 IF(NRES.EQ.0) CALL ACINF
   PRINT 9
   PRINT *, 'DO YOU WANT TO VIEW THE AIRCRAFT DATA? (1=Y/0=N)'
12 READ *, IVU
   IF(IVU.LT.0.OR.IVU.GT.1) THEN
       PRINT *, 'NOT A VALID ANSWER! ENTER (1=Y/0=N)'
       GO TO 12
   END IF
   IF(IVU.EQ.1) CALL PRINT(10)
C
C++ SUBMENU FOR AIRCRAFT DATA CHANGE
C
   PRINT *
   PRINT *, '          AIRCRAFT CHANGE SUBMENU'
   PRINT *, '-----'
   PRINT *, '      1. DELETE AN AIRCRAFT.'
   PRINT *, '      2. CHANGE THE NUMBER OF AIRCRAFT AVAILABLE.'
   PRINT *, '      3. CHANGE THE AVAILABILITY OF AN AIRCRAFT.'
   PRINT *, '      4. CHANGE THE GROUND TIME FOR AN AIRCRAFT.'
   PRINT *
   PRINT *, 'REPLY 0 IF NO MORE CHANGES DESIRED'
   PRINT *
14 READ *, MSUB
   IF(MSUB.LT.0.OR.MSUB.GT.4) THEN
       PRINT *, 'NOT A VALID REPLY! ENTER ( 1, 2, 3, OR 4)'
       GO TO 14
   END IF
C

```



```

C**      20= AIRCRAFT   30= APOD   40= ARMY
C
      GO TO (20,30,40) NRES
C
C** ENTER THE AIRCRAFT DATA
C
  20  IF(NFILE.EQ.1) THEN
        CALL ACINS(NF)
      ELSE
        CALL ACINF
      END IF
      GO TO 10
C
C** ENTER THE APOD DATA
C
  30  IF(NFILE.EQ.1) THEN
        CALL APDINS(NMAC,NP)
      ELSE
        CALL APDINF
      END IF
      GO TO 10
C
C** ENTER THE ARMY DATA
C
  40  IF(NFILE.EQ.1) THEN
        CALL ARMS
      ELSE
        CALL ARMINF
      END IF
      GO TO 10
C
C** CHECK TO ENSURE THAT ALL THE DATA HAS BEEN ENTERED
C
  50  IF(NMAC.EQ.0) THEN
        NRES=1
        PRINT 9
        PRINT *,'/*****'
        PRINT *,'/**   YOU HAVE NOT ENTERED THE AIRCRAFT DATA YET!! **/'
        PRINT *,'/*****'
        PRINT 8
        GO TO 13
      END IF
      IF(DIST(1).EQ.0) THEN
        NRES=2
        PRINT 9
        PRINT *,'/*****'
        PRINT *,'/**   YOU HAVE NOT ENTERED THE APOD DATA YET!! **/'
        PRINT *,'/*****'
        PRINT 8
        GO TO 13
      END IF
      IF(UNIT3.EQ.0) THEN
        NRES=3
        PRINT 9
        PRINT *,'/*****'

```

```

5  FORMAT(10,12,10,1,A14)
6  FORMAT(10,12,10,1,A14,10,13)
7  FORMAT(10,12,10,1,A14,10,13,10)
8  FORMAT(10,12,10,1,A14,10,13,10)
9  END

```

C*****

C** S

SUBROUTINE POSTC

C

C** THIS ASKS THE USER AFTER HE/SHE HAS MADE CHANGES IF ALL THE DATA

C** NECESSARY TO CONTINUE HAS BEEN ENTERED

C

CHARACTER NAME(10)*6,UNAME(11)*14,GNAME(3)*14,FNAME*6

CHARACTER MS(4)*5,KG(5)*4,MD(3)*5

COMMON/ACF/NMAC,NAC(10),AVAIL(10),UTE(10,4),KARG(10,5,2),KTS(10),

C GT(10),MHE(10),NFP(10),NPRK(10)

COMMON/APD/DIST(2),TINTER(10),TINTRA(10),EAS(10,4),CPI(4),

C NHA(2),NPAL,RC,GLS(9,3)

COMMON/ARM/NUNITS,NUNT(10),NTON(11,5),NFP(10),NAT(10),NFT(10),

C NTAC(10),NPK(10),NTP(11),NTV(11),IAB(10),NCI(10),IYP,NHQ(2)

COMMON/TAB/NP,LP,IP,NCR,NEQ,NPRIT,LL(10,2),LU(10,2),LPX(10,4,5,4),

C LPU(10,3,4),AIG(150,300),RHS(150),LPS(3,5,4),LPPL(4),LPR(10),

C NT(150),LMT(3),NC(10),NWGT(150),NDV(150),NID

COMMON/CHR/NAME,UNAME,GNAME,MS,KG,MD

C

C** FIND OUT WHAT HAS TO BE INPUT BEFORE CONTINUING

C

10 PRINT 9

PRINT *, 'INDICATE WHAT HAS ** NOT ** BEEN INPUT YET. SELECT ONE.'

PRINT *, '-----'

PRINT *, '1. AIRCRAFT DATA.'

PRINT *, '2. APD DATA.'

PRINT *, '3. ARMY UNIT DATA.'

PRINT *, '** INPUT 0 IF ALL DATA HAS BEEN INPUT.'

12 READ *,NRES

IF(NRES.LT.0.OR.NRES.GT.3) THEN

PRINT *, 'NOT A VALID REPLY! ENTER (0...3).'

GO TO 12

END IF

IF(NRES.EQ.0) GO TO 50

C

C** DATA STILL NEEDS TO BE INPUT- FIND OUT HOW TO ENTER THE DATA

C

PRINT 9

13 PRINT *, 'HOW WILL YOU ENTER THE DATA? SELECT ONE.'

PRINT *, '-----'

PRINT *, '1. INTERACTIVELY.'

PRINT *, '2. FROM A USER DEFINED FILE.'

14 READ *,NFILE

IF(NFILE.LT.1.OR.NFILE.GT.2) THEN

PRINT *, 'NOT A VALID OPTION, PLEASE ENTER (1,2).'

GO TO 14

END IF

C

C** GO TO THE CORRECT STATION TO INPUT THE DATA

```

        PRINT *, 'NOT A VALID RESPONSE! ENTER AGAIN.'
        GO TO 44
    END IF
    IF(NRES.EQ.0) RETURN
    PRINT *
    PRINT *, 'ENTER THE DESIRED FIREPOWER OF ', UNAME(NRES)
    READ *, NFP(NRES)
    RETURN
C
C** CHANGE THE ANTI-TANK POWER OF A UNIT
C
50  PRINT *, 'SELECT THE UNIT YOU DESIRE TO CHANGE THE ANTI-TANK POWER
    1 OF.'
    PRINT *, '-----'
    1----'
    DO 52 I=1,NUNITS
        PRINT 6,I,UNAME(I),NAT(I)
52  CONTINUE
    PRINT *
    PRINT *, '0 IF NONE.'
54  READ *, NRES
    IF(NRES.LT.0.OR.NRES.GT.NUNITS) THEN
        PRINT *, 'NOT A VALID RESPONSE! ENTER AGAIN.'
        GO TO 54
    END IF
    IF(NRES.EQ.0) RETURN
    PRINT *
    PRINT *, 'ENTER THE DESIRED ANTI-TANK POWER OF ', UNAME(NRES)
    READ *, NAT(NRES)
    RETURN
C
C** CHANGE THE FRONT LINE TRACE OF A UNIT
C
60  PRINT *, 'SELECT THE UNIT YOU DESIRE TO CHANGE THE FRONT LINE TRAC
    1E OF.'
    PRINT *, '-----'
    1----'
    DO 62 I=1,NUNITS
        PRINT 6,I,UNAME(I),NFT(I)
62  CONTINUE
    PRINT *
    PRINT *, '0 IF NONE.'
64  READ *, NRES
    IF(NRES.LT.0.OR.NRES.GT.NUNITS) THEN
        PRINT *, 'NOT A VALID RESPONSE! ENTER AGAIN.'
        GO TO 64
    END IF
    IF(NRES.EQ.0) RETURN
    PRINT *
    PRINT *, 'ENTER THE DESIRED FRONT LINE TRACE OF ', UNAME(NRES)
    READ *, NFT(NRES)
    RETURN
C
C** FORMAT STATEMENT
C

```

```

        NTON(I,J)=NTON(I-1,J)
26    CONTINUE
        NFP(I)=NFP(I+1)
        NAT(I)=NAT(I+1)
        NFT(I)=NFT(I+1)
        NTAC(I)=NTAC(I+1)
        NPK(I)=NPK(I+1)
        NTP(I)=NTP(I+1)
        NTV(I)=NTV(I+1)
        IAB(I)=IAB(I+1)
        NCI(I)=NCI(I+1)
25    CONTINUE
C
C** SET UP THE DATA FOR THE ALCE AT THE FRONT
C
        J=NUNITS+1
        UNAME(J)='ALCE-FRONT'
        DO 61 I=1,5
            NTON(J,I)=NTON(NUNITS,I)
61    CONTINUE
        NTP(J)=NTP(NUNITS)
        NTV(J)=NTV(NUNITS)
        RETURN
C
C** CHANGE THE NUMBER OF UNITS AVAILABLE FOR THE DEPLOYMENT
C
30    PRINT *, ' SELECT THE UNIT YOU DESIRE TO INCREASE THE NUMBER OF '
        PRINT *, '-----'
        DO 32 I=1,NUNITS
            PRINT 6,I,UNAME(I),NUNT(I)
32    CONTINUE
        PRINT *
        PRINT *, '0 IF NONE.'
34    READ *,NRES
        IF(NRES.LT.0.OR.NRES.GT.NUNITS) THEN
            PRINT *, 'NOT A VALID RESPONSE! ENTER AGAIN.'
            GO TO 34
        END IF
        IF(NRES.EQ.0) RETURN
        PRINT *
        PRINT *, 'ENTER THE DESIRED NUMBER OF ',UNAME(NRES)
        READ *,NUNT(NRES)
        RETURN
C
C** CHANGE THE AVAILABILITY OF AN AIRCRAFT
C
40    PRINT *, ' SELECT THE UNIT YOU DESIRE TO CHANGE THE FIREPOWER OF '
        PRINT *, '-----'
        DO 42 I=1,NUNITS
            PRINT 6,I,UNAME(I),NFP(I)
42    CONTINUE
        PRINT *
        PRINT *, '0 IF NONE.'
44    READ *,NRES
        IF(NRES.LT.0.OR.NRES.GT.NUNITS) THEN

```

```

C** CHANGE THE CAPACITY OF THE RIGGERS
C
70  PRINT *, 'ENTER THE NEW LOADS/DAY OF THE RIGGERS (REAL). '
    READ *, RC
C
C** FORMAT STATEMENTS
C
9   FORMAT(////////////////////)
C
C
    RETURN
    END
C*****
C** 5
    SUBROUTINE ARMINC(ICHG)
C
C** THIS WILL MAKE CHANGES AS DESIRED FROM THE CHANGE SUBROUTINE
C** FOR THE ARMY UNIT DATA
C
    CHARACTER NAME(10)*6, UNAME(11)*14, GNAME(3)*14, FNAME
    CHARACTER MS(4)*5, KG(5)*4, MD(3)*5
    COMMON/ARM/NUNITS, NUNT(10), NTON(11,5), NFP(10), NAT(10), NPT(10),
C   NTA(10), NPK(10), NTP(11), NTV(11), IAB(10), NCI(10), IYP, NHQ(2)
    COMMON/CHR/NAME, UNAME, GNAME, MS, KG, MD
C
C** GO TO THE CORRECT STATION FOR THE CHANGE
C
    PRINT 9
    GO TO (20,30,40,50) ICHG
C
C** DELETE AN ARMY UNIT FROM THE DATA LIST
C
20  PRINT *, '      SELECT ONE OF THE ARMY UNITS TO DELETE'
    PRINT *, '-----'
    DO 22 I=1, NUNITS-1
        PRINT 5, I, UNAME(I)
22  CONTINUE
    PRINT *
    PRINT *, '0 IF NONE.'
24  READ *, NRES
    IF(NRES.LT.0.OR.NRES.GT.NUNITS) THEN
        PRINT *, 'NOT A VALID RESPONSE! ENTER AGAIN.'
        GO TO 24
    END IF
    IF(NRES.EQ.0) RETURN
    IF(NRES.LT.NHQ(1)) NHQ(1)=NHQ(1)-1
    IF(NRES.EQ.NHQ(1)) NHQ(1)=0
    IF(NRES.LT.NHQ(2)) NHQ(2)=NHQ(2)-1
    IF(NRES.EQ.NHQ(2)) NHQ(2)=0
    IF(NRES.LE.IYP) IYP=IYP-1
    NUNITS=NUNITS-1
    DO 25 I=NRES, NUNITS
        UNAME(I)=UNAME(I+1)
        NUNT(I)=NUNT(I+1)
    DO 26 J=1,5

```

```

      END
C*****
C** 8
      SUBROUTINE APDINC(ICHG)
C
C** THIS WILL MAKE CHANGES AS DESIRED FROM THE CHANGE SUBROUTINE
C** FOR THE APOD DATA
C
      COMMON/APD/DIST(2),TINTER(10),TINTRA(10),EAS(10,4),CPI(4),
C      NHA(2),NPAL,RC,GLS(9,3)
C
C** GO TO THE CORRECT STATION FOR THE CHANGE
C
      PRINT 9
      GO TO (10,20,30,40,50,60,70) ICHG
C
C** CHANGE THE DISTANCE FROM THE US TO THE APOD
C
      10  PRINT *, 'ENTER THE NEW DISTANCE (REAL) FROM THE US TO THE APOD.'
      READ *,DIST(1)
      RETURN
C
C** CHANGE THE DISTANCE FROM THE APOD TO THE FRONT
C
      20  PRINT *, 'ENTER THE NEW DISTANCE (REAL) FROM THE APOD TO THE FRONT.'
      READ *,DIST(2)
      RETURN
C
C** CHANGE THE COMBAT VALUES OF CLOSING UNITS
C
      30  PRINT *, 'ENTER THE COMBAT VALUES FOR THE FOLLOWING PERIODS:'
      DO 32 I=1,NP
          PRINT *, '  FOR PERIOD#',I
          READ *,CPI(I)
      32  CONTINUE
      RETURN
C
C** CHANGE THE MHE PREPOSITIONED AT THE APOD
C
      40  PRINT *, 'ENTER THE NEW MHE PREPOSITIONED AT THE APOD (INTEGER).'
      READ *,NHA(1)
      RETURN
C
C** CHANGE THE MHE PREPOSITIONED AT THE FRONT
C
      50  PRINT *, 'ENTER THE NEW MHE PREPOSITIONED AT THE FRONT (INTEGER).'
      READ *,NHA(2)
      RETURN
C
C** CHANGE THE CAPACITY OF THE ALCE
C
      60  PRINT *, 'ENTER THE NEW CAPACITY OF THE ALCE (INTEGER).'
      READ *,NPAL
      RETURN
C

```

```

      SUBH(2)='    FROM CORPS:'
      DO 22 I=3,NP+2
        SUBH(I)='    PERIOD '
22    CONTINUE
      SUBH(NP+3)='    MHE AT APOD'
      SUBH(NP+4)='    MHE AT FRNT'
      SUBH(NP+5)='    ALCE CAP'
      SUBH(NP+6)='    RIGGER CAP'
C
C** WRITE THE DISTANCES
C
      WRITE(15,4)HDG(1)
      DO 23 I=1,2
        WRITE(15,6)SUBH(I),DIST(I)
23    CONTINUE
C
C** WRITE THE INTER AND INTRA TURN-TIMES
C
      WRITE(15,4)HDG(2)
      DO 24 I=1,NMAC
        WRITE(15,8)NAME(I),TINTER(I),TINTRA(I),
C          EAS(I,1),EAS(I,2),EAS(I,3)
24    CONTINUE
C
C** WRITE THE COMBAT POWER INDEXES FOR UNITS CLOSING AT THE FRONT
C
      WRITE(15,4)HDG(3)
      DO 26 I=1,4
        WRITE(15,17)SUBH(I+2),CPI(I)
26    CONTINUE
C
C** NUMBER OF MHE PREPOSITIONED
C
      WRITE(15,4)HDG(4)
      J=NP+2
      DO 28 I=1,2
        WRITE(15,19)SUBH(I+J),NHA(I)
28    CONTINUE
C
C** THE NUMBER OF PALLETS AN ALCE UNIT CAN DOWNLOAD
C
      J=NP+5
      WRITE(15,19)SUBH(J),NPAL
C
C** RIGGER CAPABILITY PER DAY DETERMINED
C
      WRITE(15,6)SUBH(J+1),RC
      CLOSE(15)
      RETURN
C
C** WRITE A FILE FOR ARMY DATA
C
30    PRINT *, 'ENTER THE FILE NAME DESIRED FOR THE ARMY DATA (6 CHAR
      1MAX): '
      READ 2, FNAME

```

```

OPEN(15,FILE=FILENAME)
REWIND(15)
TITLE= TYPE           NUME OUT OVER BULK PAK SPL CP AT FT AC
1PK TP TZ AB Q1'
WRITE(15,*)NUNITS,IYP,NHQ(1),NHQ(2)
WRITE(15,4)TITLE
DO 53 I=1,NUNITS
WRITE(15,9)UNAME(I),NUNT(I),NTON(I,1),NTON(I,2),NTON(I,3),
C      NTON(I,4),NTON(I,5),NFP(I),NAT(I),NFT(I),NTAC(I),NPK(I),
C      NTP(I),NTP(I),IAS(I),NCI(I)
53 CONTINUE
CLOSE(15)
RETURN

C
C** FORMAT STATEMENTS
C
3 FORMAT(A6)
4 FORMAT(A70)
5 FORMAT(A6,I4,5F5.1,1X,3(I3,1X,I2),I4,I4,F4.1,2(I3),I4)
6 FORMAT(A14,F6.0)
7 FORMAT(////////////////////)
8 FORMAT(A14,5F5.1)
9 FORMAT(A14,I2,4I6,I5,5I3,I4,3I3)
17 FORMAT(A14,F6.1)
19 FORMAT(A14,I6)
C
C
END

C*****
C** SP
SUBROUTINE REALC
C
C**
C** THIS BUILDS THE EQUALITY CONSTRAINTS FOR THE FOLLOWING ISSUES:
C** 1. SHIPMENT OF UNITS TO THE APOD
C** 2. SHIPMENT OF UNITS TO THE FRONT VIA DIRECT DELIVERY
C** 3. SHIPMENT OF UNITS TO THE FRONT VIA AIRBORNE DELIVERY
C** 4. SHIPMENT OF SUPPLIES FOR THE THEATER
C** 5. SHIPMENT OF SUPPLIES FOR THE FRONT
C** 6. SHIPMENT OF SUPPLIES FOR THE APOD
C** 7. RIGGER CAPABILITY
C
C
CHARACTER NAME(100)*6,UNAME(11)*14,GNAME(3)*14,MS(4)*5,KG(5)*4,
1 MD(3)*5
COMMON/ACF/NMAC,NAC(10),AVAIL(10),UTE(10,4),KARG(10,5,2),KTS(10),
C GT(10),MHE(10),NFP(10),NPRK(10)
COMMON/APD/DIST(2),TINTER(10),TINTRK(10),EAS(10,4),CPI(4),
C NHA(2),NPAL,RC,GLS(9,3)
COMMON/ARM/NUNITS,NUNT(10),NTON(11,5),NFP(10),NAT(10),NFT(10),
C NTAC(10),NPK(10),NTP(11),NTP(11),IAS(10),NCI(10),IYP,NHQ(2)
COMMON/TAB/NP,LP,IP,NCR,NEB,NFRIT,LL(10,2),LUX(10,2),LPX(10,4,5,4),
C LPU(10,3,4),AIJ(150,300),RHS(150),LPS(3,5,4),LPSL(4),LPR(10),
C NT(150),LMT(3),NC(10),NWGT(150),NDX(150),NID
COMMON/CHR/NAME,UNAME,GNAME,MS,KG,MD

```

