AD-A180 229

AFIT/GST/ENS/87M-7

DTIC FILE COPY

AIR REFUELING TANKER SCHEDULING

THESIS

Harry C. Hostler
Captain, USAF

AFIT/GST/ENS/87M-7

Approved for public release; distribution unlimited

DTIC ELECTE MAY 1 8 1987 A

87 5 15 098

AFIT/GST/ENS/87M-7

AIR REFUELING TANKER SCHEDULING

THESIS

Presented to the Faculty of the School of Engineering

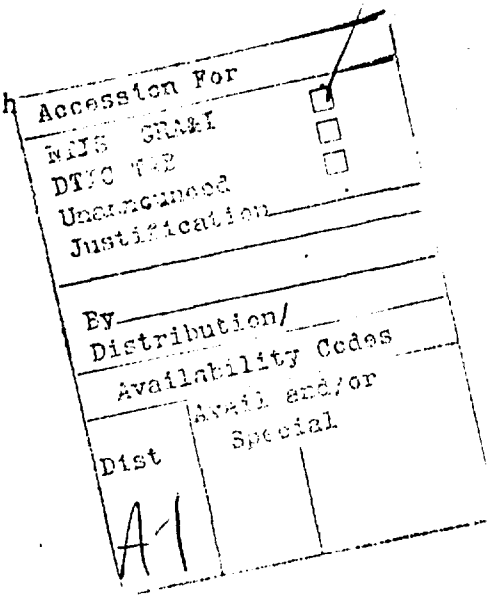
of the Air Force Institute of Technology

Air University

In Partial Fulfillment of the

Requirements for the Degree of

Masters of Science in Operations Research



Harry C. Hostler, B.A.

Captain, USAF

March 1987

Approved for public release; distribution unlimited

Preface

The purpose of this thesis was to develop a method for scheduling Strategic Air Command's fleet of air refueling tanker aircraft to perform more than one aerial refueling mission per flight. The need for a new scheduling program was driven by the ever increasing demand for aerial refueling support. The number of tankers could not keep pace with the demand.

A method was developed for scheduling tankers to perform more than one refueling on a single mission. It entails the use of original programs and a mixed integer package. Although the original programs were never automated, the algorithms for developing the programs are provided.

In performing the research and in the writing of this thesis I have been blessed with a great deal of help from others. I am deeply indebted to my thesis advisors, Lt Col T. F. Schuppe and Lt Col W. F. Rowell; without their continuing patience, sound advice, mercy, and many hours of sacrifice I would never have completed this thesis. I wish to thank Lt Col R. McElhinney and Maj E. Martin at SAC Head Quarters for their assistance and cooperation in providing the necessary information to develop this thesis. And to my wife, thank you, without her understanding and love I would have never had the strength to endure.

Harry C. Hostler

Table of Contents

List of Figures

Abstract

This thesis determined a way to schedule SAC's air refueling tanker fleet to perform, if necessary, more than one refueling mission during a flight. A preemptive goal programming approach was adopted using three priority levels. The basic formulation was that of the generalized assignment problem. Three objectives had to be considered when performing the task, maximize the number tanker requests satisfied, maximize the number of category B requests satisfied, and minimize the total flight time to perform all of the missions.

A preprocessor was developed to transform the inputs from the tanker and receiver scheduling units into a usable format to be executed by the mixed integer programming package. This preprocessor determined all of the possible refuelings that could take place, computed the flight times of the missions, and determined all of the variables to be used in the constraints and objective functions.

AIR REFUELING TANKER SCHEDULING

I. Introduction

Purpose

The purpose of this thesis is to develop a method of scheduling Strategic Air Command's (SAC) KC-135 and KC-10 air refueling tanker fleet to accomplish more airborne refuelings without increasing the number of sorties these planes must fly.

The current scheduling system used by Strategic Air Command (SAC) assigns one tanker to refuel one receiver. Units that desire an in-flight refueling submit a request for a tanker. Some requests can not be satisfied because the number of requests is greater than the number of available tankers. To decrease the number of requests that are not satisfied, SAC would like a computer scheduling program that has the flexibility to schedule, as necessary, a tanker to perform one mission that refuels two receivers.