```

C
C++ CHECK TO SEE IF ECHO DESIRED
C
  PRINT *,'/-----/'
  PRINT *,'/              EQUALITY  CONSTRAINTS/'
  PRINT *,'/-----/'
  WRITE(13,*)'/-----/'
  WRITE(13,*)'/              EQUALITY  CONSTRAINTS/'
  WRITE(13,*)'/-----/'
*   PRINT *,'/ECHO ON THE EQUALITY CONSTRAINTS?(1=Y,0=N).'/
*   READ *,LANS
  LANS=0
C-1
C++ CONSTRAINT FOR THE SHIPMENT OF DEPLOYED UNITS TO THE FRONT
C
  PRINT *,'/-----/'
  PRINT *,'/              UNIT SHIPMENT TO  AP00/'
  PRINT *,'/-----/'
  WRITE(13,*)'/-----/'
  WRITE(13,*)'/              UNIT SHIPMENT TO  AP00/'
  WRITE(13,*)'/-----/'
C++ FOR EACH PERIOD
  DO 90 L=1,NP
C
C++ FOR OUT, OVER, AND BULK CARGO
C
  DO 92 K=1,3
    NEQ=NEQ+1
C.....
C-
C++ FOR EACH TYPE OF AIRCRAFT
  DO 94 I=1,NMAC
C
C++ CHECK TO SEE IF BEYOND THE LIMITS OF EITHER THE TYPE
C++ OF MISSION L(1,1), TYPE OF CARGO L(1,2), OR FP
C
      IF(L.LT.LL(1,1) .OR. 2.GT.LU(1,1)) GO TO 94
      IF(K.LT.LL(1,2) .OR. K.GT.LU(1,2)) GO TO 94
      IF(L.LT.NFP(1)) GO TO 94
      AIOXNEQ,LPX(1,2,K,L)= KARG(1,K,1)
      WRITE(13,04)NEQ,LPX(1,2,K,L),AIOXNEQ,LPX(1,2,K,L),
1          NAME(1),MS(2),KG(K),L
      IF (LANS.EQ.0) GO TO 94
      PRINT 04,NEQ,LPX(1,2,K,L),AIOXNEQ,LPX(1,2,K,L),
1          NAME(1),MS(2),KG(K),L
94      CONTINUE
C
C++ FOR EACH TYPE OF UNIT SHIPPED TO THE AP00, J=2 FOR THOSE REMAINING
C++ AT THE AP00 OR WALKING TO THE FRONT, J=3 FOR THOSE UNITS DEPLOYED
C++ TO THE FRONT BUT FLYING INTRATHEATER AIRLIFT
C
  DO 96 I=1,NUNITS+1
    IF(IAB(1).EQ.1) GO TO 96
    DO 97 J=2,3
      IF(1.GT.IYP.AND.J.NE.2.AND.1.LE.NUNITS) GO TO 97

```

```

      AIJ(NEQ,LPU(1,J,L))= -NTON(I,K)
      WRITE(13,03)NEQ,LPU(1,J,L),AIJ(NEQ,LPU(1,J,L)),
1      UNAME(I),MDX(J),L
      IF (LANS.EQ.0) GO TO 97
      PRINT 03,NEQ,LPU(1,J,L),AIJ(NEQ,LPU(1,J,L)),
1      UNAME(I),MDX(J),L
97      CONTINUE
96      CONTINUE
C
C** THE EXCESS FOR THIS PERIOD
C
      AIJ(NEQ,LPS(2,K,L))= -1.0
      WRITE(13,02)NEQ,LPS(2,K,L),AIJ(NEQ,LPS(2,K,L))
      IF (LANS.EQ.0) GO TO 98
      PRINT 02,NEQ,LPS(2,K,L),AIJ(NEQ,LPS(2,K,L))
C
C** THE EXCESS FROM THE PREVIOUS PERIOD
C
98      IF(L.NE.1) AIJ(NEQ,LPS(2,K,L-1))= 1.0
      IF(L.EQ.1) GO TO 99
      WRITE(13,02)NEQ,LPS(2,K,L-1),AIJ(NEQ,LPS(2,K,L-1))
      IF (LANS.EQ.0) GO TO 99
      PRINT 02,NEQ,LPS(2,K,L-1),AIJ(NEQ,LPS(2,K,L-1))
C
C** INPUT THE RIGHT HAND SIDE OF THE EQUATION
C
99      RHS(NEQ)= 0.0
      WRITE(13,08)NEQ,RHS(NEQ)
      IF(LANS.EQ.0) GO TO 92
      PRINT 08,NEQ,RHS(NEQ)
C.....
92      CONTINUE
C** FOR PASSENGERS
      NEQ=NEQ+1
C.....
C** FOR EACH TYPE OF AIRCRAFT
      DO 100 I=1,NMAC
C
C** CHECK TO SEE IF BEYOND THE LIMITS OF EITHER THE TYPE
C** OF MISSION L(1,1) OR NFP
C
      IF(2.LT.LL(1,1) .OR. 2.GT.LU(1,1)) GO TO 100
      IF(L.LT.NFP(1)) GO TO 100
      DO 102 K=LL(1,2),LU(1,2)
      AIJ(NEQ,LPX(1,2,K,L))= KARG(1,K,2)
      WRITE(13,04)NEQ,LPX(1,2,K,L),AIJ(NEQ,LPX(1,2,K,L)),
1      NAME(I),MS(2),KG(K),L
      IF (LANS.EQ.0) GO TO 103
      PRINT 04,NEQ,LPX(1,2,K,L),AIJ(NEQ,LPX(1,2,K,L)),
1      NAME(I),MS(2),KG(K),L
C
C** CONSIDER SUPPLY CARGO IF BULK CARGO IS CONSIDERED
C
103      IF(K.EQ.3) THEN
      AIJ(NEQ,LPX(1,2,5,L))=KARG(1,5,1)

```

```

        WRITE(13,04)NEQ,LPU(1,2,5,L),AIJ(NEQ,LPU(1,2,5,L)) ,
1         NAME(I),MS(2),KG(5),L
        IF(LANS.EQ.0) GO TO 102
    ELSE
        GO TO 102
    END IF
        PRINT 04,NEQ,LPU(1,2,5,L),AIJ(NEQ,LPU(1,2,5,L)),
1         NAME(I),MS(2),KG(5),L
102        CONTINUE
100        CONTINUE
C
C++ FOR EACH TYPE OF UNIT
C
        DO 104 I=1,NUNITS+1
            IF(IAB(I).EQ.1) GO TO 104
            DO 105 J=2,3
                IF(I.GT.IYP.AND.J.NE.2.AND.I.LE.NUNITS) GO TO 105
                AIJ(NEQ,LPU(I,J,L))= -NTON(I,4)
                WRITE(13,03)NEQ,LPU(I,J,L),AIJ(NEQ,LPU(I,J,L)),
1                 UNAME(I),MD(J),L
                IF (LANS.EQ.0) GO TO 105
                PRINT 03,NEQ,LPU(I,J,L),AIJ(NEQ,LPU(I,J,L)),
1                 UNAME(I),MD(J),L
105                CONTINUE
104                CONTINUE
C
C++ THE EXCESS FOR THIS PERIOD
C
                AIJ(NEQ,LPS(2,4,L))= -1.0
                WRITE(13,02)NEQ,LPS(2,4,L),AIJ(NEQ,LPS(2,4,L))
                IF (LANS.EQ.0) GO TO 106
                PRINT 02,NEQ,LPS(2,4,L),AIJ(NEQ,LPS(2,4,L))
C
C++ THE EXCESS FROM THE PREVIOUS PERIOD
C
106                IF(L.NE.1) AIJ(NEQ,LPS(2,4,L-1))= 1.0
                IF(L.EQ.1) GO TO 107
                WRITE(13,02)NEQ,LPS(2,4,L-1),AIJ(NEQ,LPS(2,4,L-1))
                IF (LANS.EQ.0) GO TO 107
                PRINT 02,NEQ,LPS(2,4,L-1),AIJ(NEQ,LPS(2,4,L-1))
C
C++ INPUT THE RIGHT HAND SIDE OF THE EQUATION
C
107                RHS(NEQ)= 0.0
                WRITE(13,08)NEQ,RHS(NEQ)
                IF(LANS.EQ.0) GO TO 90
                PRINT 08,NEQ,RHS(NEQ)
C.....
90                CONTINUE
C-2
C++ SET UP EQUATIONS FOR THE SHIPMENT OF UNITS VIA
C++ DIRECT DELIVERY (J=1) ONLY IF THERE ARE DIRECT
C++ DELIVERY AC AVAILABLE (LMT(1).NE.0)
C
                IF(LMT(1).EQ.0) GO TO 07

```

```

PRINT *,/-----/
PRINT *,/          UNIT  SHIPMENT  DIRECT/
PRINT *,/-----/
WRITE(13,*)/-----/
WRITE(13,*)/          UNIT  SHIPMENT  DIRECT/
WRITE(13,*)/-----/
C** FOR EACH PERIOD
DO 10 L=1,NP
C
C** FOR OUT, OVER, AND BULK CARGO
C
DO 12 K=1,3
    NEG=NEG+1
C.....
C** FOR EACH TYPE OF AIRCRAFT
DO 14 I=1,NMAC
C
C** CHECK TO SEE IF BEYOND THE LIMITS OF EITHER THE TYPE
C** OF MISSION L(I,1), TYPE OF CARGO L(I,2), OR FP
C
    IF(L.LT.LL(I,1) .OR. 1.GT.LUX(I,1)) GO TO 14
    IF(K.LT.LL(I,2) .OR. K.GT.LUX(I,2)) GO TO 14
    IF(L.LT.NFP(I)) GO TO 14
    AIJ(NEG,LPX(I,1,K,L))= KARG(I,K,1)
    WRITE(13,04)NEG,LPX(I,1,K,L),AIJ(NEG,LPX(I,1,K,L)),
1      NAME(I),MS(I),KG(K),L
    IF (LANS.EQ.0) GO TO 14
    PRINT 04,NEG,LPX(I,1,K,L),AIJ(NEG,LPX(I,1,K,L)),
1      NAME(I),MS(I),KG(K),L
14    CONTINUE
C** FOR EACH UNIT THAT IS NOT AIRBORNE CAPABLE
C
DO 16 I=1,NUNITS-1
    IF(INAB(I).EQ.1) GO TO 16
    IF(1.GT.IYP.AND.1.LE.NUNITS) GO TO 16
    AIJ(NEG,LPU(I,1,L))= -NTON(I,K)
    WRITE(13,03)NEG,LPU(I,1,L),AIJ(NEG,LPU(I,1,L)),
1      UNAME(I),MD(I),L
    IF (LANS.EQ.0) GO TO 16
    PRINT 03,NEG,LPU(I,1,L),AIJ(NEG,LPU(I,1,L)),
1      UNAME(I),MD(I),L
16    CONTINUE
C
C** THE EXCESS FOR THIS PERIOD
C
    AIJ(NEG,LPS(I,K,L))= -1.0
    WRITE(13,02)NEG,LPS(I,K,L),AIJ(NEG,LPS(I,K,L))
    IF (LANS.EQ.0) GO TO 18
    PRINT 02,NEG,LPS(I,K,L),AIJ(NEG,LPS(I,K,L))
C
C** THE EXCESS FROM THE PREVIOUS PERIOD
C
18    IF(L.NE.1) AIJ(NEG,LPS(I,K,L-1))= 1.0
    IF(L.EQ.1) GO TO 19
    WRITE(13,02)NEG,LPS(I,K,L-1),AIJ(NEG,LPS(I,K,L-1))

```

```

      IF (LANS.EQ.0) GO TO 19
      PRINT 02,NEQ,LPX(I,K,L-1),AIJ(NEQ,LPX(I,K,L-1))
C.....
C** 1. LEFT THE RIGHT HAND SIDE OF THE EQUATION
C
      RHS(NEQ) = 0.0
      WRITE (13,03)NEQ,RHS(NEQ)
      IF (LANS.EQ.0) GO TO 12
      PRINT 08,NEQ,RHS(NEQ)
C.....
      12      CONTINUE
C**  FOR PASSENGERS
      NEQ=NEQ+1
C.....
C**  FOR EACH TYPE OF AIRCRAFT
      DO 20 I=1,NMAC
C
C**  CHECK TO SEE IF BEYOND THE LIMITS OF EITHER THE TYPE
C**  OF MISSION L(I,1) OR NFP
C
      IF(I.LT.LL(I,1) .OR. 1.GT.LU(I,1)) GO TO 20
      IF(L.LT.NFP(I)) GO TO 20
      DO 23 K=LL(I,2),LU(I,2)
      AIJ(NEQ,LPX(I,1,K,L))= KARG(I,K,2)
      WRITE(13,04)NEQ,LPX(I,1,K,L),AIJ(NEQ,LPX(I,1,K,L)),
1      NAME(I),MS(I),KG(K),L
      IF (LANS.EQ.0) GO TO 23
      PRINT 04,NEQ,LPX(I,1,K,L),AIJ(NEQ,LPX(I,1,K,L)),
1      NAME(I),MS(I),KG(K),L
C
C**  CONSIDER SUPPLY CARGO IF BULK CARGO IS CONSIDERED
C
      23      IF(K.EQ.3) THEN
      AIJ(NEQ,LPX(I,1,5,L))=KARG(I,5,1)
      WRITE(13,04)NEQ,LPX(I,1,5,L),AIJ(NEQ,LPX(I,1,5,L)),
1      NAME(I),MS(I),KG(5),L
      IF(LANS.EQ.0) GO TO 22
      ELSE
      GO TO 22
      END IF
      PRINT 04,NEQ,LPX(I,1,5,L),AIJ(NEQ,LPX(I,1,5,L)),
1      NAME(I),MS(I),KG(5),L
      22      CONTINUE
      20      CONTINUE
C**  FOR EACH TYPE OF UNIT
      DO 24 I=1,NUNITS+1
      IF(IAB(I).EQ.1.) GO TO 24
      IF(1.GT.IYP.AND.1.LE.NUNITS) GO TO 24
      AIJ(NEQ,LPU(I,1,L))= -NTON(I,4)
      WRITE(13,03)NEQ,LPU(I,1,L),AIJ(NEQ,LPU(I,1,L)),
1      UNAME(I),MD(I),L
      IF (LANS.EQ.0) GO TO 24
      PRINT 03,NEQ,LPU(I,1,L),AIJ(NEQ,LPU(I,1,L)),
1      UNAME(I),MD(I),L
      24      CONTINUE

```

```

C
C** THE EXCESS FOR THIS PERIOD
C
      AIJ(NEQ,LPS(1,4,L))= -1.0
      WRITE(13,02)NEQ,LPS(1,4,L),AIJ(NEQ,LPS(1,4,L))
      IF (LANS.EQ.0) GO TO 26
      PRINT 02,NEQ,LPS(1,4,L),AIJ(NEQ,LPS(1,4,L))

C
C** THE EXCESS FROM THE PREVIOUS PERIOD
C
26      IF(L.NE.1) AIJ(NEQ,LPS(1,4,L-1))= 1.0
      IF(L.EQ.1) GO TO 27
      WRITE(13,02)NEQ,LPS(1,4,L-1),AIJ(NEQ,LPS(1,4,L-1))
      IF (LANS.EQ.0) GO TO 27
      PRINT 02,NEQ,LPS(1,4,L-1),AIJ(NEQ,LPS(1,4,L-1))

C
C** INPUT THE RIGHT HAND SIDE OF THE EQUATION
C
27      RHS(NEQ)= 0.0
      WRITE(13,08)NEQ,RHS(NEQ)
      IF(LANS.EQ.0) GO TO 10
      PRINT 08,NEQ,RHS(NEQ)

C.....
10      CONTINUE
C-3
C** SET UP EQUATIONS FOR THE SHIPMENT
C** OF UNITS TO THE APOD VIA AIRBRN DELIVERY
C
07      IF(LMT(2).EQ.0.OR.LMT(3).EQ.0) GO TO 48
      PRINT *,'/-----/'
      PRINT *,'/          UNIT SHIPMENT AIRBRN/'
      PRINT *,'/-----/'
      WRITE(13,*)'/-----/'
      WRITE(13,*)'/          UNIT SHIPMENT AIRBRN/'
      WRITE(13,*)'/-----/'

C** FOR EACH PERIOD
      DO 30 L=1,NP

C
C** FOR OUT, OVER, AND BULK CARGO
C
      DO 32 K=1,3
      NEQ=NEQ+1

C.....
C** FOR EACH TYPE OF AIRCRAFT
      DO 34 I=1,NMAC

C
C** CHECK TO SEE IF BEYOND THE LIMITS OF EITHER THE TYPE
C** OF MISSION L(1,1), TYPE OF CARGO L(1,2), OR FP
C
      IF(4.LT.LL(1,1) .OR. 4.GT.LU(1,1)) GO TO 34
      IF(K.LT.LL(1,2) .OR. K.GT.LU(1,2)) GO TO 34
      IF(L.LT.NFP(1)) GO TO 34
      AIJ(NEQ,LPX(1,4,K,L))= KARG(1,K,1)
      WRITE(13,04)NEQ,LPX(1,4,K,L),AIJ(NEQ,LPX(1,4,K,L)),
1          NAME(1),MS(4),KG(K),L

```

```

        IF (LANS.EQ.0) GO TO 34
        PRINT 04,NEQ,LPU(I,4,K,L),AIJ(NEQ,LPU(I,4,K,L),
1      NAME(I),MS(4),KG(K),L
34      CONTINUE
C
C** FOR EACH TYPE OF UNIT AIRBORNE CAPABLE
C
        DO 35 I=1,NUNITS+1
            IF(IAB(I).EQ.0) GO TO 36
            AIJ(NEQ,LPU(I,1,L))= -NTON(I,K)
            WRITE(13,03)NEQ,LPU(I,1,L),AIJ(NEQ,LPU(I,1,L)),
1      UNAME(I),MD(I),L
            IF (LANS.EQ.0) GO TO 36
            PRINT 03,NEQ,LPU(I,1,L),AIJ(NEQ,LPU(I,1,L)),
1      UNAME(I),MD(I),L
36      CONTINUE
C
C** THE EXCESS FOR THIS PERIOD
C
            AIJ(NEQ,LPS(3,K,L))= -1.0
            WRITE(13,02)NEQ,LPS(3,K,L),AIJ(NEQ,LPS(3,K,L))
            IF (LANS.EQ.0) GO TO 38
            PRINT 02,NEQ,LPS(3,K,L),AIJ(NEQ,LPS(3,K,L))
C
C** THE EXCESS FROM THE PREVIOUS PERIOD
C
38      IF(L.NE.1) AIJ(NEQ,LPS(3,K,L-1))= 1.0
            IF(L.EQ.1) GO TO 39
            WRITE(13,02)NEQ,LPS(3,K,L-1),AIJ(NEQ,LPS(3,K,L-1))
            IF (LANS.EQ.0) GO TO 39
            PRINT 02,NEQ,LPS(3,K,L-1),AIJ(NEQ,LPS(3,K,L-1))
C
C** INPUT THE RIGHT HAND SIDE OF THE EQUATION
C
39      RHS(NEQ)= 0.0
            WRITE(13,08)NEQ,RHS(NEQ)
            IF(LANS.EQ.0) GO TO 32
            PRINT 08,NEQ,RHS(NEQ)
C.....
32      CONTINUE
C
C** FOR PASSENGERS
C
            NEQ=NEQ+1
C.....
C** FOR EACH TYPE OF AIRCRAFT
C
            DO 40 I=1,NMAC
C
C** CHECK TO SEE IF BEYOND THE LIMITS OF EITHER THE TYPE
C** OF MISSION L01,1 OR NFP
C
                IF(4.LT.LU(I,1) .OR. 4.GT.LU(I,1)) GO TO 40
                IF(L.LT.NFP(I)) GO TO 40
                DO 42 K=L01(I,2),LU(I,2)
                    AIJ(NEQ,LPU(I,4,K,L))= KARG(I,K,2)

```

```

WRITE(13,04)NEQ,LPS(1,4,8,L),AIJ(NEQ,LPS(1,4,8,L),
1      NAME(I),MS(4),KG(K),L
IF (LANS.EQ.0) GO TO 43
PRINT 04,NEQ,LPS(1,4,8,L),AIJ(NEQ,LPS(1,4,8,L),
1      NAME(I),MS(4),KG(K),L
C
C** CONSIDER SUPPLY CARGO IF BULK CARGO IS CONSIDERED
C
43  IF(K.EQ.3) THEN
      AIJ(NEQ,LPS(1,4,5,L))=KARG(1,5,1)
      WRITE(13,04)NEQ,LPS(1,4,5,L),AIJ(NEQ,LPS(1,4,5,L),
1      NAME(I),MS(4),KG(5),L
      IF(LANS.EQ.0) GO TO 42
    ELSE
      GO TO 42
    END IF
      PRINT 04,NEQ,LPS(1,4,5,L),AIJ(NEQ,LPS(1,4,5,L),
1      NAME(I),MS(4),KG(5),L
42  CONTINUE
40  CONTINUE
C
C** FOR EACH TYPE OF UNIT AIRBORNE CAPABLE
C
DO 44 I=1,IYP
  IF(IAB(I).EQ.0) GO TO 44
  AIJ(NEQ,LPU(I,1,L))= -NTON(I,4)
  WRITE(13,03)NEQ,LPU(I,1,L),AIJ(NEQ,LPU(I,1,L),
1      UNAME(I),MD(I),L
  IF (LANS.EQ.0) GO TO 44
  PRINT 03,NEQ,LPU(I,1,L),AIJ(NEQ,LPU(I,1,L),
1      UNAME(I),MD(I),L
44  CONTINUE
C
C** THE EXCESS FOR THIS PERIOD
C
  AIJ(NEQ,LPS(3,4,L))= -1.0
  WRITE(13,02)NEQ,LPS(3,4,L),AIJ(NEQ,LPS(3,4,L))
  IF (LANS.EQ.0) GO TO 46
  PRINT 02,NEQ,LPS(3,4,L),AIJ(NEQ,LPS(3,4,L))
C
C** THE EXCESS FROM THE PREVIOUS PERIOD
C
46  IF(L.NE.1) AIJ(NEQ,LPS(3,4,L-1))= 1.0
  IF(L.EQ.1) GO TO 47
  WRITE(13,02)NEQ,LPS(3,4,L-1),AIJ(NEQ,LPS(3,4,L-1))
  IF (LANS.EQ.0) GO TO 47
  PRINT 02,NEQ,LPS(3,4,L-1),AIJ(NEQ,LPS(3,4,L-1))
C
C** INPUT THE RIGHT HAND SIDE OF THE EQUATION
C
47  RHS(NEQ)= 0.0
  WRITE(13,05)NEQ,RHS(NEQ)
  IF(LANS.EQ.0) GO TO 30
  PRINT 05,NEQ,RHS(NEQ)
C.....

```