Background

The KC-135 Stratotanker and KC-10 Extender aircraft provide in-flight refueling for a host of Air Force aircraft as well as Navy, Marine, and some foreign aircraft. Strategic Air Command (SAC) is the single Department of Defense agency tasked with the responsibility of managing and scheduling the aerial tanker fleet.

Several advantages are to be gained by airborne refueling. An obvious one is extended range. Another advantage is promptness. For example, with in-flight refueling, fighter aircraft can fly non-stop

from the east coast of the United States to Saudi Arabia in 15 hours rather than the 47 hours required by landing enroute to refuel. By refueling airborne, combat planes can avoid reliance on other countries for land-based refueling stations. Another benefit is security; combat aircraft can be based in safe areas.

The KC-135 first entered the Air Force inventory in 1956. Its mission was to extend the range of SAC's B-52 strategic bomber force (1:1). Today, nearly every major aircraft purchased by the Air Force can be refueled while airborne. There are nearly 10,000 aircraft in the Air Force inventory capable of being refueled in flight. This number is five times larger than the number of aircraft 20 years ago that were refuelable (1:1).

While the number of aircraft requiring inflight refueling (receivers) has grown considerably, the fleet of tankers has actually shrunk. The present fleet of KC-135 tankers consist of 642 aircraft, nearly 200 fewer than 20 years ago. The Air Force Reserve and Air National Guard manage 128 of these aircraft. Along with the KC-135, there are 51 KC-10 aircraft in the inventory with another nine remaining to be delivered (16).

The availability of tankers has not kept pace with the growing demand for refuelings. As the operational demand for tankers has increased, so have the training sorties to prepare for these missions.

Complicating matters is the tanker aircraft versatility. Approximately thirty percent of the KC-135's and crews are committed to "stand alert" in support of SAC's strategic bombers. The nearly 200 tankers on daily alert are not available to perform other missions. Another five

percent of the tanker fleet is not available because of repairs, scheduled maintenance, and major overhauls. Also, the location of many tanker bases limits their availability to perform refuelings. Most of the tanker aircraft are co-located with strategic bomber units in the northern portions of the CONUS (Continental United States) while a majority of the non-strategic aircraft perform their training and refuelings in southern CONUS locations. Refueling requests in these southern locations are often declined because the distance a tanker must fly to support the refueling is too great (11).

Rather than increasing the number of tankers to offset the tanker shortages, the existing KC-135A's are being reengined to increase the amount of fuel available for delivery to receiver aircraft. The 128 tankers flown by the Air National Guard and Air Force Reserve have been reengined with more fuel efficient engines making these KC-135Es equivalent to 1.2 KC-135As in fuel delivery. The active Air Force KC-135As are being fitted with new engines that make each of these reengined aircraft, KC-135R, equivalent to 1.5 KC-135As. By the mid 1990s it is expected that the entire fleet of KC-135As will have been reengined. The KC-10 is considered the equivalent of three KC-135As (1:24).

Scheduling. Headquarters Strategic Air Command Tanker Operations Application Division (HQ SAC/DO8A) is the overall peace time manager for the scheduling of KC-135A, KC-135E, KC-135R, and KC-10 air refueling aircraft with CONUS refueling requests (1:2). Scheduling is performed on a quarter-by-quarter basis. To accomplish the scheduling, each receiver unit submits its quarterly requests for tanker support and each tanker unit submits their available tankers for the quarter. HQ

SAC/DO8A then matches the available tankers with the requests. One available tanker is matched against one tanker request.

Not all of the scheduling is directly performed by DO8A. The individual tanker units can directly schedule up to 75 percent of their refuelings with the receiver units to ensure that each tanker unit receives the necessary types of refuelings it must have to meet crew training requirements. These tankers that are scheduled by the local unit are referred to as precoordinated tankers. When a tanker unit submits its quarterly available tankers to DO8A, it also advises DO8A of its precoordinated tankers for the quarter.