```

30  CONTINUE
C
C++  SET UP EQUATIONS FOR THE SHIPMENT
C++  OF SUPPLIES THE THEATER
C
48  PRINT *, '-----'
    PRINT *, '          THEATER  SUPPLIES'
    PRINT *, '-----'
    WRITE(13,*) '-----'
    WRITE(13,*) '          THEATER  SUPPLIES'
    WRITE(13,*) '-----'
C
C++  FOR EACH PERIOD AFTER THE INITIAL PERIOD OF DEPLOYMENT
C
    DO 49 L=2,NP
      NEQ=NEQ+1
C.....
C++  FOR EACH TYPE OF AIRCRAFT
      DO 50 I=1,NMAC
C
C++  FOR EACH MISSION EXCEPT INTRA
C
        DO 51 J=LL(I,1),LUX(I,1)
          IF(J.EQ.3) GO TO 51
C
C++  CHECK TO SEE IF BEYOND THE LIMITS OF
C++  EITHER THE TYPE OF CARGO L(I,2), OR FF
C
          IF(3.LT.LL(I,2) .OR. 3.GT.LUX(I,2)) GO TO 51
          IF(L.LT.NFP(I)) GO TO 51
          AIJ(NEQ,LPX(I,J,5,L))= KARG(I,5,1)
          WRITE(13,04)NEQ,LPX(I,J,5,L),AIJ(NEQ,LPX(I,J,5,L)),
1            NAME(I),MS(J),KG(5),L
          IF (LANS.EQ.0) GO TO 51
          PRINT 04,NEQ,LPX(I,J,5,L),AIJ(NEQ,LPX(I,J,5,L)),
1            NAME(I),MS(J),KG(5),L
C
51      CONTINUE
50    CONTINUE
C++  FOR EACH TYPE OF UNIT
      DO 52 I=1,NUNITS+1
C
C++  FOR EACH MODE- VIA FRONT & APCD
C
        DO 53 M=1,3
          IF(1.GT.IYP.AND.1.LE.NUNITS.AND.M.NE.2) GO TO 53
          IF(LMT(1).LT.1.AND.IAB(1).EQ.0.AND.M.EQ.1) GO TO 53
          IF(LMT(2).EQ.0.AND.M.EQ.1) GO TO 53
          IF(LMT(3).EQ.0.AND.M.EQ.1) GO TO 53
          IF(IAB(1).EQ.1.AND.M.NE.1) GO TO 53
C
C++  FOR EACH DEPLOYED UNIT BEFORE L
C
        DO 55 IL=1,L-1
          AIJ(NEQ,LPU(I,M,IL))= -INTCN(I,5)*LP
          WRITE(13,03)NEQ,LPU(I,M,IL),AIJ(NEQ,LPU(I,M,IL)),

```

```

1          UNAME(1),MD(M),IL
          IF (LANS.EQ.0) GO TO 55
          PRINT 03,NEQ,LPS(1,M,IL),AIJ(NEQ,LPS(1,M,IL)),
1          UNAME(1),MD(M),IL
55          CONTINUE
53          CONTINUE
52          CONTINUE
C
C** THE EXCESS FOR THIS PERIOD
C
          AIJ(NEQ,LPS(1,5,L))= -1.0
          WRITE(13,02)NEQ,LPS(1,5,L),AIJ(NEQ,LPS(1,5,L))
          IF (LANS.EQ.0) GO TO 54
          PRINT 02,NEQ,LPS(1,5,L),AIJ(NEQ,LPS(1,5,L))
C
C** THE EXCESS FROM THE PREVIOUS PERIOD
C
54          IF(L.NE.1) AIJ(NEQ,LPS(1,5,L-1))= 1.0
          IF(L.EQ.1) GO TO 56
          WRITE(13,02)NEQ,LPS(1,5,L-1),AIJ(NEQ,LPS(1,5,L-1))
          IF (LANS.EQ.0) GO TO 56
          PRINT 02,NEQ,LPS(1,5,L-1),AIJ(NEQ,LPS(1,5,L-1))
C
C** INPUT THE RIGHT HAND SIDE OF THE EQUATION
C
56          RHS(NEQ)= 0.0
          WRITE(13,08)NEQ,RHS(NEQ)
          IF(LANS.EQ.0) GO TO 49
          PRINT 03,NEQ,RHS(NEQ)
C.....
49          CONTINUE
C-5
C** SET UP EQUATIONS FOR THE SHIPMENT
C** OF SUPPLIES TO THE FRONT
C
          PRINT *,'/-----/'
          PRINT *,          FRONT    SUPPLIES'
          PRINT *,'/-----/'
          WRITE(13,*)'/-----/'
          WRITE(13,*)          FRONT    SUPPLIES'
          WRITE(13,*)'/-----/'
C** FOR EACH PERIOD
          DO 59 L=2,NP
              NEQ=NEQ+1
C.....
C** FOR EACH TYPE OF AIRCRAFT
              DO 60 I=1,NMPC
C
C** FOR EACH MISSION EXCEPT AFOD & INTRA DELIVERED
C
                  DO 61 J=LL(1,1),LD(1,1)
                      IF(J.EQ.2) GO TO 61
                      IF(J.EQ.3) GO TO 61
C
C** CHECK TO SEE IF BEYOND THE LIMITS OF EITHER THE CARGO

```

```

C++ L=1,2, OR FR NFR 1)
C
      IF (3.LT.LL(1,2) .OR. 3.GT.LU(1,2)) GO TO 61
      IF (1.LT.NFR(1)) GO TO 61
      AIJ(NEQ,LPX(1,J,5,L))= KARG(1,5,1)
      WRITE(13,04)NEQ,LPX(1,J,5,L),AIJ(NEQ,LPX(1,J,5,L)),
1      NAME(1),MS(J),KG(5),L
      IF (LANS.EQ.0) GO TO 61
      PRINT 04,NEQ,LPX(1,J,5,L),AIJ(NEQ,LPX(1,J,5,L)),
1      NAME(1),MS(J),KG(5),L
61      CONTINUE
62      CONTINUE
-
C++ THE REQUIREMENT OF SUPPLIES FOR EACH TYPE OF UNIT AT
C++ THE FRONT (I=1...IYP      M=1,2,3; PERIOD=1...L-1)
C
      DO 62 I=1,NUNITS+1
      IF (1.GT.IYP.AND.1.LE.NUNITS) GO TO 62
      DO 65 M=1,3
      IF (LMT(1).LT.1.AND.IAB(1).EQ.0.AND.M.EQ.1) GO TO 65
      IF (LMT(2).EQ.0.AND.M.EQ.1) GO TO 65
      IF (LMT(3).EQ.0.AND.M.EQ.1) GO TO 65
      IF (IAB(1).EQ.1.AND.M.NE.1) GO TO 65
      DO 63 IL=1,L-1
      AIJ(NEQ,LPU(1,M,IL))= -NTON(1,5)*LP
      WRITE(13,03)NEQ,LPU(1,M,IL),AIJ(NEQ,LPU(1,M,IL)),
1      UNAME(1),MD(M),IL
      IF (LANS.EQ.0) GO TO 63
      PRINT 03,NEQ,LPU(1,M,IL),AIJ(NEQ,LPU(1,M,IL)),
1      UNAME(1),MD(M),IL
63      CONTINUE
64      CONTINUE
62      CONTINUE
-
C++ CONSIDER THE SL RESTRICTION FOR TRUCKING
C
      AIJ(NEQ,LPSL(1))= 1.0
      WRITE(13,02)NEQ,LPSL(1),AIJ(NEQ,LPSL(1))
      IF (LANS.EQ.1) PRINT 02,NEQ,LPSL(1),AIJ(NEQ,LPSL(1))
C
C++ THE EXCESS FOR THIS PERIOD
C
      AIJ(NEQ,LPS(2,5,L))= -1.0
      WRITE(13,02)NEQ,LPS(2,5,L),AIJ(NEQ,LPS(2,5,L))
      IF (LANS.EQ.1) PRINT 02,NEQ,LPS(2,5,L),
1      AIJ(NEQ,LPS(2,5,L))
C
C++ THE EXCESS FROM THE PREVIOUS PERIOD
C
64      IF (L.NE.1) AIJ(NEQ,LPS(2,5,L-1))= 1.0
      IF (L.EQ.1) GO TO 66
      WRITE(13,02)NEQ,LPS(2,5,L-1),AIJ(NEQ,LPS(2,5,L-1))
      IF (LANS.EQ.0) GO TO 66
      PRINT 02,NEQ,LPS(2,5,L-1),AIJ(NEQ,LPS(2,5,L-1))
C

```

```

C++ INPUT THE RIGHT HAND SIDE OF THE EQUATION.
C
DO
    RHEXNEQ = 0.0
    WRITE (3,08)NEQ,RHEXNEQ
    IF(LANG.EQ.0) GO TO 59
    PRINT 03,NEQ,RHEXNEQ
C.....
59 CONTINUE
C-5
C++ SET UP EQUATIONS FOR THE SHIPMENT
C++ OF SUPPLIES TO THE APOD
C
    PRINT *, '-----'
    PRINT *, '          APOD  SUPPLIES'
    PRINT *, '-----'
    WRITE(13,*)
    WRITE(13,*) '          APOD  SUPPLIES'
    WRITE(13,*)
C++ FOR EACH PERIOD
DO 69 L=2,NP
    NEQ=NEQ+1
C.....
C++ FOR EACH TYPE OF AIRCRAFT
DO 70 I=1,NMAC
C
C++ FOR EACH APOD MISSION, J=2
C++ CHECK TO SEE IF BEYOND THE LIMITS OF EITHER THE
C++ TYPE OF MISSION LL(1,1), CARGO L(1,2), OR FP
C
    IF(2.LT.LL(1,1) .OR. 2.GT.LU(1,1)) GO TO 70
    IF(3.LT.LL(1,2) .OR. 3.GT.LU(1,2)) GO TO 70
    IF(4.LT.NFP(1)) GO TO 70
    AJOINED,LPU(1,2,5,L)= KARG(1,5,1)
    WRITE (13,04)NEQ,LPU(1,2,5,L),AJOINED,LPU(1,2,5,L),
1      NAME(1),MD(2),KG(5),L
    IF (LANG.EQ.0) GO TO 70
    PRINT 04,NEQ,LPU(1,2,5,L),AJOINED,LPU(1,2,5,L),
1      NAME(1),MD(2),KG(5),L
70 CONTINUE
C
C++ FOR EACH TYPE OF UNIT AT THE APOD (IYP+1 TO NUNITS)
C
DO 73 I=IYP+1,NUNITS
C
C++ FOR MODES VIA APOD, M=2
C++ FOR EACH OF THE PREVIOUS PERIODS
C
DO 75 IL=1,L-1
    AJOINED,LPU(I,2,IL)= -NTON(1,5)+LF
    WRITE (13,03)NEQ,LPU(I,2,IL),AJOINED,LPU(I,2,IL),
1      NAME(I),MD(2),IL
    IF (LANG.EQ.0) GO TO 75
    PRINT 03,NEQ,LPU(I,2,IL),AJOINED,LPU(I,2,IL),
1      NAME(I),MD(2),IL
75 CONTINUE

```

```

PRINT *,-----
WRITE(13,*)-----
WRITE(13,*)      UNIT LINKAGE  CEILING
WRITE(13,*)-----
NEQ=NEQ+1
C.....
C++  FOR ALL UNITS
      DO 190 I=1,NUNITS
C++  FOR EACH MODE
      DO 192 M=1,3
        IF(I.GT.IYP.AND.1.LE.NUNITS.AND.M.NE.2) GO TO 192
        IF(LMT(I).LT.1.AND.IAB(I).EQ.0.AND.M.EQ.1) GO TO 192
        IF(LMT(2).EQ.0.AND.M.EQ.1) GO TO 192
        IF(LMT(3).EQ.0.AND.M.EQ.1) GO TO 192
        IF(IAB(I).EQ.1.AND.M.NE.1) GO TO 192
C++  FOR ALL THE PERIODS
        DO 194 L=1,NP
          AIJ(NEQ,LPU(I,M,L))= -NCI(I)
          WRITE(13,08)NEQ,LPU(I,M,L),AIJ(NEQ,LPU(I,M,L)),
1            UNAME(I),MD(M),L
          IF (LANS.EQ.0) GO TO 194
          PRINT 08,NEQ,LPU(I,M,L),AIJ(NEQ,LPU(I,M,L)),
1            UNAME(I),MD(M),L
194      CONTINUE
192      CONTINUE
190      CONTINUE
C++  INPUT THE RIGHT HAND SIDE OF THE EQUATION
      RHS(NEQ)= 0.0
      WRITE(13,09)NEQ,RHS(NEQ)
      IF(LANS.EQ.0) GO TO 196
      PRINT 08,NEQ,RHS(NEQ)
C.....
196  CONTINUE
C-13
C++  EQUATIONS FOR UNIT LINKAGE  FLOOR
C
      PRINT *,-----
      PRINT *,      UNIT LINKAGE  FLOOR
      PRINT *,-----
      WRITE(13,*)-----
      WRITE(13,*)      UNIT LINKAGE  FLOOR
      WRITE(13,*)-----
      IF(NHQ(1).EQ.0) RETURN
      IF(NHQ(2).EQ.0) THEN
        NHQ=1
      ELSE
        NHQ=2
      END IF
      DO 197 K=1,NHQ
        NEQ=NEQ+1
C.....
C++  FOR COMBAT AND HQ UNITS
      DO 198 I=NHQ+1,NHQK
C++  FOR EACH MODE
      DO 200 M=1,3

```

```

      IF (3.0*LT.LL(I,1) .OR. 3.0*LT.LUX(I,1)) GO TO 182
      IF (3.0*LT.LL(I,2) .OR. 3.0*LT.LUX(I,2)) GO TO 182
      COEF=KARS(I,5,1)
      AIJ(NEQ,LPX(I,3,5,L))= -COEF
      WRITE(13,04)NEQ,LPX(I,3,5,L),AIJ(NEQ,LPX(I,3,5,L)),
1      NAME(I),MS(3),KG(5),L
      IF (LANG.EQ.0) GO TO 182
      PRINT 04,NEQ,LPX(I,3,5,L),AIJ(NEQ,LPX(I,3,5,L)),
1      NAME(I),MS(3),KG(5),L
182      CONTINUE
C
C++ FOR EACH TRANSPORTATION UNIT AT THE APOD
C
      DO 184 I=1,NUNITS
C
C++ FOR DELIVERY TO THE APOD (M=2) BUT TRUCKED TO THE FRONT
C
      IF (NTP(I).EQ.0) GO TO 184
C
C++ TIME TO TRAVEL FROM THE APOD TO THE FRONT=MY
C
      M=(UNIT-DIST(2)/REAL(NTP(I)+LP))
      IF (M.GT.L) GO TO 184
      DO 186 IL=1,L-MY
      COEF= NTP(I)+LP)/REAL(2+DIST(2))
      AIJ(NEQ,LPU(I,2,IL))= -COEF
      WRITE(13,03)NEQ,LPU(I,2,IL),AIJ(NEQ,LPU(I,2,IL)),
1      UNAME(I),MD(2),IL
      IF (LANG.EQ.0) GO TO 186
      PRINT 03,NEQ,LPU(I,2,IL),AIJ(NEQ,LPU(I,2,IL)),
1      UNAME(I),MD(2),IL
186      CONTINUE
184      CONTINUE
C
C++ CONSIDER THE SL RESTRICTION FOR TRUCKING
C
      AIJ(NEQ,LPSL(L))= 1.0
      WRITE(13,02)NEQ,LPSL(L),AIJ(NEQ,LPSL(L))
      IF (LANG.EQ.0) GO TO 188
      PRINT 02,NEQ,LPSL(L),AIJ(NEQ,LPSL(L))
C
C++ INPUT THE RIGHT HAND SIDE OF THE EQUATION
C
188      RHS(NEQ)= 0.0
      WRITE(13,08)NEQ,RHS(NEQ)
      IF (LANG.EQ.0) GO TO 190
      PRINT 08,NEQ,RHS(NEQ)
C.....
190      CONTINUE
C-9
C++ EQUATIONS FOR UNIT LINKAGE CEILING
C
      PRINT +,-----
      PRINT +,UNIT LINKAGE CEILING

```

```

C**  OF MISSION LX(1,1) OR NFP
C
      IF(3.LT.LL(1,1) .OR. 3.GT.LU(1,1)) GO TO 174
      IF(4.LT.NFP(1)) GO TO 174
      DO 176 K=LL(1,2),LU(1,2)
        AIJ(NEQ,LPX(1,3,K,L))= KARS(1,K,2)
        WRITE(13,04)NEQ,LPX(1,3,K,L),AIJ(NEQ,LPX(1,3,K,L)),
1          NAME(1),MS(3),KG(K),L
        IF (LANS.EQ.0) GO TO 176
        PRINT 04,NEQ,LPX(1,3,K,L),AIJ(NEQ,LPX(1,3,K,L)),
1          NAME(1),MS(3),KG(K),L
176      CONTINUE
174      CONTINUE
C**  FOR EACH TYPE OF UNIT
      DO 178 I=1,NUNIT3+1
        J=3
        IF(1.GT.IYP.AND.J.NE.2.AND.I.LE.NUNIT3) GO TO 178
        IF(IAB(I).EQ.1) GO TO 178
        AIJ(NEQ,LPU(1,J,L))= -NTON(1,4)
        WRITE(13,03)NEQ,LPU(1,J,L),AIJ(NEQ,LPU(1,J,L)),
1          UNAME(1),MD(J),L
        IF (LANS.EQ.0) GO TO 178
        PRINT 03,NEQ,LPU(1,J,L),AIJ(NEQ,LPU(1,J,L)),
1          UNAME(1),MD(J),L
178      CONTINUE
C
C**  INPUT THE RIGHT HAND SIDE OF THE EQUATION
C
      RHS(NEQ)= 0.0
      WRITE(13,08)NEQ,RHS(NEQ)
      IF(LANS.EQ.0) GO TO 166
      PRINT 08,NEQ,RHS(NEQ)
C.....
166  CONTINUE
C-3
C**  EQUATIONS FOR THE SHIPMENT OF SUPPLIES
C**  TO THE FRONT VIA THE APOD
C
      PRINT +,/'-----/'
      PRINT +,/'  FRONT SUPPLIES -VIA APOD/'
      PRINT +,/'-----/'
      WRITE(13,*)/'-----/'
      WRITE(13,*)/'  FRONT SUPPLIES -VIA APOD/'
      WRITE(13,*)/'-----/'
C**  FOR EACH PERIOD
      DO 180 L=2,NP
        NEQ=NEQ+1
C.....
C**  FOR EACH TYPE OF AIRCRAFT AVAILABLE
      DO 182 I=1,NMAC
        IF(4.LT.NFP(1)) GO TO 182
C
C**  FOR THE INTRATHEATER MISSIONS(J=3), WITH SUPPLIES(J=3).
C**  CHECK TO SEE IF BEYOND THE LIMITS OF EITHER THE
C**  TYPE OF MISSION LX(1,1) OR CARGO LX(1,2)

```

```

C
      DO 168 K=1,3
      NEQ=NEQ+1
C.....
C**  FOR EACH TYPE OF AIRCRAFT
C
      DO 170 I=1,NMAC
C
C**  CHECK TO SEE IF BEYOND THE LIMITS OF EITHER THE TYPE
C**    OF MISSION L(I,1), TYPE OF CARGO L(I,2), OR FP
C
      IF(3.LT.LL(I,1) .OR. 3.GT.LUX(I,1)) GO TO 170
      IF(K.LT.LL(I,2) .OR. K.GT.LUX(I,2)) GO TO 170
      IF(0.LT.NFP(I)) GO TO 170
      AIJ(NEQ,LPX(I,3,K,L))= -KARG(I,K,1)
      WRITE(13,04)NEQ,LPX(I,3,K,L),AIJ(NEQ,LPX(I,3,K,L)),
1      NAME(I),MS(K),KG(K),L
      IF (LANS.EQ.0) GO TO 170
      PRINT 04,NEQ,LPX(I,3,K,L),AIJ(NEQ,LPX(I,3,K,L)),
1      NAME(I),MS(K),KG(K),L
170      CONTINUE
C
C**  FOR EACH TYPE OF UNIT SHIPPED TO THE APOD, J=2 FOR THOSE REMAINING
C**  AT THE APOD OR WALKING TO THE FRONT, J=3 FOR THOSE UNITS DEPLOYED
C**  TO THE FRONT BUT FLYING INTRATHEATER AIRLIFT
C
      DO 172 I=1,NUNITS+1
      J=3
      IF(I.GT.IYP.AND.J.NE.2.AND.I.LE.NUNITS) GO TO 172
      IF(IAB(I).EQ.1) GO TO 172
      AIJ(NEQ,LPU(I,J,L))= NTON(I,K)
      WRITE(13,03)NEQ,LPU(I,J,L),AIJ(NEQ,LPU(I,J,L)),
1      UNAME(I),MD(J),L
      IF (LANS.EQ.0) GO TO 172
      PRINT 03,NEQ,LPU(I,J,L),AIJ(NEQ,LPU(I,J,L)),
1      UNAME(I),MD(J),L
172      CONTINUE
C
C**  INPUT THE RIGHT HAND SIDE OF THE EQUATION
C
      RHS(NEQ)= 0.0
      WRITE(13,08)NEQ,RHS(NEQ)
      IF(LANS.EQ.0) GO TO 168
      PRINT 08,NEQ,RHS(NEQ)
C.....
168      CONTINUE
C
C**  FOR PASSENGERS
C
      NEQ=NEQ+1
C.....
C**  FOR EACH TYPE OF AIRCRAFT
      DO 174 I=1,NMAC
C
C**  CHECK TO SEE IF BEYOND THE LIMITS OF EITHER THE TYPE

```

```

        PRINT 08,NEQ,LPU(I,M,L),AIJ(NEQ,LPU(I,M,L)),
1      UNAME(I),MD(M),L
153      CONTINUE
154      CONTINUE
        RHS(NEQ)=NUNT(I)
        WRITE(13,08)NEQ,RHS(NEQ)
        IF (LANS.EQ.0) GO TO 154
        PRINT 08,NEQ,RHS(NEQ)
C.....
154      CONTINUE
C
C++  FOR EACH ALCE UNIT
C
        NEQ=NEQ+1
        DO 157 I=NUNITS,NUNITS+1
C.....
C++  FOR EACH MODE
        DO 160 M=1,3
            IF(I.GT.IYP.AND.I.LE.NUNITS.AND.M.NE.2) GO TO 160
            IF(LMT(1).LT.1.AND.IAB(I).EQ.0.AND.M.EQ.1) GO TO 160
            IF(LMT(2).EQ.0.AND.M.EQ.1) GO TO 160
            IF(LMT(3).EQ.0.AND.M.EQ.1) GO TO 160
C++  FOR EACH PERIOD
            DO 162 L=1,NP
                AIJ(NEQ,LPU(I,M,L))=1.0
                WRITE(13,03)NEQ,LPU(I,M,L),AIJ(NEQ,LPU(I,M,L)),
1      UNAME(I),MD(M),L
                IF (LANS.EQ.0) GO TO 162
                PRINT 03,NEQ,LPU(I,M,L),AIJ(NEQ,LPU(I,M,L)),
1      UNAME(I),MD(M),L
162      CONTINUE
160      CONTINUE
157      CONTINUE
        RHS(NEQ)=NUNT(NUNITS)
        WRITE(13,08)NEQ,RHS(NEQ)
        IF (LANS.EQ.0) GO TO 164
        PRINT 08,NEQ,RHS(NEQ)
C.....
C
C++  INEQUALITY FOR THE SHIPMENT OF UNITS FROM THE APOD TO THE FRONT.
C++  THIS INSURES THAT WHAT IS SHIPPED TO THE FRONT DOES NOT EXCEED
C++  WHAT IS AVAILABLE TO BE SHIPPED AT THE APOD.
C
164      PRINT +, '-----'
        PRINT +, 'UNIT SHIPMENT FROM APOD TO FRONT'
        PRINT +, '-----'
        WRITE(13,+,'-----'
        WRITE(13,+,'UNIT SHIPMENT FROM APOD TO FRONT'
        WRITE(13,+,'-----'
C
C++  FOR EACH PERIOD
C
        DO 166 L=1,NP
C
C++  FOR OUT, OVER, AND BULK CARGO

```

```

150      CONTINUE
C
C** IF WALK TO THE FRONT (M=2), CONSIDER TIME DISTANCE RELATIONSHIP TO
C** GET TO THE FRONT FROM THE APOD
C
      M=2
      IF(NTV(I).EQ.0) GO TO 151
      MY=INT((DIST(2))/REAL(NTV(I)*LP))
      IF(MY.GT.L) GO TO 151
      DO 153 IL=1,L-MY
          COEF=NPAL*LP
          AIJ(NEQ,LPU(I,M,IL))= -COEF
          WRITE(13,03)NEQ,LPU(I,M,IL),AIJ(NEQ,LPU(I,M,IL)),
1          UNAME(I),MD(M),IL
          IF (LANS.EQ.0) GO TO 153
          PRINT 03,NEQ,LPU(I,M,IL),AIJ(NEQ,LPU(I,M,IL)),
1          UNAME(I),MD(M),IL
153      CONTINUE
C
C** INPUT THE RIGHT HAND SIDE OF THE EQUATION
C
151      RHS(NEQ)= NHA(1)
      WRITE(13,08)NEQ,RHS(NEQ)
      IF(LANS.EQ.0) GO TO 142
      PRINT 08,NEQ,RHS(NEQ)
C.....
142      CONTINUE
C-6
C** SET UP EQUATIONS FOR THE ARMY UNITS ****
C
      PRINT *,'/-----/'
      PRINT *,'/          DEPLOYED  UNITS/'
      PRINT *,'/-----/'
      WRITE(13,*)'/-----/'
      WRITE(13,*)'/          DEPLOYED  UNITS/'
      WRITE(13,*)'/-----/'
C
C** FOR EACH UNIT EXCEPT THE ALDE UNITS
C
      DO 154 I=1,NUNITS-1
          NEQ=NEQ+1
C.....
C** FOR EACH MODE
      DO 156 M=1,3
          IF(1.GT.I/P.AND.1.LE.NUNITS.AND.M.NE.2) GO TO 156
          IF(LMT(1).LT.1.AND.IAB(1).EQ.0.AND.M.EQ.1) GO TO 156
          IF(LMT(2).EQ.0.AND.M.EQ.1) GO TO 156
          IF(LMT(3).EQ.0.AND.M.EQ.1) GO TO 156
          IF(IAB(1).EQ.1.AND.M.NE.1) GO TO 156
C** FOR EACH PERIOD
          DO 158 L=1,NP
              AIJ(NEQ,LPU(I,M,L))=1.0
              WRITE(13,03)NEQ,LPU(I,M,L),AIJ(NEQ,LPU(I,M,L)),
1              UNAME(I),MD(M),L
              IF (LANS.EQ.0) GO TO 158

```

```

C** FOR EACH MISSION TO THE FRONT, DIRECT (J=1) AND INTRA (J=3)
C
      DO 146 J=1,3,2
C
C** CHECK TO SEE IF BEYOND THE LIMITS OF
C** THE TYPE OF MISSION L(I,1)
C
      IF(J.LT.LL(I,1) .OR. J.GT.LU(I,1)) GO TO 146
      DO 148 K=LL(I,2),LU(I,2)
        COEF=EAS(I,K)*MHE(I)
        AIJ(NEQ,LPX(I,J,K,L))= COEF
        WRITE(13,04)NEQ,LPX(I,J,K,L),AIJ(NEQ,LPX(I,J,K,L)),
1          NAME(I),MS(J),KG(K),L
        IF (LANS.EQ.0) GO TO 149
        PRINT 04,NEQ,LPX(I,J,K,L),AIJ(NEQ,LPX(I,J,K,L)),
1          NAME(I),MS(J),KG(K),L
C
C** CONSIDER SUPPLY CARGO IF BULK CARGO IS CONSIDERED
C
149      IF(K.EQ.3) THEN
        AIJ(NEQ,LPX(I,J,5,L))=COEF
        WRITE(13,04)NEQ,LPX(I,J,5,L),AIJ(NEQ,LPX(I,J,5,L)),
1          NAME(I),MS(J),KG(5),L
        IF(LANS.EQ.0) GO TO 148
      ELSE
        GO TO 148
      END IF
      PRINT 04,NEQ,LPX(I,J,5,L),AIJ(NEQ,LPX(I,J,5,L)),
1          NAME(I),MS(J),KG(5),L
148      CONTINUE
146      CONTINUE
144      CONTINUE
C
C** FOR THE ALCE UNIT AT THE FRONT
C
      I=NUNITS+1
C
C** FOR EACH MEANS OF GETTING TO THE FRONT: 1- DIRECT
C** AND 3- APOD & FLY TO THE FRONT
C
      DO 150 M=1,3,2
        IF(LMT(I).LT.1.AND.M.EQ.1) GO TO 150
        DO 152 IL=1,L
          IF(IL.EQ.L) THEN
            COEF=NPAL*LP*0.5
          ELSE
            COEF=NPAL*LP
          END IF
          AIJ(NEQ,LPU(I,M,IL))= -COEF
          WRITE(13,03)NEQ,LPU(I,M,IL),AIJ(NEQ,LPU(I,M,IL)),
1          UNAME(I),MD(M),IL
          IF (LANS.EQ.0) GO TO 152
          PRINT 03,NEQ,LPU(I,M,IL),AIJ(NEQ,LPU(I,M,IL)),
1          UNAME(I),MD(M),IL
152      CONTINUE

```

```

        ELSE
          GO TO 138
        END IF
        PRINT 04,NEQ,LPX(1,J,5,L),AIJ(NEQ,LPX(1,J,5,L)),
1        NAME(1),MS(J),KG(5),L
138      CONTINUE
136      CONTINUE
134      CONTINUE
C
C** FOR THE ALCE UNIT(LAST TERM IN THE FILE)=NUNITS
C
      I=NUNITS
C
C** FOR THE APCD MODE=2
C** FOR ALL PREVIOUS AND NEWLY DEPLOYED ALCE UNITS
C
      DO 140 IL=1,L
        IF(IL.EQ.L) THEN
          COEF=NPAL*LP*0.5
        ELSE
          COEF=NPAL*LP
        END IF
        AIJ(NEQ,LPU(1,2,IL))=-COEF
        WRITE(13,03)NEQ,LPU(1,2,IL),AIJ(NEQ,LPU(1,2,IL)),
1        UNAME(1),MD(2),IL
        IF (LANS.EQ.0) GO TO 140
        PRINT 03,NEQ,LPU(1,2,IL),AIJ(NEQ,LPU(1,2,IL)),
1        UNAME(1),MD(2),IL
140      CONTINUE
C
C** INPUT THE RIGHT HAND SIDE OF THE EQUATION
C
      RHS(NEQ)=NHA(2)
      WRITE(13,08)NEQ,RHS(NEQ)
      IF(LANS.EQ.0) GO TO 132
      PRINT 08,NEQ,RHS(NEQ)
C.....
132    CONTINUE
C
C** EQUATIONS FOR THE MHE LIMITATION AT FRONT
C-5
      PRINT *,'/-----/'
      PRINT *,'/      MHE  LIMITATION-FRONT/'
      PRINT *,'/-----/'
      WRITE(13,*)'/-----/'
      WRITE(13,*)'/      MHE  LIMITATION-FRONT/'
      WRITE(13,*)'/-----/'
C** FOR EACH PERIOD
      DO 142 L=1,NP
        NEQ=NEQ+1
C.....
C** FOR EACH TYPE OF AIRCRAFT AVAILABLE
      DO 144 I=1,NMAC
        IF(L.LT.NFP(1)) GO TO 144
C

```

```

      COEF=REAL(10000*NTAC(I))/REAL(LP*NPX(I))
      AIJ(NEQ,LPU(I,2,L))=COEF
      WRITE(13,03)NEQ,LPU(I,2,L),AIJ(NEQ,LPU(I,2,L)),
1      UNAME(I),MD(2),L
      IF(LANS.EQ.0) GO TO 130
      PRINT 03,NEQ,LPU(I,2,L),AIJ(NEQ,LPU(I,2,L)),
1      UNAME(I),MD(2),L
130    CONTINUE
      RHS(NEQ)=10000
      WRITE(13,08)NEQ,RHS(NEQ)
      IF (LANS.EQ.0) GO TO 122
      PRINT 09,NEQ,RHS(NEQ)
C.....
122    CONTINUE
C-4
C** EQUATIONS FOR THE MHE LIMITATION AT AP0D
C
      PRINT *,'/-----/'
      PRINT *,'/          MHE  LIMITATION-AP0D/'
      PRINT *,'/-----/'
      WRITE(13,*)'/-----/'
      WRITE(13,*)'/          MHE  LIMITATION-AP0D/'
      WRITE(13,*)'/-----/'
C** FOR EACH PERIOD
      DO 132 L=1,NP
        NEQ=NEQ+1
C.....
C** FOR EACH TYPE OF AIRCRAFT AVAILABLE
        DO 134 I=1,NMAC
          IF(L.LT.NFP(I)) GO TO 134
C
C** FOR EACH MISSION INTO THE AP0D(J=2,3)
C
          DO 136 J=2,3
C
C** CHECK TO SEE IF BEYOND THE LIMITS OF
C** THE TYPE OF MISSION L(I,1)
C
            IF(J.LT.LL(I,1) .OR. J.GT.LU(I,1)) GO TO 136
            DO 138 K=LL(I,2),LU(I,2)
              COEF=EAS(I,K)*MHE(I)
              AIJ(NEQ,LPX(I,J,K,L))= COEF
              WRITE(13,04)NEQ,LPX(I,J,K,L),AIJ(NEQ,LPX(I,J,K,L)),
1              NAME(I),MS(J),KG(K),L
              IF (LANS.EQ.0) GO TO 139
              PRINT 04,NEQ,LPX(I,J,K,L),AIJ(NEQ,LPX(I,J,K,L)),
1              NAME(I),MS(J),KG(K),L
C
C** CONSIDER SUPPLY CARGO IF BULK CARGO IS CONSIDERED
C
139      IF(K.EQ.3) THEN
              AIJ(NEQ,LPX(I,J,5,L))=COEF
              WRITE(13,04)NEQ,LPX(I,J,5,L),AIJ(NEQ,LPX(I,J,5,L)),
1              NAME(I),MS(J),KG(5),L
              IF(LANS.EQ.0) GO TO 138

```

```

      RHS(NEQ)=(1000*MAC(I)*AVAIL(I)*KTS(I)*LP)/REAL(12*DIST(I))
      WRITE(13,08)NEQ,RHS(NEQ)
      IF (LANS.EQ.0) GO TO 112
      PRINT 08,NEQ,RHS(NEQ)
C.....
112    CONTINUE
110    CONTINUE
C-2
C**  SET UP EQUATIONS FOR THE AIRPORT LIMITATIONS
C
      PRINT *,'/-----/'
      PRINT *,'/      AIRPORT  LIMITATIONS/'
      PRINT *,'/-----/'
      WRITE(13,*)'/-----/'
      WRITE(13,*)'/      AIRPORT  LIMITATIONS/'
      WRITE(13,*)'/-----/'
C**  FOR EACH PERIOD
      DO 122 L=1,NP
        NEQ=NEQ+1
C.....
C**  FOR EACH MAC TYPE AIRCRAFT
C
        DO 124 I=1,NMAC
          IF (L.LT.NFP(I)) GO TO 124
C
C**  FOR EACH MISSION TRANSITING THE APOD (J=2,3)
C
          DO 126 J=2,3
            IF(J.LT.LL(I,1).OR.J.GT.LU(I,1)) GO TO 126
            COEF=10000*GT(I)/REAL(NPRK(I)*24*LP)
            DO 128 K=LL(I,2),LU(I,2)
              AIJ(NEQ,LPX(I,J,K,L))=COEF
              WRITE(13,04)NEQ,LPX(I,J,K,L),AIJ(NEQ,LPX(I,J,K,L)),
1              NAME(I),MS(J),KG(K),L
              IF (LANS.EQ.0) GO TO 129
              PRINT 04,NEQ,LPX(I,J,K,L),AIJ(NEQ,LPX(I,J,K,L)),
1              NAME(I),MS(J),KG(K),L
129          IF(K.EQ.3) THEN
              AIJ(NEQ,LPX(I,J,5,L))=COEF
              WRITE(13,04)NEQ,LPX(I,J,5,L),AIJ(NEQ,LPX(I,J,5,L)),
1              NAME(I),MS(J),KG(5),L
              IF(LANS.EQ.0) GO TO 128
            ELSE
              GO TO 129
            END IF
              PRINT 04,NEQ,LPX(I,J,5,L),AIJ(NEQ,LPX(I,J,5,L)),
1              NAME(I),MS(J),KG(5),L
123          CONTINUE
124          CONTINUE
124          CONTINUE
C
C**  FOR EACH FIGHTER TYPE AC AT THE APOD
C
          DO 130 I=1,NFIGHTS
            IF(NTAC(I).EQ.0) GO TO 130

```

```

      COEF=1000/UTE(I,3)
      DO 116 K=LL(I,2),LU(I,2)
        AIJ(NEQ,LPX(I,J,K,L))=COEF
        WRITE(13,04)NEQ,LPX(I,J,K,L),AIJ(NEQ,LPX(I,J,K,L)),
1          NAME(I),MS(J),KG(K),L
        IF (LANS.EQ.0) GO TO 117
        PRINT 04,NEQ,LPX(I,J,K,L),AIJ(NEQ,LPX(I,J,K,L)),
1          NAME(I),MS(J),KG(K),L
      C
      C** CONSIDER SUPPLY CARGO IF BULK CARGO IS CONSIDERED
      C
      117 IF(K.EQ.3) THEN
        AIJ(NEQ,LPX(I,J,5,L))=COEF
        WRITE(13,04)NEQ,LPX(I,J,5,L),AIJ(NEQ,LPX(I,J,5,L)),
1          NAME(I),MS(J),KG(5),L
        IF(LANS.EQ.0) GO TO 116
        ELSE
          GO TO 116
        END IF
        PRINT 04,NEQ,LPX(I,J,5,L),AIJ(NEQ,LPX(I,J,5,L)),
1          NAME(I),MS(J),KG(5),L
      116 CONTINUE
      114 CONTINUE
      C
      C** FOR THE INTRATHEATER MISSIONS
      C** CHECK FOR MS OUT OF LIMITS AND AC THAT SPECIALIZE IN INTRA(LL=3)
      C
        IF(LL(I,1).GT.3 .OR.LU(I,1).LT.3) GO TO 118
        IF(LL(I,1).NE.3) COEF=(1000*DIST(2))/(DIST(1)*UTE(I,3))
        IF(LL(I,1).EQ.3) COEF=(1000/UTE(I,3))
        DO 120 K=LL(I,2),LU(I,2)
          AIJ(NEQ,LPX(I,3,K,L))=COEF
          WRITE(13,04)NEQ,LPX(I,3,K,L),AIJ(NEQ,LPX(I,3,K,L)),
1            NAME(I),MS(3),KG(K),L
          IF (LANS.EQ.0) GO TO 121
          PRINT 04,NEQ,LPX(I,3,K,L),AIJ(NEQ,LPX(I,3,K,L)),
1            NAME(I),MS(3),KG(K),L
        C
        C** CONSIDER SUPPLY CARGO IF BULK CARGO IS CONSIDERED
        C
        121 IF(K.EQ.3) THEN
          AIJ(NEQ,LPX(I,3,5,L))=COEF
          WRITE(13,04)NEQ,LPX(I,3,5,L),AIJ(NEQ,LPX(I,3,5,L)),
1            NAME(I),MS(3),KG(5),L
          IF(LANS.EQ.0) GO TO 120
          ELSE
            GO TO 120
          END IF
          PRINT 04,NEQ,LPX(I,3,5,L),AIJ(NEQ,LPX(I,3,5,L)),
1            NAME(I),MS(3),KG(5),L
        120 CONTINUE
        118 CONTINUE
      C
      C** INSERT THE RHS OF THE CONSTRAINT
      C

```