Each quarter, fiscal year 1986 and before, DO8A would host a three day air refueling conference to bring together schedulers from the tanker and receiver units. Prior to each conference, DO8A would build the tentative quarterly refueling schedule based on the tanker and receiver unit's inputs. By bringing the schedulers face-to-face, receiver schedulers could negotiate the generation of additional available tankers to refuel receivers that were not satisfied in the tentative schedule. Due to budget cuts the conferences were discontinued in fiscal year 1987 (10).

Following the discontinuation of the conferences the number of requests that are unscheduled have increased while the actual number of requests made have remained constant. Each quarter an average of less than 500 tanker requests were unfilled when the air refueling conferences were conducted. Since cancellation, the average number of unfilled requests has increased to 765 per quarter. To decrease the number of unfilled requests DO8A is interested in the possibility of a tanker

performing a refueling against one receiver and then satisfying a second request. Rather than one tanker satisfying just one request, they are interested in the possibility of one tanker satisfying two requests (13).

During fiscal year 1986 the average quarterly number of tanker requests was 7733 of which 5393 had been precoordinated. Of the 2340 requests that were not precoordinated there were 2234 tankers made available. An average of 493 requests were unfilled per quarter (13). Daily, an average of 90 tankers were precoordinated, 39 requests for tankers were made, and only 30 tankers were made available for refuelings. Available tankers were fairly evenly distributed throughout the week days while requests for tankers were heaviest for Tuesday through Thursday (13).

HQ SAC/DO8A schedules the available tankers to the air refueling requests with the aid of a 1950's vintage Honeywell 6000 computer. The computer program schedules one tanker request against one receiver request. The scheduling of tankers to receivers is formulated as a transportation problem. A procedure presented by Ford and Fulkerson, in a Management Science article, is used to solve the problem (7).

Optimization for matching the available tankers to receiver requests is by minimizing the flight times the tankers must fly from their home base to the air refueling area.

Shortfalls in the Current Software. The current scheduling program used by HQ SAC/DO8A has many significant shortfalls. When the program was written in the early 1970s the scheduling of one tanker to refuel one request was adequate. It was not envisioned at the time that major

changes in the way tankers are scheduled would be needed. Also, other factors that should be considered when scheduling are not accounted for in the current program. Furthermore, the scheduling program's design does not easily lend itself to major modifications and additions.

The scheduling program was written in JOVIAL in the early 1970's. The Air Force has since discontinued use of JOVIAL as a programming language and only uses JOVIAL to maintain programs that were written in the language. The program code does not employ the use of structured programming techniques making it difficult to read and understand the flow of logic. Documentation is minimal. Since the program was written several years ago, numerous updates have been made to it. These updates further complicate the program code. Updates to the program have not kept pace with changes in regulations and policies. Due to lack of documentation and poor coding techniques, the extensive modifications needed to schedule a tanker against two refuelings would be difficult.

The computer lacks the temporary memory space necessary to permit expansion of the current scheduling program. When running the program all of the temporary memory space is absorbed due the size of the program (12).

The primary use of the computer is to run Top Secret material. Since HQ SAC/DO8A uses the computer to run unclassified data they have the lowest user priority. In order to gain access to the computer they must often come in on weekends and late at night. When other computer users are running certain levels of classified data, HQ SAC/DO8A is not permitted to access the computer (11).

Minimizing the total flight time of all the tankers from the tanker

base to the ARCP, as is currently done, will not necessarily minimize the overall flight time. It is possible for an air refueling area tc be near the base but the direction of the refueling track takes the tanker further from the base. After completing its refueling, the tanker may then have an extensive distance to return to the base. A tanker from another base may have a longer time to fly to the refueling area than the tanker that is scheduled, but, it may have a shorter flight time to return to the base. The total mission length may be less for the latter tanker to perform the refueling than for the tanker that was actually scheduled.

When the scheduling program was written, fuel availability and cost were not driving factors in scheduling aircraft. It was of no concern if a tanker completed its assigned refueling mission far from the base since a navigational training leg could be flown to make use of the time to fly back to the base. Since the program was written, requirements for navigational training legs have been changed due to fuel costs and it cannot be readily assumed that a tanker can automatically be placed in a position to fly a navigational training leg. Minimizing the flight time to return from the refueling area has become critical. Of primary concern to many unit schedulers is to fly as many missions as possible and expend as little time as possible per mission.