```

C
SUBROUTINE CONSTR(ATTR)
C
C**
C** THIS BUILDS THE FOLLOWING INEQUALITY CONSTRAINTS INTO THE AIC MATRIX
C**      2. AIRCRAFT UTE RATE
C**      3. AIRPORT LIMITATIONS
C**      4. MHE LIMITATIONS AT THE APOD
C**      5. MHE LIMITATIONS AT THE FRONT
C**      6. POSSIBLE NUMBER OF DEPLOYED UNITS
C**      7. INTRATHEATER SHIPMENT OF UNITS FROM THE APOD TO THE FRONT
C**      8. INTRATHEATER SHIPMENT OF SUPPLIES FROM THE APOD TO THE FRONT
C**      9. UNIT LINKAGE CEILING
C**     10. UNIT LINKAGE FLOOR
C**
C
CHARACTER NAME(10)*6,UNAME(11)*14,GNAME(3)*14
CHARACTER MS(4)*5,KG(5)*4,MD(3)*5
COMMON/ACF/NMAC,NAC(10),AVAIL(10),UTE(10,4),KARG(10,5,2),KTS(10),
C   GT(10),MHE(10),NFP(10),NPRK(10)
COMMON/APD/DIST(2),TINTER(10),TINTRA(10),EAS(10,4),CPI(4),
C   NHA(2),NPAL,RC,GLS(9,3)
COMMON/ARM/NUNITS,NUNT(10),NTON(11,5),NFP(10),NAT(10),NFT(10),
C   NTAC(10),NPK(10),NTP(11),NTV(11),IAB(10),NCI(10),IYP,NHQ(2)
COMMON/TAB/NP,LP,IP,NCR,NEQ,NPRIT,LL(10,2),LUX(10,2),LPX(10,4,5,4),
C   LPU(10,3,4),AIJ(150,300),RHS(150),LPS(3,5,4),LPSL(4),LPR(10),
C   NT(150),LMT(3),NC(10),NWGT(150),NDV(150),NID
COMMON/CHR/NAME,UNAME,GNAME,MS,KG,MD
C
C** CHECK TO SEE IF ECHO DESIRED
C
*   PRINT *, 'ECHO ON THE #1 PRIORITY CONSTRAINTS?(1=Y/0=N) .'
*   READ *, LANS
LANS=0
PRINT *, '-----'
PRINT *, '          INEQUALITY CONSTRAINTS [ <= ]'
PRINT *, '-----'
C-2
C** SET UP EQUATIONS FOR THE AIRCRAFT UTE RATE **
C
PRINT *, '-----'
PRINT *, '          AIRCRAFT UTE RATE'
PRINT *, '-----'
WRITE(13,*) '-----'
WRITE(13,*) '          AIRCRAFT UTE RATE'
WRITE(13,*) '-----'
C** FOR EACH TYPE OF AIRCRAFT
DO 110 I=1,NMAC
C** FOR EACH PERIOD
DO 112 L=NFP(1),NFP
NEQ=NEQ+1
C.....
C** FOR THE INTRATHEATER MISSIONS
DO 114 J=LL(1,1),LUX(1,1)
IF (J.EQ.3) GO TO 114

```

```

1          NAME(I),MS(J),KG(K),L
C
C**  CONSIDER SUPPLY CARGO IF BULK CARGO IS CONSIDERED
C
87      IF(K.EQ.3) THEN
          AIJ(NEQ,LPX(I,J,5,L))=KARG(I,5,1)
          WRITE(13,04)NEQ,LPX(I,J,5,L),AIJ(NEQ,LPX(I,J,5,L)),
1          NAME(I),MS(J),KG(5),L
          IF(LANS.EQ.0) GO TO 85
      ELSE
          GO TO 85
      END IF
          PRINT 04,NEQ,LPX(I,J,5,L),AIJ(NEQ,LPX(I,J,5,L)),
1          NAME(I),MS(J),KG(5),L
85      CONTINUE
83      CONTINUE
82      CONTINUE
C
C**  THE EXCESS FOR THIS PERIOD
C
          AIJ(NEQ,LPR(L))= -1.0
          WRITE(13,02)NEQ,LPR(L),AIJ(NEQ,LPR(L))
          IF (LANS.EQ.0) GO TO 84
          PRINT 02,NEQ,LPR(L),AIJ(NEQ,LPR(L))
C
C**  THE EXCESS FROM THE PREVIOUS PERIOD
C
84      IF(L.NE.1) AIJ(NEQ,LPR(L-1))= 1.0
          IF(L.EQ.1) GO TO 86
          WRITE(13,02)NEQ,LPR(L-1),AIJ(NEQ,LPR(L-1))
          IF (LANS.EQ.0) GO TO 86
          PRINT 02,NEQ,LPR(L-1),AIJ(NEQ,LPR(L-1))
C
C**  INPUT THE RIGHT HAND SIDE OF THE EQUATION
C
86      RHS(NEQ)= RC*LP
          WRITE(13,08)NEQ,RHS(NEQ)
          IF(LANS.EQ.0) GO TO 80
          PRINT 08,NEQ,RHS(NEQ)
C.....
80      CONTINUE
C
          NCR=NEQ
          WRITE(13,*)'NCR=',NCR
C
C**  FORMAT STATEMENTS
C
02      FORMAT('AIJ(',2I4,',')=',F14.5)
03      FORMAT('AIJ(',2I4,',')=',F14.5,5X,A14,2X,A5,I3)
04      FORMAT('AIJ(',2I4,',')=',F14.5,5X,A6,2X,A5,2X,A4,I3)
08      FORMAT('RHS(',I4,',')=',F14.5,/, '-----')
C
C
      END
C*****

```

```

72      CONTINUE
C
C**  CONSIDER THE SL RESTRICTION FOR TRUCKING
C
      AIJ(NEQ,LPSL(L))= -1.0
      WRITE(13,02)NEQ,LPSL(L),AIJ(NEQ,LPSL(L))
      IF(LANS.EQ.1) PRINT 02,NEQ,LPSL(L),AIJ(NEQ,LPSL(L))
C
C**  THE EXCESS FOR THIS PERIOD
C
      AIJ(NEQ,LPS(3,5,L))= -1.0
      WRITE(13,02)NEQ,LPS(3,5,L),AIJ(NEQ,LPS(3,5,L))
      IF (LANS.EQ.0) GO TO 74
      PRINT 02,NEQ,LPS(3,5,L),AIJ(NEQ,LPS(3,5,L))
C
C**  THE EXCESS FROM THE PREVIOUS PERIOD
C
74      IF(L.NE.1) AIJ(NEQ,LPS(3,5,L-1))= 1.0
      IF(L.EQ.1) GO TO 76
      WRITE(13,02)NEQ,LPS(3,5,L-1),AIJ(NEQ,LPS(3,5,L-1))
      IF (LANS.EQ.0) GO TO 76
      PRINT 02,NEQ,LPS(3,5,L-1),AIJ(NEQ,LPS(3,5,L-1))
C
C**  INPUT THE RIGHT HAND SIDE OF THE EQUATION
C
76      RHS(NEQ)= 0.0
      WRITE(13,08)NEQ,RHS(NEQ)
      IF(LANS.EQ.0) GO TO 69
      PRINT 08,NEQ,RHS(NEQ)
C.....
69      CONTINUE
C
C**  EQUATIONS FOR UNIT ABILITIES
C
      PRINT *,'/-----/'
      PRINT *,'/          RIGGER  CONSTRAINT/'
      PRINT *,'/-----/'
      WRITE(13,*)'/-----/'
      WRITE(13,*)'/          RIGGER  CONSTRAINT/'
      WRITE(13,*)'/-----/'
C**  FOR EACH PERIOD
      DO 80 L=1,NP
      NEQ=NEQ+1
C.....
      DO 82 I=1,NMAC
      DO 83 J=LL(I,1),LUX(I,1)
      IF(J.NE.4) GO TO 83
      DO 85 K=LL(I,2),LUX(I,2)
      IF(K.EQ.4) GO TO 85
      IF(L.LT.NFP(I)) GO TO 85
      AIJ(NEQ,LPX(I,J,K,L))= KARG(I,K,1)
      WRITE(13,04)NEQ,LPX(I,J,K,L),AIJ(NEQ,LPX(I,J,K,L)),
1      NAME(I),MS(J),KG(K),L
      IF (LANS.EQ.0) GO TO 87
      PRINT 04,NEQ,LPX(I,J,K,L),AIJ(NEQ,LPX(I,J,K,L)),

```

```

      IF(IABK1.EQ.1.AND.M.NE.1) GO TO 200
C** FOR ALL THE PERIODS
      DO 202 L=1,NP
        IF(I.EQ.1) COEF=1
        IF(I.EQ.2) COEF=-3
        AIJ(NEQ,LPU(I,M,L))= COEF
        WRITE(13,03)NEQ,LPU(I,M,L),AIJ(NEQ,LPU(I,M,L)),
1          UNAME(I),MD(M),L
        IF (LANS.EQ.0) GO TO 202
        PRINT 03,NEQ,LPU(I,M,L),AIJ(NEQ,LPU(I,M,L)),
1          UNAME(I),MD(M),L
202      CONTINUE
200      CONTINUE
198      CONTINUE
197      CONTINUE
C
C** INPUT THE RIGHT HAND SIDE OF THE EQUATION
C
      RHS(NEQ)= 2.0
      WRITE(13,08)NEQ,RHS(NEQ)
      IF(LANS.EQ.0) GO TO 204
      PRINT 08,NEQ,RHS(NEQ)
C.....
204      CONTINUE
C
C** FORMAT STATEMENTS
C
02      FORMAT('AIJ(',2I4,',')= ',F14.5)
03      FORMAT('AIJ(',2I4,',')= ',F14.5,5X,A14,2X,A5,I3)
04      FORMAT('AIJ(',2I4,',')= ',F14.5,5X,A6,2X,A5,2X,A4,I3)
08      FORMAT ('RHS(',I4,',')= ',F14.5,/, '-----')
C
C
      LANS=0
      RETURN
      END
C*****
C**
      SUBROUTINE SORTIE
C
C** THIS BUILDS THE CONSTRAINTS FOR THE NUMBER OF POSSIBLE AIRCRAFT
C** SORTIES POSSIBLE.
C
      CHARACTER NAME(10)*6,UNAME(11)*14,GNAME(3)*14,FNAME*6
      CHARACTER MS(4)*5,KG(5)*4,MD(3)*5
      COMMON/ACF/NMAC,NAC(10),AVAIL(10),UTE(10,4),KARG(10,5,2),KTS(10),
C      GT(10),MHE(10),NFR(10),NPRK(10)
      COMMON/APD/OIST(2),TINTER(10),TINTRA(10),EAS(10,4),CPI(4),
C      NHA(2),NPAL,RC,GLS(9,3)
      COMMON/TAB/NP,LP,IP,NCR,NEQ,NPRIT,LL(10,2),LUX(10,2),LPX(10,4,5,4),
C      LPU(10,3,4),AIJ(150,300),RHS(150),LPS(3,5,4),LPSL(4),LPR(10),
C      NT(150),LMT(3),NC(10),NWGT(150),NDV(150),NID
      COMMON/CHR/NAME,UNAME,GNAME,MS,KG,MD
      LANS=0

```

D-1

```

C** SET UP EQUATIONS FOR THE AIRCRAFT SORTIES
C
      NC(2)=0
      PRINT *,'/-----/'
      PRINT *,'/      AIRCRAFT  SORTIES/'
      PRINT *,'/-----/'
      WRITE(13,*)'/-----/'
      WRITE(13,*)'/      AIRCRAFT  SORTIES/'
      WRITE(13,*)'/-----/'
C
C** FOR EACH TYPE OF AIRCRAFT
C
      DO 100 I=1,NMAC
C
C** FOR EACH PERIOD
C
      DO 102 L=NFP(1),NP
        NEQ=NEQ+1
        NC(2)=NC(2)+1
C.....
        DO 104 J=LL(I,1),LU(I,1)
C
C** FOR THE INTERTHEATER MISSIONS
C
          IF (J.NE.3) THEN
            COEF=TINTER(I)
C
C** FOR THE INTRATHEATER MISSIONS
C
            ELSE IF(J.EQ.3) THEN
              COEF=TINTRA(I)
            END IF
            DO 106 K=LL(I,2),LU(I,2)
              AIJ(NEQ,LPX(I,J,K,L))=COEF
              WRITE(13,04)NEQ,LPX(I,J,K,L),AIJ(NEQ,LPX(I,J,K,L)),
1                NAME(I),MS(J),KG(K),L
              IF (LANS.EQ.0) GO TO 107
              PRINT 04,NEQ,LPX(I,J,K,L),AIJ(NEQ,LPX(I,J,K,L)),
1                NAME(I),MS(J),KG(K),L
C
C** CONSIDER THE SUPPLY CARGO IF BULK IS CONSIDERED
C
107          IF(K.EQ.3) THEN
            AIJ(NEQ,LPX(I,J,5,L))=COEF
            WRITE(13,04)NEQ,LPX(I,J,5,L),AIJ(NEQ,LPX(I,J,5,L)),
1            NAME(I),MS(J),KG(5),L
            IF(LANS.EQ.0) GO TO 106
          ELSE
            GO TO 106
          END IF
          PRINT 04,NEQ,LPX(I,J,5,L),AIJ(NEQ,LPX(I,J,5,L)),
1            NAME(I),MS(J),KG(5),L
106          CONTINUE
104          CONTINUE
C** INPUT THE RHS

```

```

      RHS(NEQ)=NAC(1)*AVAIL(1)*LP
      WRITE(13,98)NEQ,RHS(NEQ)
      IF (LANS.EQ.0) GO TO 102
      PRINT 08,NEQ,RHS(NEQ)
C.....
  102      CONTINUE
  100      CONTINUE
C
C**  FORMAT STATEMENTS
C
  04      FORMAT('AIJ(',2I4,')= ',F14.5,3X,A6,2X,A5,2X,A4,I3)
  08      FORMAT('RHS(',I4,')= ',F14.5,/, '-----')
C
C      RETURN
C      END
C*****
C** 3
      SUBROUTINE MGOAL
C
C**  MAKES THE GOALS DESIRED BY THE USER
C
      CHARACTER NAME(10)*6,UNAME(11)*14,GNAME(3)*16,FNAME*6
      CHARACTER MS(4)*5,KG(5)*4,MD(3)*5
      COMMON/ACF/NMAC,NAC(10),AVAIL(10),UTE(10,4),KARG(10,5,2),KTS(10),
C      GT(10),MHE(10),NFP(10),NPRK(10)
      COMMON/APD/DIST(2),TINTER(10),TINTRA(10),EAS(10,4),CPI(4),
C      NHA(2),NPAL,RC,GLS(9,3)
      COMMON/ARM/NUNITS,NUNT(10),NTON(11,5),NFP(10),NAT(10),NFT(10),
C      NTAC(10),NPK(10),NTP(11),NTX(11),IAB(10),NCI(10),IYP,NHQ(2)
      COMMON/TAB/NP,LP,IP,NCR,NEQ,NPRIT,LL(10,2),LU(10,2),LPX(10,4,5,4),
C      LPU(10,3,4),AIJ(150,300),RHS(150),LPS(3,5,4),LPSL(4),LPR(10),
C      NT(150),LMT(3),NC(10),NWGT(150),NDX(150),NID
      COMMON/CHR/NAME,UNAME,GNAME,MS,KG,MD
C
C**  PRINT OUT THE INFORMATION NEEDED FOR THIS FILE
C
      PRINT 5
      PRINT *, 'THE FOLLOWING DATA IS NEEDED TO ESTABLISH THE GOALS'
      PRINT *, '-----'
      PRINT *
      PRINT *, '1. PRIORITY NUMBER (2 THROUGH 9, INTEGER)'
      PRINT *
      PRINT *, '2. TYPE OF GOAL, FROM ONE OF THE FOLLOWING:'
      PRINT *, '      A. FRONT LINE TRACE DISTANCE'
      PRINT *, '      B. ANTI-TANK STRENGTH'
      PRINT *, '      C. FIREPOWER'
      PRINT *, '      D. DEPLOYMENT OF AN ARMY UNIT'
      PRINT *
      PRINT *, '3. SPECIFIED AMOUNT OF THE TYPE GOAL (REAL)'
      PRINT *
      PRINT *, '4. PERIOD COMPLETION DESIRED (INTEGER)'
      PRINT *
      PRINT *, 'IF YOU DO NOT HAVE ALL THE ABOVE DATA, YOU MAY TERMINATE'
      PRINT *, 'OR CONTINUE.'

```

```

      PRINT *, 'DO YOU WANT TO CONTINUE? (1=Y,0=N)'
10  READ *, NRES
      IF (NRES.GT.1.OR.NRES.LT.0) THEN
          PRINT *, 'NOT A VALID REPLY, PLEASE ENTER (1=Y/0=N).'
          GO TO 10
      END IF
      IF (NRES.EQ.0) STOP
C
C** INPUT THE GOAL NAMES
C
      GNAME(1)='FRONT LINE TRACE'
      GNAME(2)='ANTI-TANK POWER'
      GNAME(3)='FIREPOWER'
C
C** START THE INPUT PROCESS
C
      I=1
12  PRINT 5
      I=I+1
      PRINT *, 'SELECT THE GOAL TYPE FOR PRIORITY ', I
      K=I*3
      DO 18 J=1,K
          IF (J.LE.3) PRINT 9,J,GNAME(J)
          IF (J.GT.3) PRINT 9,J,UNAME(J-3)
18  CONTINUE
      PRINT *, '** IF FINISHED TYPE 0'
19  READ *, GLS(I,1)
      IF (GLS(I,1).EQ.0) GO TO 20
      IF (GLS(I,1).GT.K.OR.GLS(I,1).LT.1) THEN
          PRINT *, 'NOT A VALID REPLY, PLEASE RE-ENTER.'
          GO TO 19
      END IF
      NC(I)=1
      NPRINT=1
      PRINT *, 'ENTER THE DESIRED OBJECTIVE OF THE PRIORITY.'
      READ *, GLS(I,2)
      PRINT *, 'ENTER THE PERIOD COMPLETION DESIRED.'
      READ *, GLS(I,3)
C
C** CHECK TO SEE IF THERE IS MEMORY FOR MORE GOALS
C
      IF (I.EQ.8) PRINT *, '*** YOU HAVE MEMORY FOR ONLY ONE MORE GOAL.'
      IF (I.EQ.9) THEN
          PRINT *, 'YOU ARE OUT OF MEMORY FOR GOALS.'
          GO TO 20
      END IF
      GO TO 12
C
C** ASK IF THE FILE IS TO BE VIEWED
C
20  PRINT *, 'DO YOU WANT TO SEE THE LIST OF GOALS? (1=Y/0=N)'
21  READ *, IVU
      IF (IVU.GT.1.OR.IVU.LT.0) THEN
          PRINT *, 'NOT A VALID REPLY, PLEASE ENTER (1=Y/0=N).'
          GO TO 21

```

```

      END IF
C
C** PRINT OUT THE FILE IF IVU = 1
C
      IF(IVU.EQ.1) THEN
        DO 24 J=2,NPRIT
          IF(GLS(J,1).GT.3) THEN
            K=INT(GLS(J,1)-3)
            L=INT(GLS(J,3))
            PRINT 25,J,GLS(J,2),UNAME(K),L
            GO TO 24
          ELSE
            K=INT(GLS(J,1))
            L=INT(GLS(J,3))
            PRINT 25,J,GLS(J,2),GNAME(K),L
          END IF
24    CONTINUE
      END IF
C
C** SET UP THE NUMBER OF CONSTRAINTS FOR EACH GOAL AND THE NUMBER
C** OF GOALS
C
      DO 40 I=1,NG
        NC(I)=1
40    CONTINUE
C
C** FORMAT STATEMENTS
C
      2    FORMAT(A6)
      5    FORMAT(////////////////////)
      9    FORMAT(' -- ',I2,2X,A16)
      25   FORMAT('GOAL #',I2,' IS ',F5.1,' OF ',A16,' BY PERIOD',I2)
C
      RETURN
      END
C*****
C 57
      SUBROUTINE GOALS(IDP,OBJ,ITYPE,LANG)
C
C** THIS SETS UP THE GOALS
C
      REAL OBJ
      CHARACTER NAME(10)*5,UNAME(11)*14,GNAME(3)*14
      CHARACTER MS(4)*5,KG(5)*4,MD(3)*5
      COMMON/ACF/NMAC,NAC(10),AVAIL(10),UTE(10,4),KARG(10,5,2),KTS(10),
C    GT(10),MHE(10),NFP(10),NPRK(10)
      COMMON/APD/DIST(2),TINTER(10),TINTRA(10),EAS(10,4),CPI(4),
C    NHA(2),NPAL,RC,GLS(9,3)
      COMMON/ARM/NUNITS,NUNT(10),NTON(11,5),NFP(10),NAT(10),NFT(10),
C    NTAC(10),NPK(10),NTP(11),NTV(11),IAB(10),NCI(10),IYP,NHQ(2)
      COMMON/TAB/NP,LP,IP,NCR,NEQ,NPRIT,LL(10,2),LU(10,2),LPX(10,4,5,4),
C    LPU(10,3,4),AIJ(150,300),RHS(150),LPS(3,5,4),LPSL(4),LPR(10),
C    NT(150),LMT(3),NC(10),NWGT(150),NDV(150),NID
      COMMON/CHR/NAME,UNAME,GNAME,MS,KG,MD
C

```

```

C** GO TO THE PROPER TYPE
C
  IF(IDP.GT.NP) IDP=NP
  IF(ITYPE.GT.3) THEN
    IUT=ITYPE-3
    ITYPE=4
  END IF
  GO TO (09,19,29,39)ITYPE
C
C** TYPE 1
C** SET UP EQUATIONS FOR FRONT LINE TRACE
C
  09 PRINT *,'/-----/'
  PRINT *,'/  FRONT LINE TRACE'
  PRINT *,'/-----/'
  WRITE(13,*)'/-----/'
  WRITE(13,*)'/  FRONT LINE TRACE'
  WRITE(13,*)'/-----/'
C** INCREMENT THE ROW BY 1
C
  NEQ=NEQ+1
C.....
C** FROM THE UNITS DELIVERED TO THE FRONT (M=1) AND THOSE
C** UNITS DELIVERED BY INTRATHEATER AC (M=3)
C
C** FOR EACH PERIOD UP TO THE DESIRED TIME
C
  DO 10 L=1,IDP
C
C** FOR EACH UNIT WITH NFT(I).NE.0
C
  DO 11 I=1,NUNITS+1
    IF (NFT(I).EQ.0) GO TO 11
    DO 15 M=1,3
      IF(I.GT.IYP.AND.I.LE.NUNITS.AND.M.NE.2) GO TO 15
      IF(I.LE.IYP.AND.M.EQ.2) GO TO 15
      IF(LMT(1).LT.1.AND.IAB(I).EQ.0.AND.M.EQ.1) GO TO 15
      IF(LMT(2).EQ.0.AND.M.EQ.1) GO TO 15
      IF(LMT(3).EQ.0.AND.M.EQ.1) GO TO 15
      IF(IAB(I).EQ.1.AND.M.NE.1) GO TO 15
      AIJ(NEQ,LPU(I,M,L))=NFT(I)
      WRITE(13,03)NEQ,LPU(I,M,L),AIJ(NEQ,LPU(I,M,L)),
1      UNAME(I),MD(M),L
      IF (LANS.EQ.0) GO TO 15
      PRINT 03,NEQ,LPU(I,M,L),AIJ(NEQ,LPU(I,M,L)),
1      UNAME(I),MD(M),L
15    CONTINUE
11    CONTINUE
10    CONTINUE
C
C** FOR EACH UNIT GOING TO THE FRONT (1...IYP) WITHOUT
C** AIRLIFT (M=2)
C
  DO 12 I=1,IYP
    IF(NFT(I).EQ.0.OR.NTV(I).EQ.0) GO TO 12

```

```

      IF(IAB(I).EQ.1) GO TO 12
      MY=INT((DIST(2))/REAL(NTV(1)*LP))
      IF (MY.GT.IDP) GO TO 12
C
C** FOR EACH PERIOD UNTIL IDP-MY
C
      DO 13 L=1,IDP-MY
        AIJ(NEQ,LPU(1,2,L))=NFT(I)
        WRITE(13,03)NEQ,LPU(1,2,L),AIJ(NEQ,LPU(1,2,L)),
1          UNAME(I),MD(2),L
        IF (LANS.EQ.0) GO TO 13
        PRINT 03,NEQ,LPU(1,2,L),AIJ(NEQ,LPU(1,2,L)),
1          UNAME(I),MD(2),L
13      CONTINUE
12      CONTINUE
      RHS(NEQ)=OBJ
      WRITE(13,08)NEQ,RHS(NEQ)
      IF(LANS.EQ.1) PRINT 08,NEQ,RHS(NEQ)
C.....
      RETURN
C
C** TYPE 2
C** SET UP EQUATIONS FOR ANTI TANK
C
19      PRINT *,'/-----/'
      PRINT *,'/          ANTI TANK/'
      PRINT *,'/-----/'
      WRITE(13,*)'/-----/'
      WRITE(13,*)'/          ANTI TANK/'
      WRITE(13,*)'/-----/'
C
C** INCREMENT THE ROW BY 1
C
      NEQ=NEQ+1
C.....
C** FROM THE FRONT
C** FOR EACH PERIOD UP TO THE DESIRED TIME
C
      DO 20 L=1,IDP
C
C** FOR EACH UNIT WITH NAT(I).NE.0
C
      DO 21 I=1,NUNITS+1
        IF(NAT(I).EQ.0) GO TO 21
      DO 25 M=1,3
        IF(1.GT.IYP.AND.1.LE.NUNITS.AND.M.NE.2) GO TO 25
        IF(1.LE.IYP.AND.M.EQ.2) GO TO 25
        IF(LMT(1).LT.1.AND.IAB(I).EQ.0.AND.M.EQ.1) GO TO 25
        IF(LMT(2).EQ.0.AND.M.EQ.1) GO TO 25
        IF(LMT(3).EQ.0.AND.M.EQ.1) GO TO 25
        IF(IAB(I).EQ.1.AND.M.NE.1) GO TO 25
        AIJ(NEQ,LPU(1,M,L))=NAT(I)
        WRITE(13,03)NEQ,LPU(1,M,L),AIJ(NEQ,LPU(1,M,L)),
1          UNAME(I),MD(M),L
        IF (LANS.EQ.0) GO TO 25

```

```

                PRINT 03,NEQ,LPU(I,M,L),AIJ(NEQ,LPU(I,M,L)),
1                UNAME(I),MD(M),L
25        CONTINUE
21        CONTINUE
20        CONTINUE
C
C** FROM THE APCD FOR EACH UNIT WALKING TO THE
C** FRONT (J=1...IYP,M=2)
C
        DO 22 I=1,IYP
            IF(NAT(I).EQ.0.OR.NTV(I).EQ.0) GO TO 22
            MY=INT((DIST(2))/REAL(NTV(I)*LP))
            IF (MY.GT.IDP) GO TO 22
C
C** FOR EACH PERIOD UNTIL IDP-MY
C
            DO 23 L=1,IDP-MY
                AIJ(NEQ,LPU(I,2,L))=NAT(I)
                WRITE(13,03)NEQ,LPU(I,2,L),AIJ(NEQ,LPU(I,2,L)),
1                UNAME(I),MD(2),L
                IF (LANS.EQ.0) GO TO 23
                PRINT 03,NEQ,LPU(I,2,L),AIJ(NEQ,LPU(I,2,L)),
1                UNAME(I),MD(2),L
23        CONTINUE
22        CONTINUE
            RHS(NEQ)=OBJ
            WRITE(13,08)NEQ,RHS(NEQ)
            IF(LANS.EQ.1) PRINT 08,NEQ,RHS(NEQ)
C.....
            RETURN
C
C** TYPE 3
C** SET UP EQUATIONS FOR COMBAT POWER
C
29        PRINT *,'/-----/'
            PRINT *,'/      FIREPOWER/'
            PRINT *,'/-----/'
            WRITE(13,*)'/-----/'
            WRITE(13,*)'/      FIREPOWER/'
            WRITE(13,*)'/-----/'
C** INCREMENT THE ROW BY 1
            NEQ=NEQ+1
C.....
C** FROM THE FRONT
C** FOR EACH PERIOD UP TO THE DESIRED TIME
C
            DO 30 L=1,IDP
C** FOR EACH UNIT
                DO 31 I=1,NUNITS+1
                    IF(NFP(I).EQ.0.OR.CPI(L).EQ.0) GO TO 31
                    DO 35 M=1,3
                        IF(I.GT.IYP.AND.I.LE.NUNITS.AND.M.NE.2) GO TO 35
                        IF(I.LE.IYP.AND.M.EQ.2) GO TO 35
                        IF(LMT(1).LT.1.AND.IAB(I).EQ.0.AND.M.EQ.1) GO TO 35
                        IF(LMT(2).EQ.0.AND.M.EQ.1) GO TO 35

```

```

                IF(LMT(3).EQ.0.AND.M.EQ.1) GO TO 35
                IF(IAB(1).EQ.1.AND.M.NE.1) GO TO 35
                AIJ(NEQ,LPU(I,M,L))=NFP(I)*CPI(L)
                WRITE(13,03)NEQ,LPU(I,M,L),AIJ(NEQ,LPU(I,M,L)),
1                UNAME(I),MD(M),L
                IF (LANS.EQ.0) GO TO 35
                PRINT 03,NEQ,LPU(I,M,L),AIJ(NEQ,LPU(I,M,L)),
1                UNAME(I),MD(M),L
35                CONTINUE
31                CONTINUE
30                CONTINUE
C
C** FROM THE APOD FOR EACH UNIT WALKING TO THE FRONT (I=1...IYP,M=2)
C
        DO 32 I=1,IYP
            IF(NFP(I).EQ.0.OR.NTV(I).EQ.0) GO TO 32
            IF(IAB(I).EQ.1) GO TO 32
            MY=INT((DIST(2))/REAL(NTV(I)*LP))
            IF (MY.GT.IDP) GO TO 32
C
C** FOR EACH PERIOD UNTIL IDP-MY
C
            DO 33 L=1,IDP-MY
                AIJ(NEQ,LPU(I,2,L))=NFP(I)*CPI(L)
                WRITE(13,03)NEQ,LPU(I,2,L),AIJ(NEQ,LPU(I,2,L)),
1                UNAME(I),MD(2),L
                IF (LANS.EQ.0) GO TO 33
                PRINT 03,NEQ,LPU(I,2,L),AIJ(NEQ,LPU(I,2,L)),
1                UNAME(I),MD(2),L
33                CONTINUE
32                CONTINUE
34                RHS(NEQ)=OBJ
                WRITE(13,08)NEQ,RHS(NEQ)
                IF(LANS.EQ.1) PRINT 08,NEQ,RHS(NEQ)
C.....
                RETURN
C
C** TYPE 4
C** GOAL FOR A SPECIFIC UNIT (IUT) DEPLOYED BY PERIOD IDP
C
39        PRINT *,'/-----/'
            PRINT *,'/          UNIT GOAL/'
            PRINT *,'/-----/'
            WRITE(13,*)'/-----/'
            WRITE(13,*)'/          UNIT GOAL/'
            WRITE(13,*)'/-----/'
C** INCREMENT THE ROW BY 1
            NEQ=NEQ+1
C.....
C** FROM THE FRONT
C** FOR EACH PERIOD UP TO THE DESIRED TIME
C
            DO 40 L=1,IDP
                DO 41 M=1,3,2
                    IF(LMT(1).LT.1.AND.IAB(IUT).EQ.0.AND.M.EQ.1) GO TO 41

```

```

                IF(LMT(2).EQ.0.AND.M.EQ.1) GO TO 41
                IF(LMT(3).EQ.0.AND.M.EQ.1) GO TO 41
                IF(IAB(IUT).EQ.1.AND.M.NE.1) GO TO 41
                AIJ(NEQ,LPU(IUT,M,L))=1.0
                WRITE(13,03)NEQ,LPU(IUT,M,L),AIJ(NEQ,LPU(IUT,M,L)),
1                UNAME(IUT),MD(M),L
                IF (LANS.EQ.0) GO TO 41
                PRINT 03,NEQ,LPU(IUT,M,L),AIJ(NEQ,LPU(IUT,M,L)),
1                UNAME(IUT),MD(M),L
41          CONTINUE
40          CONTINUE
C
C** FROM THE APOD FOR IUT
C
                IF (NTV(IUT).EQ.0) GO TO 42
                MY=INT((DIST(2))/REAL(NTV(IUT)*LP))
                IF (MY.GT.IDP) GO TO 42
C
C** FOR EACH PERIOD UNTIL IDP-MY
C
                DO 43 L=1,IDP-MY
                AIJ(NEQ,LPU(IUT,2,L))=1.0
                WRITE(13,03)NEQ,LPU(IUT,2,L),AIJ(NEQ,LPU(IUT,2,L)),
1                UNAME(IUT),MD(2),L
                IF (LANS.EQ.0) GO TO 43
                PRINT 03,NEQ,LPU(IUT,2,L),AIJ(NEQ,LPU(IUT,2,L)),
1                UNAME(IUT),MD(2),L
43          CONTINUE
42          CONTINUE
44          RHS(NEQ)=OBJ
                WRITE(13,08)NEQ,RHS(NEQ)
                IF(LANS.EQ.1) PRINT 08,NEQ,RHS(NEQ)
                LANS=0
C.....
                RETURN
03          FORMAT('AIJ(',2I4,',')=',F14.5,3X,A14,2X,A5,I3)
08          FORMAT('RHS(',I4,',')=',F14.5,/, '-----')
                END
C*****
C** S3
                SUBROUTINE HEADER
C
C** PRINTS OUT THE HEADER TO THE PROGRAM AS AN INTRODUCTION
C
                CHARACTER BLANK
                PRINT 12
                PRINT *, ' WELCOME TO'
                PRINT *
                PRINT *
                PRINT *, ' DDDDD EEEEE PFFFF L 000
C Y Y'
                PRINT *, ' D D E P P L O O
C Y Y'
                PRINT *, ' D D E P P L O O
C Y Y'

```

```

      PRINT *,'    D        D     EEEEE   PFFFFF       L         O         O
C      Y'
      PRINT *,'    D        D     E         P           L         O         O
C      Y'
      PRINT *,'    D        D     E         P           L         O         O
C      Y'
      PRINT *,'    D        D     E         P           L         O         O
C      Y'
      PRINT *,'    DDDDD      EEEEEE   P           LLLLLL      000
C      Y'
      PRINT *
      PRINT *
      PRINT *,'                                A'
      PRINT *
      PRINT *,'                                GOAL PROGRAMMING MODEL'
      PRINT *
      PRINT *,'                                FOR THE'
      PRINT *
      PRINT *,'                                RAPID DEPLOYMENT OF ARMED FORCES'
      PRINT *
      PRINT *,'INPUT ANY CHARACTER TO CONTINUE.'
      READ *,BLANK
      PRINT 12
12  FORMAT(////////////////////)
      RETURN
      END
C*****
C S11
      SUBROUTINE FMAT
C
      COMMON/TAB/NP,LP,IP,NCR,NEG,NPRIT,LL(10,2),LU(10,2),LPX(10,4,5,4),
C      LPX(10,3,4),AIJ(150,300),RHS(150),LPS(3,5,4),LPSL(4),LPR(10),
C      NT(150),LMT(3),NC(10),NWGT(150),NDV(150),NID
C
C** SETS UP THE FILE FOR THE PASP PROGRAM
C
      PRINT *,'FMAT'
      OPEN(15,FILE='IFILE')
      OPEN(14,FILE='TFILE')
      REWIND(15)
      REWIND(14)
C
C WRITE THE IFILE AND FIND THE NUMBER OF TERMS (NT) FOR EACH ROW
C
      DO 10 I=1,NEQ
          DO 12 J=1,IP
              IF(AIJ(I,J).EQ.0) GO TO 12
              NT(I)=NT(I)+1
              WRITE(15,5)I,J,AIJ(I,J)
5             FORMAT(2I4,F15.9)
12          CONTINUE
              WRITE(15,5)I,0,RHS(I)
10          CONTINUE
      CLOSE(15)
C