Each request for a tanker is prioritized based on the type of training mission to be conducted by the receiver aircraft. Category A training is normal recurring air refueling training. Category B training is in support of formal course training, exercises, predeployment air refueling, deployments, redeployments, rotations, and tests

(1:2-3). Category B training has a higher priority than A (11). The current scheduling program does not consider the priority of the training. Often, to insure that the category B refuelings are accomplished, receiver units will precoordinate these requests with tanker units.

Specific Problem

Primary Objective. The specific problem to be addressed by this thesis is that the number of air refueling requests that are not satisfied is too high and needs to be reduced. To satisfy more requests, HQ SAC/DO8A is interested in a scheduling program that can schedule one available tanker to satisfy two requests and also schedule a precoordinated tanker to a second request.

Today, with the greater demand on tanker resources, scheduling one tanker to one request does not meet the demand for tankers. Also, with the increased fuel delivery capability of the reengined KC-135s and the addition of the KC-10s, the possibility of a tanker performing more than one refueling per mission is greatly increased.

Subobjectives. The primary subobjectives are to correct the shortfalls in the current software program used by HQ SAC/DO8A. The current program does not give category B requests higher priority than category A. Also, the current program only minimizes the flight times to the air refueling areas. Because current scheduling policies are designed to minimize fuel, the total flight time of the mission must be considered.

Often when a receiver unit submits a request for a tanker they do not have in mind a specific time the refueling has to be accomplished.

The receiver unit simply desires a refueling sometime during the day or possibly anytime at night. However, when the request is submitted to HQ SAC/DO8A the scheduling program requires an exact time to be specified. The request would sometimes go unscheduled because a tanker was not available for the specified time. A tanker may have been available at another time that would have met the receiver unit's desires. As a means of considering the receiver unit's desires, the scheduling program should allow the receiver unit to submit either an exact time for the refueling or submit a time window in which they desire the refueling to be accomplished.

Another objective is to avoid the problems that degrade the usefulness of the current programming code, which does not allow for ease in expansion or major modifications. Programming techniques are available that make a program easier to understand, allow for expansion, and permit ease in major modifications.

II. Literature Review

This literature review begins by looking at the rules and regulations that must be considered when scheduling tankers to receivers. Next, an explanation of the current method used by Head Quarters Strategic Air Command Tanker Operations Applications Division (HQ SAC/DO8A) to schedule tankers is given. After examining the current system, two approaches to solving the scheduling task are examined.

Pertinent Rules and Regulations

Air Force Regulation 55-47, Air Refueling Management, is the primary regulation used by HQ SAC/DO8A for the scheduling of tankers to requests. It also governs the responsibilities of the tanker and the receiver unit schedulers. Almost all of the information that pertain to the scheduling task is contained in this regulation. The pertinent information is as follows:

1. Category B refuelings have a higher priority than category A refuelings (3:3).
2. Information to be included when requesting a tanker (3:5).
3. Information to be included in the available tanker and precoordinated tanker reports (3:4).
4. A tanker unit can precoordinate up to 75 percent of its daily tanker sorties (3:2).
5. Consideration should be given to limit the amount of time a tanker expends in non-refueling phases of flight (3:4).

Air refueling operations are normally conducted on established air refueling tracks. The locations and the availability times of these tracks have been precoordinated with Air Traffic Control Centers (ARTCC) (2:4-1).

An established restriction that must be considered when computing flight time is that a tanker must arrive at the Air Refueling Control

Point (ARCP) at least 15 minutes prior to the Air Refueling Control Time (ARCT) (4:4-7).

Current Scheduling System

The scheduling system currently used by HQ SAC/DO8A schedules one tanker to one receiver by minimizing the sum of the flight times to the air refueling areas.

The approach used for solving the scheduling task is based on a procedure presented by Ford and Fulkerson in a 1956 Management Science article (7). In the article, a procedure is demonstrated for solving a transportation problem that has several locations with surpluses to be distributed among locations with shortages. The procedure is based on Kuhn's "Hungarian Method" for the assignment problem. The procedure was originally designed to solve maximal flow problems in networks. The purpose of the article is to show how the procedure can be used to solve Hitchcock's transportation problem (5:24-32).