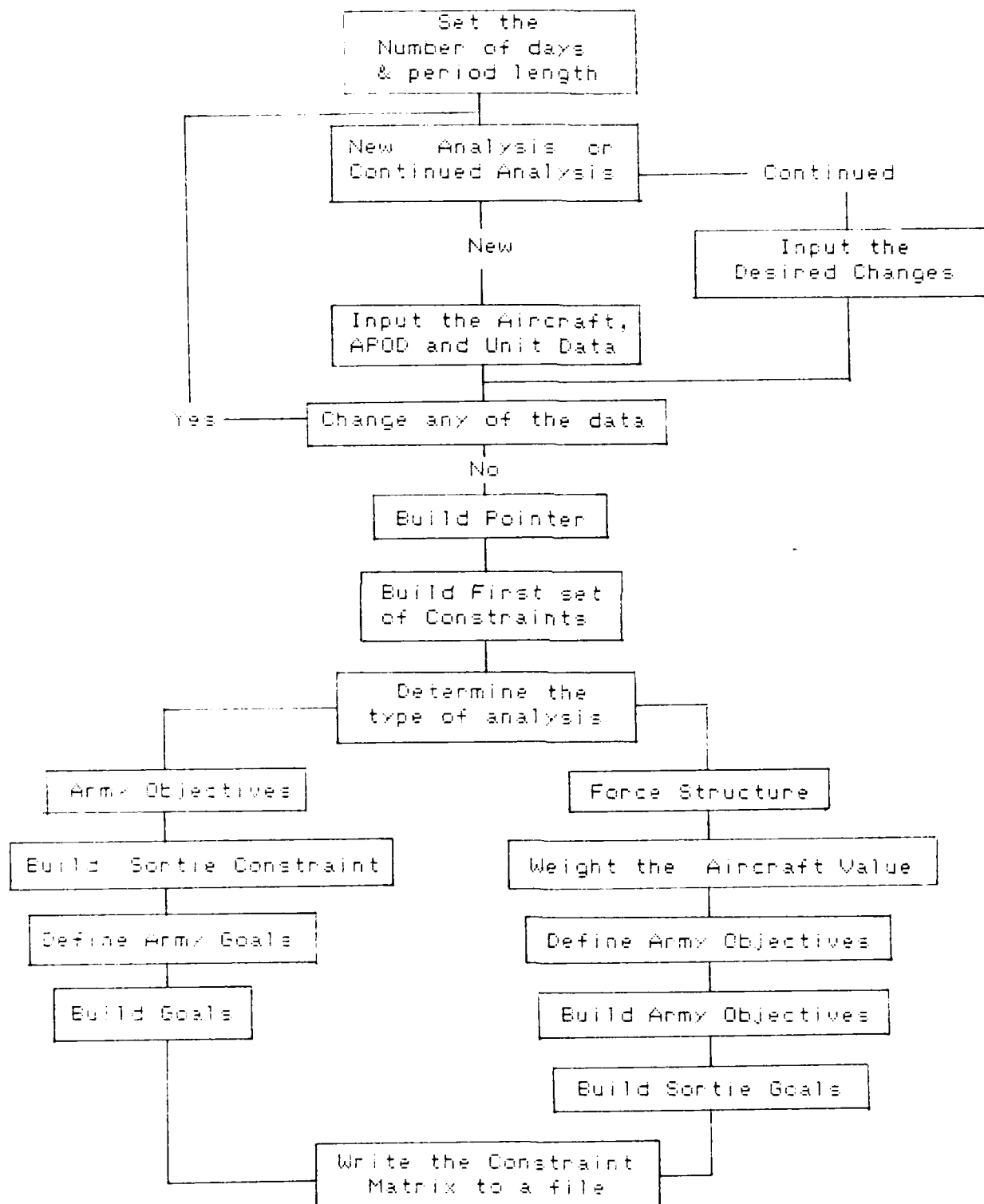
```

```

C++ WRITE THE INPUT PARAMETERS AND THE NUMBER OF TERMS PER ROW
C
      WRITE(14,*)NPRIT,IP,NCR,NEQ,NID
C
C++ THE NUMBER OF CONSTRAINTS FOR EACH PRIORITY
C
      DO 14 I=1,NPRIT
        WRITE(14,*)I,NC(I)
      14 CONTINUE
C
C++ THE TYPE OF DEVIATION VARIABLE TO MINIMIZE AND THE NUMBER OF
C++ TERMS IN THE EQUATION
C
      DO 16 I=1,NEQ
        WRITE(14,*)I,NDV(I),NWGT(I),NT(I)
      16 CONTINUE
      CLOSE(14)
C
      RETURN
      END
.
```

APPENDIX E

DEPLOY FLOWCHART



APPENDIX F

PAGE Subroutine and Function Listing

CINDEX -- Computes the relative cost coefficients for each variable in the current tableau and the objective function value at the current priority.

FIX -- (function) Places floating point values that are within $1.E-5$ of an integer to that integer.

PHSE1 -- Read in any real (equality) constraints and performs a simplex procedure to find an initial basic feasible solution.

PLACE -- Puts the objective function weights for the deviation variables at the current priority in the correct position in the tableau.

POUT -- Prepares and prints the solution information.

READ2 -- Reads in the goal constraints and objective function terms assigned to the current priority.

TEST -- Determines the next entering variable's column and row.

AD-A152 007

DEPLOY: AN INTERACTIVE GOAL PROGRAMMING MODEL FOR THE
RAPID DEPLOYMENT OF (U) AIR FORCE INST OF TECH
WRIGHT-PATTERSON AFB OH SCHOOL OF ENGI.. D O TATE

3/3

UNCLASSIFIED

06 DEC 84 AFIT/GOR/05/84D-14

F/G 15/7

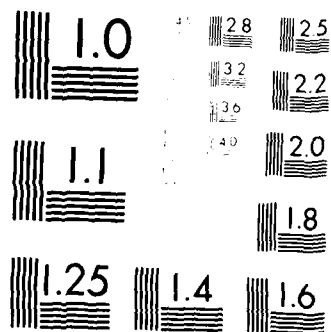
NL



END

FILED

DTIC



MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A

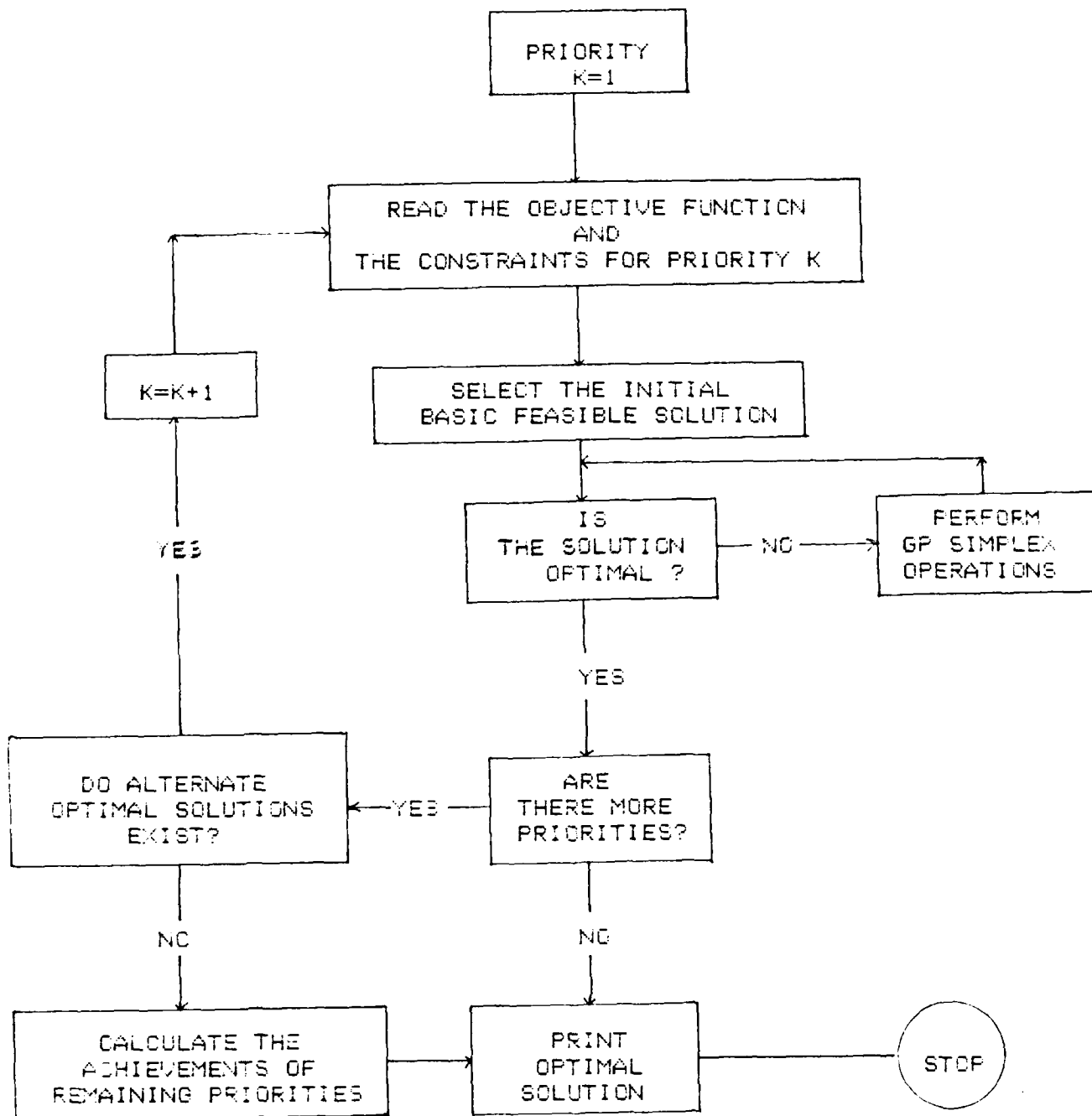
APPENDIX G

PAGE Variable Listing

Variable	Definition
IND(I)	-- Indicator row that marks the eligibility of a variable to enter the basis.
JCOL(I,1)	-- The type of variable in column I.
JCOL(I,2)	-- The subscript of the variable in column I.
JROW(I,1)	-- The type of basic variable in row I, where type is X=2; p=3; n=4.
JROW(I,2)	-- The subscript of the basic variable in row I.
NC(N)	-- The number of constraints assigned to priority N.
NCOLI	-- The number of columns in the current tableau.
NPRIC	-- The priority currently being optimized.
NPRIT	-- Total number of priorities in the problem.
NROWI	-- The number of rows in the current working tableau.
NVAR	-- The number of decision variables.
TA(N)	-- The total deviation from the goals at priority N.
TB(I)	-- The right hand side constant of the constraint in row I.
TE(I,J)	-- The coefficient of the variable in column J of the constraint in row I.
TL(N,J)	-- The weight assigned to the basic variable in row I at priority N.
TT(N,J)	-- The weight of the variable in column J at priority N.

APPENDIX H

Flowchart for PAGE Allocation Algorithm



APPENDIX I

PAGE Listing

```

PROGRAM PAGP
C-----+-----+-----+-----+-----+-----+-----+-----+
C-  CAPT D. O. TATE  THESIS WORK
C-----+-----+-----+-----+-----+-----+-----+
      CHARACTER KID(188)*24
      COMMON TT(10,402),TB(130),TE(130,402),TL(130,10),TA(10),INB(130),
C    TI(10,402),JCOL(402,2),NCOLI,NROWI,NPRIC,NC(10),JROW(130,2),
C    NVAR,NPRIT,IND(402),NTR(130),NDV(130),NID,NWGT(130),LCTR
      COMMON /PHASE1/ W,NRCON,NDVR
      COMMON /CHNG/NCON(130,10),NTOF(10)
      COMMON/CHAR/KID
      INTEGER ALTST

C
C **** READ IN PROBLEM DATA
C
      OPEN(13,FILE='IFILE')
      REWIND(13)
      OPEN(14,FILE='TFILE')
      REWIND(14)
      OPEN(15,FILE='RSFILE')
      REWIND(15)

C
C- INPUT THE #PRIORITIES, #VARIABLES, #REAL
C- CONSTRAINTS, AND NEQ
C
      READ(14,*)NPRIT,NVAR,NRCON,NEQ,NID
      OPEN(16,FILE='IDFILE')
      REWIND(16)
      DO 23 I=1,NID
          READ(16,6)K,KID(I)
23  CONTINUE
6    FORMAT(14,3X,A24)

C
C- INPUT THE NUMBER OF CONSTRAINTS THAT WILL
C- BE ADDED AT EACH PRIORITY
C
      DO 22 NP=1,NPRIT
          READ(14,*)I,NC(NP)
          IF(I.NE.NP) THEN
              PRINT *, 'I <> NP- I=',I,' NP=',NP
              STOP
          END IF
22  CONTINUE

C
C- READ IN THE DEVIATION VAR WEIGHTS AND THE NUMBER OF TERMS/EQUATION
C
      DO 20 NR=1,NEQ
          READ(14,*)I,NDV(I),NWGT(I),NTR(I)
          IF(I.NE.NR) THEN
              PRINT *, 'I <> NR- I= ',I,' NR=',NR

```

```

        STOP
      END IF
20  CONTINUE
    PRINT 16,NPRIT,NVAR,NRCON,NEQ
    WRITE(15,16)NPRIT,NVAR,NRCON,NEQ
16  FORMAT('NPRIT=',I4,' NVAR=',I4,' NRCON=',I4,' NEQ=',I4)
    DO 10 NP=1,NPRIT
10  CONTINUE
17  FORMAT('NC(',I4,')=',I4)
    I=0
C
C- INPUT THE SUBSCRIPTS OF THE CONSTRAINTS
C- TO BE ADDED AT A SPECIFIC PRIORITY
C
    DO 101 NP=1,NPRIT
      IF (NC(NP).EQ.0) GO TO 101
      NCTMP=NC(NP)
      DO 12 N=1,NCTMP
        I=I+1
        NCON(N,NP)=I
12  CONTINUE
101 CONTINUE
18  FORMAT('NCON(',I3,')=',I3)
C
C- INPUT THE NUMBER OF TERMS IN THE OBJECTIVE
C- FUNCTION AT EACH PRIORITY
C
    NTOF(1)=NC(1)
    DO 14 J=2,NPRIT
      NTOF(J)=NC(J)
14  CONTINUE
19  FORMAT('NTOF(',I3,')=',I3)
C
C **** INITIALIZE SUBPROBLEM DIMENSIONS AND COLUMN INDICATORS.
C
    LCTR=0
    NCOLI=0
    NROWI=0
    NPRIC=0
    DO 104 NCR=1,402
      IND(NCR)=1
      DO 102 NR=1,130
102  TE(NR,NCR)=0.
      DO 103 NP=1,5
        TI(NP,NCR)=0.
103  TT(NP,NCR)=0.
104 CONTINUE
      DO 105 NR=1,130
        INB(NR)=0
      DO 105 NP=1,10
105  TL(NR,NP)=0.
C
C **** CHECK FOR REAL CONSTRAINTS.
C
    IF (NRCON.EQ.0) GO TO 106

```

```

      CALL PHSE1
      IF (NDVR.LE.0) GO TO 116
      IF (W.GT.0.) GO TO 117
C
C **** THE PARTITIONING ALGORITHM BEGINS.
C
106 NPRIC=NPRIC+1
   IF (NPRIC.EQ.1.AND.NRCON.EQ.0) GO TO 107
   GO TO 108
107 PRINT *, 'MUST INCLUDE SUBROUTINE READ1'
   GO TO 109
108 CALL READ2
   NPIV=0
109 CALL CINDX
   CALL TEST (NEVC,NDVR)
C
C **** IF NEVC IS LESS THAN ZERO, THE SUBPROBLEM IS OPTIMIZED.
C
   IF (NEVC.LE.0) THEN
     PRINT *, 'SUBPROBLEM OPTIMIZED'
     WRITE(15,*) 'SUBPROBLEM OPTIMIZED'
     GO TO 110
   END IF
C
C **** IF NDVR IS LESS THAN ZERO, NO MINIMUM POSITIVE RATIO WAS FOUND.
C
   IF (NDVR.LE.0) THEN
     PRINT *, 'NO MINIMUM POSITIVE RATIO FOUND'
     WRITE(15,*) 'NO MINIMUM POSITIVE RATIO FOUND'
     GO TO 116
   END IF
   CALL PERM (NEVC,NDVR)
   NPIV=NPIV+1
   PRINT *, 'PIVOT# ',NPIV,' NPRIC=',NPRIC
   WRITE(15,*) 'PIVOT# ',NPIV,' NPRIC=',NPRIC
   GO TO 109
C
C **** IF THERE ARE NO MORE PRIORITIES, TOTAL PROBLEM IS OPTIMIZED.
C **** PRINT THE OPTIMAL SOLUTION.
C
110 IF (NPRIC.EQ.NPRIT) GO TO 115
C
C **** SINCE THERE ARE MORE PRIORITIES, MOVE ON TO THE NEXT SUBPROBLEM
C **** IF THERE ARE ALTERNATE SOLUTIONS. FIRST, ELIMINATE THOSE
C **** COLUMNS WHICH CAN NOT ENTER THE BASIS. IF THERE ARE NO
C **** ALTERNATE SOLUTIONS, PRINT THE UNIQUE OPTIMAL SOLUTION.
C
   ALTST=0
   DO 112 NCR=1,NCOL1
     IF (IND(NCR).EQ.0) GO TO 112
     IF (TI(NPRIC,NCR).GT.0.) GO TO 112
     DO 111 NP=1,NROW1
       IF (JROW(NR,1).EQ.JCOL(NCR,1).AND.JROW(NR,2).EQ.JCOL(NCR,2))
1         GO TO 112
111   CONTINUE

```

```

      ALTST=1
112 CONTINUE
C
C **** IF ALTST=1, THERE ARE ALTERNATE SOLUTIONS.
C
      IF (ALTST.EQ.1) THEN
        PRINT *, 'THERE ARE ALTERNATE SOLUTIONS'
        WRITE(15,*) 'THERE ARE ALTERNATE SOLUTIONS'
        GO TO 113
      END IF
      GO TO 115
C
C **** ELIMINATE THOSE COLUMNS WITH A POSITIVE RELATIVE COST AT
C **** PRIORITY NPRIC.
C
      113 DO 114 NCR=1,NCOLI
      114 IF (TI(NPRIC,NCR).GT.0.) IND(NCR)=0
        GO TO 106
C
C **** THE OPTIMIZATION IS OVER. PRINT OUT THE FINAL SOLUTION.
C
      115 CALL POUT
        GO TO 119
      116 WRITE (6,123) NPRIC
        GO TO 119
      117 WRITE (6,124) W
        WRITE (6,125)
        DO 118 NR=1,NROWI
          WRITE (6,126) JROW(NR,1),JROW(NR,2),TB(NR)
      118 CONTINUE
      119 WRITE(6,*) 'OPTIMIZATION IS FINISHED. ANSWER IS IN FILE GPANS.'
        STOP
C
      120 FORMAT (3I5)
      121 FORMAT (10I5)
      122 FORMAT (16I5)
      123 FORMAT (/ 40H THE PROGRAM TERMINATED ON SUBPROBLEM ,I4, 42H NO
        1 MINIMUM POSITIVE RATIO COULD BE FOUND)
      124 FORMAT (/ 65H THE PROGRAM TERMINATED IN PHASE 1 WITH OBJECTIVE F
        1UNCTION VALUE,F15.4)
      125 FORMAT (/ 55H THE OPTIMAL SOLUTION TO THE PHASE 1 PROBLEM IS
        1 // 6H TYPE,2X, 3H SUB,8X, 5H VALUE)
      126 FORMAT (2I5,F15.4)
C
      END
      SUBROUTINE PHSE1
C
C **** SUBROUTINE PHSE1 READS IN ANY REAL CONSTRAINTS AND PERFORMS A
C **** PHASE 1 SIMPLEX PROCEDURE IN ORDER TO FIND AN INITIAL BASIC
C **** FEASIBLE SOLUTION.
C
      COMMON TT(10,402),TB(130),TE(130,402),TL(130,10),TA(10),INB(130),
C      TI(10,402),JCOL(402,2),NCOLI,NROWI,NPRIC,NC(10),JROW(130,2),
C      NVAR,NPRIT,IND(402),NTR(130),NDV(130),NID,NWGT(130),LCTR
      COMMON /PHASE1/ W,NRCON,NDVR

```

```

        DIMENSION C(402), CR(402), CB(130)
        NPIV=0
C
C **** SET COLUMN AND ROW HEADINGS
C
        DO 101 NV=1,NVAR
            JCOL(NV,1)=2
101 JCOL(NV,2)=NV
        DO 102 NR=1,NRCON
            JROW(NR,1)=1
            JROW(NR,2)=NR
            NAR=NVAR+NR
            JCOL(NAR,1)=1
102 JCOL(NAR,2)=NR
C
C **** READ IN COEFFICIENTS AND RHS OF REAL CONSTRAINTS
C
        DO 103 NR=1,NRCON
C-
C- INPUT THE RHS OF REAL CONSTRAINT NUMBER ,NR
C- INPUT THE COEFFICIENTS OF REAL CONSTRAINT NUMBER
C-
            DO 30 NV=1,NTR(NR)
                READ(13,6)I,J,TE(I,J)
31          FORMAT('TE(I,J)=',I3,3X,I3,3X,F9.2)
                IF(I.NE.NR) THEN
                    PRINT *, '** I <> NR ** I=',I,' NR=',NR,' J=',J
                    STOP
                END IF
30          CONTINUE
                READ(13,6)I,J,TB(I)
6          FORMAT(2I4,F15.9)
32          FORMAT('TB(I)=',I3,3X,F9.2)
103 CONTINUE
            WRITE(15,*) 'PHASE1 DATA ENTERED'
C
C **** PUT IDENTITY MATRIX IN FOR ARTIFICIAL VARIABLES
C
        DO 104 NR=1,NRCON
            NAR=NVAR+NR
            INB(NR)=NAR
104 TE(NR,NAR)=1.
            WRITE(15,*)
            WRITE(15,*) 'THE STARTING BASIS.'
            PRINT *, 'THE STARTING BASIS.'
            DO 55 I=1,NRCON
                WRITE(15,56)I,INB(I)
55          PRINT 56,I,INB(I)
56          FORMAT('INB:',I3,3X,I4)
C
C **** SET C(J)=0 FOR ALL DECISION VARIABLES AND C(J)=1 FOR ALL
C **** ARTIFICIAL VARIABLES
C
        DO 105 NV=1,NVAR
105 C(NV)=0.

```

```

      DO 105 NR=1,NRCON
        CB(NR)=1.
        NAR=NVAR+NR
106 C(NAR)=1.
C
C*** CALCULATE RELATIVE COST COEFFICIENTS CR(.)
C
      NCOL=NVAR+NRCON
107 DO 108 NV=1,NCOL
        CR(NV)=C(NV)
        DO 108 NR=1,NRCON
108 CR(NV)=CR(NV)-CB(NR)*TE(NR,NV)
C
C **** CHECK FOR OPTIMALITY
C
      NEVC=0.
      NEVC=0
      DO 109 NCO=1,NCOL
        NV=NCO
        IF (CR(NV).GE.0.) GO TO 109
        IF (CR(NV).GE.NEVC) GO TO 109
        NEVC=CR(NV)
        NEVC=NV
109 CONTINUE
C
C **** IF NEVC=0, PHASE 1 IS OPTIMIZED. CALCULATE OBJECTIVE FUNCTION.
C
      IF (NEVC.EQ.0) GO TO 115
C
C **** DETERMINE DEPARTING VARIABLES ROW.
C
      NDUR=0
      UDUR=10.0E+20
      DO 111 NRI=1,NRCON
        NR=NRI
        IF (TE(NR,NEVC).LE.0.) GO TO 111
        U=TB(NR)/TE(NR,NEVC)
        IF (NDUR.EQ.0) GO TO 110
        IF (U-UDUR) 110,110,111
110 UDUR=U
        NDUR=NR
111 CONTINUE
C
C **** IF NDUR=0, MINIMUM RATIO RULE FAILED. RETURN.
C
      IF (NDUR.EQ.0) RETURN
C
C **** PERFORM THE PIVOT. REPLACE HEADINGS AND COST COEFFICIENT.
C
      JROW(NDUR,1)=JCOL(NEVC,1)
      JROW(NDUR,2)=JCOL(NEVC,2)
      CB(NDUR)=C(NEVC)
      NDVC=INB(NDUR)
      INB(NDUR)=NEVC
      PRINT 5,NDUR,NDVC,UDUR,NEVC,NEVC

```

```

        WRITE(15,5)NDVR,NDVC,VDVR,NEVC,VEVC
5      FORMAT(5X,'NDVR=',I4,3X,'NDVC=',I4,3X,'VDVR=',F10.3,
1      /17X,'NEVC=',I4,3X,'VEVC=',F10.3)
C
C **** COMPUTE NEW TE ARRAY
C
      PIV=TE(NDVR,NEVC)
      PIB=TB(NDVR)
      DO 113 NR=1,NRCON
        IF (NR.EQ.NDVR) GO TO 113
        IF (ABS(TE(NR,NEVC)).LE.0.0005) GO TO 113
        PIX=TE(NR,NEVC)/PIV
        TB(NR)=FIX(TB(NR)-PIX*PIB)
        DO 112 NV=1,NCOL
112      TE(NR,NV)=FIX(TE(NR,NV)-TE(NDVR,NV)*PIX)
113 CONTINUE
        TB(NDVR)=FIX(PIB/PIV)
        DO 114 NV=1,NCOL
114      TE(NDVR,NV)=FIX(TE(NDVR,NV)/PIV)
C
C **** END OF PIVOT OPERATIONS. PROCEED TO NEXT ITERATION.
C
      NPIV=NPIV+1
      PRINT *, 'PIVOT # ',NPIV, ' NPRIC=',NPRIC
      WRITE(15,*) 'PIVOT # ',NPIV, ' NPRIC=',NPRIC
      GO TO 107
C
C **** CALCULATE W, THE PHASE-1 OBJECTIVE FUNCTION.
C
115 W=0.
      DO 116 NR=1,NRCON
116 W=W+TB(NR)*CB(NR)
C
C **** INITIALIZE THOSE PORTIONS OF THE TABLEAU ASSIGNED TO THE
C **** ARTIFICIAL VARIABLES.
C
      DO 117 NR=1,NRCON
        DO 117 NV=1,NCOL
          IF (NV.LE.NVAR) GO TO 117
          TE(NR,NV)=0.
117 CONTINUE
C
C **** UPDATE NCOLI AND NROWI PARAMETERS.
C
      NROWI=NRCON
      NCOLI=NVAR
      PRINT *, 'TOTAL PIVOTS=',NPIV
      WRITE(15,*) 'TOTAL PIVOT=',NPIV
      RETURN
C
C
      END
      SUBROUTINE READ2
C
C **** SUBROUTINE READ2 READS IN THE GOAL CONSTRAINTS AND OBJECTIVE

```

C **** FUNCTION TERMS ASSIGNED TO PRIORITY NPRIC.
 C **** SUBROUTINE READ2 IS ALSO USED TO READ IN THE FIRST PRIORITY GOAL
 C **** CONSTRAINTS AND OBJECTIVE FUNCTION TERMS IF REAL CONSTRAINTS ARE
 C **** PRESENT.