The procedure uses two matrices, a cost matrix and a zero matrix. The cost matrix is made up of the different costs that would be incurred in moving a unit of supply from a source to a location. The zero matrix is the same size as the cost matrix and has no values entered except for the current allocation of supplies and the possible locations for the future allocation of supplies.

The cost matrix is used to determine the most likely locations on the zero matrix to allocate the supplies. The zero matrix examines the most likely locations to determine if the corresponding sink has a shortage and surplus available at the corresponding source. After the zero matrix has allocated the supplies as much as possible, the cost

matrix is revisited to determine new locations to assign surplus. Allocations continue until all surpluses and shortages are zero.

For scheduling tankers to requests, the tankers are considered to be the sources and the requests as the sinks with the shortage. Each surplus location (tanker) starts with a surplus of one unit to be allocated to the requests. Each request is treated as a sink with the shortage being the number of tankers desired by the request. If the number of available tankers does not equal the demand for tankers, dummy tankers or dummy requests with demand are added as necessary to make them equal. The cost matrix is made up of the distances the tanker would have to fly from its home base to reach the refueling area of the request. If the area is beyond the allowable distance to send a tanker, the distance is set to a value of 9999 (7).

The approach presented by Ford and Fulkerson does an adequate job of solving the scheduling task of having each tanker perform one refueling. Since the approach is intended to solve a transportation problem with a surplus of more than one, it is not very efficient when all of the sources start with a value of one. The actual description of scheduling task more closely resembles a generalized assignment problem which allows each request to be assigned more than one tanker. Ignizio presents a much simpler procedure to perform the current scheduling task (9:333-337).

The task of scheduling a tanker to perform more than one refueling presents some problems that the procedure outlined by Ford and Fulkerson could not solve in its current state. Provision is not made for the tanker to either satisfy one or two requests. This is the same as

saying that one resource can satisfy two shortages. Also, the number of shortages must equal the number of surpluses for the procedure to work; a tanker can be assigned to satisfy either one or two requests, should the tanker be counted as one or two surplus resources?

Approaches to the Desired Schedule

HQ SAC/DO8A has stated three primary objectives to be considered when performing the scheduling task (10). The three objectives are to satisfy as many requests as possible, satisfy as many category B refuelings as possible, and minimize the total flight time for all of the tankers.

The objectives to be achieved when scheduling tankers to requests can be solved using either a single-objective approach or a multiple-objective approach.

A single-objective approach would incorporate all of the objectives one objective function. To do this, all of the objectives would be expressed in terms of a single measure. The advantage of the single-objective approach is that it seeks to find the optimal solution within the feasible region (6:19). By having one objective function, the problem only needs to be solved once. Other advantages to the use of single objective functions are described by Ignizio:

> Historically, it was the first to be developed and thus has received considerably more exposure, been put to more use, and generally considered to be at a relatively higher level of refinement. (9:19)

Also, the single-objective approach is appropriate when the best course of action is well defined, namely, the alternative that maximizes or minimizes a well-defined (scalar-value) objective function (14:133).

The single-objective approach requires determination of a credible weight that expresses each of the scheduling objectives by using a single measure (9:374). A problem with finding a single measure is that it may not be appropriate to express all of the objectives in terms of one common measure. With the objectives of maximizing satisfied requests and minimizing flight time it is difficult to express requests and flight time in terms of a single measure. The measure must show how much more important satisfying a request is to a decrease in flight time. That is, each request is equal to "X" number of hours of flight time, or an hour of flight time is equal to "Y" requests. Somehow, a measure must be developed that would be equivalent to saying that "X" number of requests equates to "Y" number of category B requests which equates to "Z" number of flying hours.

After a common measure is determined, the single objective problem can be formulated and solved. The optimum solution that is produced is only as credible as the weights assigned to each of the objectives.