```

COMMON TT(10,402),TB(130),TE(130,402),TL(130,10),TA(10),INB(130),
C   TI(10,402),JCOL(402,2),NCOLI,NROWI,NPRIC,NC(10),JROW(130,2),
C   NVAR,NPRIT,IND(402),NTR(130),NDV(130),NID,NWGT(130),LCTR
COMMON /CHNG/NCON(130,10),NTOF(10)
    IF(NC(NPRIC).EQ.0) GO TO 107
  
```

C
 C **** READ IN THE COEFFICIENTS OF THE X'S.

```

NCTMP=NC(NPRIC)
DO 106 NRI=1,NCTMP
  NR=NRI+NROWI
  NC1=NCOLI+2*NRI-1
  NC2=NCOLI+2*NRI
  JCOL(NC1,1)=4
  JCOL(NC1,2)=NCON(NRI,NPRIC)
  JCOL(NC2,1)=3
  JCOL(NC2,2)=NCON(NRI,NPRIC)
  
```

C
 C- INPUT THE COEFFICIENTS OF THE DECISION VARIABLES
 C- AND RHS FOR PRIORITY #NPRIC AND CONSTRAINT #NRI

```

DO 30 NV=1,NTR(NR)
  READ(13,6)I,J,TE(I,J)
31  FORMAT('TE(I,J)=',I3,3X,I3,3X,F9.2)
  IF(I.NE.NR) THEN
    PRINT *,'** I <> NR ** I=',I,' NR=',NR,' J=',J
    STOP
  END IF
30  CONTINUE
  READ(13,6)I,J,TE(I)
6  FORMAT(2I4,F15.9)
32  FORMAT('TB(I)=',I3,3X,F9.2)
  TE(NR,NC1)=1.
  INB(NR)=NC1
  TE(NR,NC2)=-1.
  
```

C
 C **** PERFORM THE ROW REDUCTION.

```

DO 102 NRC=1,NROWI
  IF (JROW(NRC,1).NE.2) GO TO 102
  J=JROW(NRC,2)
  TB(NR)=TB(NR)-TE(NR,J)*TB(NRC)
  DO 101 NCR=1,NC2
    IF (NCR.EQ.J) GO TO 101
    TE(NR,NCR)=TE(NR,NCR)-TE(NR,J)*TE(NRC,NCR)
101  CONTINUE
    TE(NR,J)=0.
102  CONTINUE
  
```

C
 C **** DETERMINE THE DEVIATIONAL VARIABLE TO ENTER THE BASIS.

```

C
C      IF (TB(NR)) 103,105,105
C
C **** SINCE TB IS LESS THAN ZERO, MULTIPLY THE ROW BY -1 AND ENTER D+
C **** IN THE BASIS.
C
103      DO 104 NCR=1,NC2
104      TE(NR,NCR)=-TE(NR,NCR)
          TB(NR)=-TB(NR)
          JROW(NR,1)=3
          JROW(NR,2)=NCON(NR1,NPRIC)
          GO TO 106
C
C **** SINCE TB IS GREATER THAN OR EQUAL TO ZERO ENTER D- IN THE BASIS.
C
105      JROW(NR,1)=4
          JROW(NR,2)=NCON(NR1,NPRIC)
106 CONTINUE
C
C **** INCREASE THE PARAMETERS NCOLI AND NROWI.
C
          NX=NROWI
          NCOLI=NC2
          NROWI=NR
C
C- AT PRIORITY #NPRIC, INPUT THE SUBSCRIPT, TYPE, AND
C- WEIGHT OF EACH TERM IN THE OBJECTIVE FUNCTION.
C   (IF TERM IS POSITIVE DEVIATIONAL, TYPE=3)
C   (IF TERM IS NEGATIVE DEVIATIONAL, TYPE=4)
C
107 NTTMP=NTOF(NPRIC)
      DO 108 NT=1,NTTMP
          LCTR=LCTR+1
          ISUB=LCTR
          IYPE=NDV(NX+NT)
          WGT=NWGT(NX+NT)
          CALL PLACE (ISUB,IYPE,WGT)
108 CONTINUE
      RETURN
C
C
      END
      SUBROUTINE PLACE (ISUB,IYPE,WGT)
C
C **** SUBROUTINE PLACE PUTS THE OBJECTIVE FUNCTION WEIGHTS FOR THE
C **** DEVIATION VARIABLES AT THE CURRENT PRIORITY LEVEL (NPRIC) IN THE
C **** CORRECT POSITIONS IN THE AUGMENTED TABLEAU.
C ****
C **** ISUB=THE SUBSCRIPT OF THE DEVIATIONAL VARIABLE
C ****
C **** IYPE=3, IF POSITIVE DEVIATIONAL VARIABLE (D+)
C ****      4, IF NEGATIVE DEVIATIONAL VARIABLE (D-)
C ****
C **** WGT=THE CARDINAL WEIGHT OF THIS DEVIATIONAL VARIABLE AT THE
C ****      CURRENT PRIORITY LEVEL

```

```

C      COMMON TT(10,402),TB(130),TE(130,402),TL(130,10),TA(10),INB(130),
C      TT(10,402),JCOL(402,2),NCOLI,NROWI,NPRIC,NCK(10),JROW(130,2),
C      NPAR,NPRIT,IND(402),NTR(130),NDV(130),NID,NWGT(130),LCTR
C      COMMON /CHNG/NCONC(130,10),NTOF(10)
C
C      **** PLACE THE WEIGHT IN THE PROPER COLUMN IN THE TOP STUB.
C
C      NC1=NVAR+1
C      DO 101 NCR=NC1,NCOLI
C          IF (JCOL(NCR,1).EQ.ITYPE.AND.JCOL(NCR,2).EQ.ISUB) GO TO 102
101 CONTINUE
102 TT(NPRIC,NCR)=WGHT
C
C      **** PLACE THE WEIGHT IN THE PROPER ROW IN THE LEFT STUB.
C
C      DO 103 NR=1,NROWI
C          IF (JROW(NR,1).EQ.ITYPE.AND.JROW(NR,2).EQ.ISUB) GO TO 104
103 CONTINUE
C      GO TO 105
104 TL(NR,NPRIC)=WGHT
105 CONTINUE
C      RETURN
C
C      END
C      SUBROUTINE CINDX
C
C      **** SUBROUTINE CINDX COMPUTES THE RELATIVE COST COEFFICIENTS FOR EACH
C      **** VARIABLE IN THE CURRENT TABLEAU (THE TI( . , . ) ARRAY) AND THE
C      **** OBJECTIVE FUNCTION VALUE (THE TA( . ) ARRAY) AT THE CURRENT
C      **** PRIORITY (NPRIC)
C
C      COMMON TT(10,402),TB(130),TE(130,402),TL(130,10),TA(10),INB(130),
C      TT(10,402),JCOL(402,2),NCOLI,NROWI,NPRIC,NC(10),JROW(130,2),
C      NPAR,NPRIT,IND(402),NTR(130),NDV(130),NID,NWGT(130),LCTR
C
C      **** COMPUTE TH(NPRIC) AND TI(NPRIC,NC) NC=1,.....,NCOLI
C
C      TH(NPRIC)=0.
C      DO 101 NR=1,NROWI
101 TH(NPRIC)=TH(NPRIC)+TB(NR)*TL(NR,NPRIC)
C      DO 102 NCR=1,NCOLI
C          TI(NPRIC,NCR)=TT(NPRIC,NCR)
C      DO 102 NR=1,NROWI
102 TI(NPRIC,NCR)=TI(NPRIC,NCR)+TB(NR,NCR)*TL(NR,NPRIC)
C      RETURN
C
C      END
C      SUBROUTINE TEST (NEVC,NDUR)
C
C      **** SUBROUTINE TEST DETERMINES THE NEXT ENTERING VARIABLE'S COLUMN
C      **** (NEVC) AND THE NEXT DEPARTING VARIABLE'S ROW (NDUR). IF NO
C      **** FURTHER OPTIMIZATION IS POSSIBLE, THE VALUE NEVC=0 IS RETURNED.
C      **** IF NDUR=0 IS RETURNED, NO MINIMUM POSITIVE RATIO COULD BE FOUND
C      **** IN THE CURRENT PIVOT OPERATION, I.E., ALL OF THE COEFFICIENTS

```

```

C **** TE(130,402),NEVC ARE NONPOSITIVE.
C
      CHARACTER KID(188)*24
      COMMON TT(10,402),TB(130),TE(130,402),TL(130,10),TA(10),INB(130),
C      TI(10,402),JCOL(402,2),NCOLI,NROWI,NPRIC,NC(10),JROW(130,2),
C      NVAR,NPRIT,IND(402),NTR(130),NDV(130),NID,NWGT(130),LCTR
      COMMON/CHAR/KID
      NDVR=0
      NEVC=0
      VEVC=0.
      VDVR=10.0E+20
C
C **** DETERMINE ENTERING VARIABLE'S COLUMN.
C
      DO 101 NCR=1,NCOLI
        IF (TI(NPRIC,NCR).GE.0.) GO TO 101
        IF (IND(NCR).EQ.0) GO TO 101
        IF (TI(NPRIC,NCR).GE.VEVC) GO TO 101
        NEVC=NCR
        VEVC=TI(NPRIC,NCR)
      101 CONTINUE
C
C **** IF NEVC=0, SUBPROBLEM NPRIC IS OPTIMIZED. RETURN.
C
      IF (NEVC.EQ.0) GO TO 107
C
C **** DETERMINE DEPARTING VARIABLE'S ROW.
C
      DO 105 NR=1,NROWI
        IF (TE(NR,NEVC).LE.0.) GO TO 105
        V=TB(NR)/TE(NR,NEVC)
        IF (NDVR.EQ.0) GO TO 104
        IF (V-VDVR) 104,102,105
      102 DO 103 NP=1,NPRIC
        IF (TL(NR,NP)-TL(NDVR,NP)) 105,103,104
      103 CONTINUE
      104 VDVR=V
      NDVR=NR
      105 CONTINUE
      107 IF (NDVR.EQ.0) RETURN
      NDVC=INB(NDVR)
      INB(NDVR)=NEVC
      PRINT 5,NDVR,NDVC,VDVR,NEVC,VEVC
      WRITE(15,5)NDVR,NDVC,VDVR,NEVC,VEVC
      RETURN
5     FORMAT(5X,'NDVR=',I4,3X,'NDVC=',I4,3X,'VDVR=',F10.3,
1     /17X,'NEVC=',I4,3X,'VEVC=',F10.3)
C
      END
      SUBROUTINE PERM (NEVC,NDVR)
C
C **** SUBROUTINE PERM PERFORMS THE PIVOT OPERATION USING THE PIVOT
C **** ELEMENT IN COLUMN NEVC AND ROW NDVR AND COMPUTES THE NEW TABLEAU.
C
      COMMON TT(10,402),TB(130),TE(130,402),TL(130,10),TA(10),INB(130),

```

```

C   TI(10,402),JCOL(402,2),NCOLI,NROWI,NPRIC,NC(10),JROW(130,2),
C   NVAR,NPRIT,IND(402),NTR(130),NDV(130),NID,NWGT(130),LCTR
C
C **** REPLACE HEADING FOR ROW NDVR.
C
C   JROW(NDVR,1)=JCOL(NEVC,1)
C   JROW(NDVR,2)=JCOL(NEVC,2)
C
C **** REPLACE TL VECTOR FOR ROW NDVR
C
C   DO 101 NP=1,NPRIC
101  TL(NDVR,NP)=TT(NP,NEVC)
C
C **** COMPUTE NEW TE ARRAY.
C
C   PIV=TE(NDVR,NEVC)
C   PIB=TB(NDVR)
C   DO 103 NR=1,NROWI
C     IF (NR.EQ.NDVR) GO TO 103
C     IF (ABS(TE(NR,NEVC)).LE.0.0005) GO TO 103
C     PIX=TE(NR,NEVC)/PIV
C     TB(NR)=FIX(TB(NR)-PIX*PIB)
C     DO 102 NCR=1,NCOLI
102    TE(NR,NCR)=FIX(TE(NR,NCR)-TE(NDVR,NCR)*PIX)
103  CONTINUE
C     TB(NDVR)=FIX(PIB/PIV)
C     DO 104 NCR=1,NCOLI
104    TE(NDVR,NCR)=FIX(TE(NDVR,NCR)/PIV)
C   RETURN
C
C   END
C   FUNCTION FIX(Z)
C
C **** FUNCTION FIX BRINGS FLOATING POINT VALUES THAT ARE WITHIN 1.E-3
C **** OF AN INTEGER TO THAT INTEGER.
C
C   FIX=AIN(TZ+SIGN(.5,Z))
C   IF (ABS(FIX-Z).GT. 1.E-3) FIX=Z
C
C   RETURN
C
C   END
C   SUBROUTINE POUT
C
C **** SUBROUTINE POUT PREPARES AND PRINTS THE SOLUTION INFORMATION.
C
C   CHARACTER KID(188)*24
C   COMMON TT(10,402),TB(130),TE(130,402),TL(130,10),TA(10),INB(130),
C   TI(10,402),JCOL(402,2),NCOLI,NROWI,NPRIC,NC(10),JROW(130,2),
C   NVAR,NPRIT,IND(402),NTR(130),NDV(130),NID,NWGT(130),LCTR
C   COMMON /CHNG/NCON(130,10),NTOF(10)
C   COMMON/CHAR/KID
C   DIMENSION WOUT(402,4), RLHS(130,10)
C   DIMENSION DIFF(130)
C   OPEN (11,FILE='GPANS')

```

```

        REWIND 11
        WRITE (11,122)
        WRITE (11,123) NPRIC,NROWI
C
C **** OUTPUT ARRAY IS ZEROED.
C
        DO 101 I=1,402
        DO 101 J=1,4
101 WOUT(I,J)=0.
C
C **** OUTPUT ARRAY IS FILLED.
C
        DO 102 NP=1,NPRIC
102 WOUT(NP,1)=FIX(TA(NP))
        DO 103 NR=1,NROWI
            I1=JROW(NR,1)
            I2=JROW(NR,2)
103 WOUT(I2,I1)=FIX(TB(NR))
C
C **** IF ALL PRIORITIES HAVE BEEN INCLUDED, PRINT OPTIMAL SOLUTION.
C **** IF NOT, WE MUST CALCULATE VALUES FOR REMAINING TA'S AND D- AND D+
C
        IF (NPRIC.GE.NPRIT) GO TO 114
        NP1=NPRIC+1
        DO 113 NP=NP1,NPRIT
            TA(NP)=0.
            IF (NC(NP).EQ.0) GO TO 106
C
C **** READ IN THE GOAL CONSTRAINTS ASSIGNED TO PRIORITY NP.
C
        NCTMP=NC(NP)
        DO 105 NCI=1,NCTMP
            NR=NROWI+NCI
C
C- INPUT THE COEFFICIENTS OF THE DECISION VARIABLES
C- AND RHS FOR PRIORITY #NPRIC AND CONSTRAINT #NRI
C
        DO 30 NV=1,NTR(NR)
            READ(13,6) I,J,TE(I,J)
            IF (I.NE.NR) THEN
                PRINT *, '** I (> NR ** I=',I,' NR=',NR,' J=',J
                STOP
            END IF
30      CONTINUE
            READ(13,6) I,J,TB(I)
6        FORMAT(2I4,F15.9)
            RLHS(NCI,NP)=0.
            DO 104 NV=1,NVAR
104      RLHS(NCI,NP)=RLHS(NCI,NP)+TE(NR,NV)*WOUT(NV,2)
            DIFF(NCI)=TB(NR)-RLHS(NCI,NP)
105      CONTINUE
C
C
C- AT PRIORITY #NPRIC, INPUT THE SUBSCRIPT, TYPE, AND
C- WEIGHT OF EACH TERM IN THE OBJECTIVE FUNCTION.

```

```

C      (IF TERM IS POSITIVE DEVIATIONAL, TYPE=3)
C      (IF TERM IS NEGATIVE DEVIATIONAL, TYPE=4)
C
106      NTTMP=NTOF(NP)
        DO 112 NT=1,NTTMP
            LCTR=LCTR+1
            ISUB=LCTR
            ITYPE=NDV(NT+NROWI)
            WGHT=NWGT(NT+NROWI)
            IF (NC(NP).EQ.0) GO TO 111
            NCTMP=NC(NP)
            DO 110 NCI=1,NCTMP
                IF (ISUB.NE.NCON(NCI,NP)) GO TO 110
                IF (DIFF(NCI)) 107,108,109
107            IF (ITYPE.NE.3) GO TO 110
                WOUT(ISUB,3)=-DIFF(NCI)
108            GO TO 110
109            IF (ITYPE.NE.4) GO TO 110
                WOUT(ISUB,4)=DIFF(NCI)
110            CONTINUE
111            TA(NP)=TA(NP)+WGHT*WOUT(ISUB,ITYPE)
112            CONTINUE
            NROWI=NROWI+NC(NP)
C
C      **** FILL IN THE OUTPUT VALUE FOR ATTAINMENT OF PRIORITY NP.
C
            WOUT(NP,1)=FIX(TA(NP))
113 CONTINUE
C
C      **** PRINT OPTIMAL SOLUTION
C
114 WRITE (11,126)
        WRITE (11,127)
        DO 115 NV=1,NID
            IF(WOUT(NV,2).EQ.0) GO TO 115
            IF(NV.LE.NID) WRITE (11,7) NV,KID(NV),WOUT(NV,2)
            IF(NV.GT.NID) WRITE(11,128) NV,WOUT(NV,2)
115 CONTINUE
7      FORMAT(13,3X,A24,3X,F15.4)
        WRITE (11,126)
        WRITE (11,129)
        DO 116 NP=2,NPRIT
            IF (NC(NP).EQ.0) GO TO 116
            NCTMP=NC(NP)
            DO 139 NCO=1,NCTMP
                N=NCON(NCO,NP)
                WRITE (11,130) NP,N,WOUT(N,3),WOUT(N,4)
139 CONTINUE
116 CONTINUE
        J=1
        IF(J.EQ.1) RETURN
        WRITE (11,126)
        WRITE (11,131)
        DO 117 NP=1,NPRIT
            WRITE (11,132) NP,WOUT(NP,1)

```

```

117 CONTINUE
    WRITE (11,135)
    WRITE (11,136)
    WRITE (11,137)
    I=MAX0(NPRIT,NVAR,NROW)
    DO 121 K=1,I
        IF (K.GT.NPRIT) GO TO 119
        IF (K.GT.NVAR) GO TO 118
        WRITE (11,135) K,(WOUT(K,J),J=1,4)
        GO TO 121
118    WRITE (11,136) K,WOUT(K,1),(WOUT(K,J),J=3,4)
        GO TO 121
119    IF (K.GT.NVAR) GO TO 120
        WRITE (11,137) K,(WOUT(K,J),J=2,4)
        GO TO 121
120    WRITE (11,138) K,(WOUT(K,J),J=3,4)
121 CONTINUE
    WRITE (11,126)
    RETURN

```

C

```

122 FORMAT (1H1)
123 FORMAT (/ 39H THE OPTIMIZATION ENDED ON SUBPROBLEM ,I5 / 13H T
    1HERE WERE ,I5, 42H CONSTRAINTS IN THE FINAL OPTIMAL TABLEAU.)
126 FORMAT (//80(1H*))
127 FORMAT (1H0, 52HTHE OPTIMAL SOLUTION FOR THE DECISION VARIABLES X(
    1J))
128 FORMAT (1H0, 2HX(,I3, 2H)=,F15.4)
129 FORMAT (1H0, 25HTHE GOAL ACHIEVEMENTS ARE // 9H PRIORITY,2X, 11H
    1GOAL NUMBER,2X, 16HOVER-ACHIEVEMENT,2X, 17HUNDER-ACHIEVEMENT)
130 FORMAT (4X,I2,10X,I2,10X,F10.4,10X,F10.4)
131 FORMAT (1H0, 29HTHE PRIORITY ACHIEVEMENTS ARE // 9H PRIORITY,8X,
    1 11HACHIEVEMENT)
132 FORMAT (4X,I2,10X,F10.4)
133 FORMAT (1H0, 15H OUTPUT SUMMARY)
134 FORMAT (1H0, 9HSUBSCRIPT,11X, 8H A OPT,7X, 8H X OPT,7X, 9H
    1 POS DEV,6X, 9H NEG DEV /)
135 FORMAT (18,7X,4F15.4)
136 FORMAT (18,7X,F15.4,15X,2F15.4)
137 FORMAT (18,22X,3F15.4)
138 FORMAT (18,37X,2F15.4)

```

C

END

David Oliver Tate was born on 13 August 1953 in Holyoke, Massachusetts. Being the son of an Air Force Officer, he traveled much of his life. After graduating from High School in 1971, he attended Willard School in Bandon, Oregon. Then, in 1972, he entered the U.S. Air Force Academy and graduated in 1976 with a B.S. in Management. After completing Undergraduate Pilot Training in 1977, he was assigned to the 4th MAW as a C-141 pilot at McChord AFB, Washington. While at McChord, he served as a Squadron Safety Officer, Assistant Executive Officer to the Wing Commander, and Wing Staff Flight Examiner. He graduated from EOE as a Distinguished Graduate in 1983.

Permanent Address: 311 N. Lowe Street
Hobart, OK 73651

END

FILMED

5-85

DTIC