When more than one objective must be considered in determining a solution, it is sometimes difficult to define what is the best solution. The multiple-objective approach is ideal when "no single objective function can adequately serve to compare the difference in desirability among feasible solutions" (14:133). A multiple-objective problem does not result in an optimal solution as does the single-objective approach. Rather, a "most preferred" solution is achieved (14:135).

Goal Programming is a multi-objective approach that strives toward several objectives simultaneously (8:172). By use of a preemptive goal programming approach, (sometimes known as lexicographical goal

programming) each of the objectives are prioritized according to their perceived importance and then optimized, one at a time in order of priority. As each objective is optimized, its solution becomes a constraint on any lower priority objectives. By solving the objectives in priority order it gives the best attainable value for the highest priority objective (14:137). The ranking or prioritizing of the objectives reduces the need to determine a common measure as is the case of the single-objective approach.

Goal programming is not without its disadvantages. It does not allow for small degradation in a high priority objective for a large improvement in a low priority objective. Ignizio points out that "the problem with the ranking approach is how to associate the results of a given solution to the satisfaction of the ranking" (9:375).

The most important criteria of any solution to a decision maker is that it is reliable and believable. Since the preemptive goal programming approach solves each of the objectives according to its perceived priority rather than assigning a numeric value to each objective, it offers the advantage of being understandable and easy to explain how the solution was obtained.

By definition, the scheduling problem presented in this thesis is a generalized assignment problem. A generalized assignment problem is one in which the more than one individual (e.g., tanker) can be assigned to a single job (e.g., request) (9:332). Formulation of the generalized assignment problem is as follows (9:333):

$$\text{minimize} \quad z = \sum_{i=1}^{m} \sum_{j=1}^{n} C_{ij} X_{ij}$$

Subject to

$$\sum_{i=1}^{m} X_{ij} = 1 \quad \text{for all } i,\ i = 1,2,\ldots,m$$

$$\sum_{j=1}^{n} C_{ij}X_{ij} \leq A_j \quad \text{for all } j,\ j = 1,2,\ldots,n$$

$$X_{ij} = 0,1 \quad \text{for all } i \text{ and } j$$

Where

$$X_{ij} = \begin{cases} 1 & \text{if job } i \text{ is assigned to worker } j \\ 0 & \text{otherwise} \end{cases}$$

C_{ij} = time required to perform job i by worker j

A_j = total amount of time that worker j can be assigned

Applying the generalized assignment problem to the scheduling task, the variables would be assigned as follows:

C_{ij}= flight time required for tanker i to satisfy request j

A_j = number of tankers needed to satisfy request j

$$X_{ij} = \begin{cases} 1 & \text{if tanker } i \text{ is assigned to request } j \\ 0 & \text{otherwise} \end{cases}$$

Instead of the constraint $\sum C_{ij}X_{ij} \leq A_j$, the constraint $\sum X_{ij} \leq A_j$ would be used for the scheduling task. The constraint will limit the tankers assigned to request j to the number of tankers requested by request j.

Summary

To approach the scheduling task, the problem will be formulated as a generalized assignment problem. All of the objectives that are to be

considered when scheduling tankers to receivers will be incorporated into the scheduling process through use of the goal programming approach.

III. Methodology

Overview

Presented in this chapter is a methodology for solving the tanker scheduling problem. The chapter begins with the formulation of the problem, then the approach to solving the problem, and finally an in-depth explanation of the programs and procedures used to solve the problem.

Introduction

The task of scheduling tankers to satisfy requests is not one that easily lends itself to being performed by paper and pencil with hopes of minimizing flight time while satisfying as many requests as possible. Due to the number of requests and tankers, accomplishing the schedule manually would require hundreds of hours to produce the best schedule. To simplify the task a computer solution to the scheduling problem is pursued.

The scheduling task is performed through the combined use of commercial software, tailored programs, and changes to the tankers and receiver scheduling inputs.

Using the methodology outlined in this chapter to perform the scheduling task, the following items must be available to the computer:

- Files
    - Tanker units input files
        - Precoordinated tankers
        - Available tankers
    - Receiver units request file
    - Permanent flight time files
        - From the base to the air refueling areas
        - From the air refueling area to the bases
        - From air refueling area to air refueling area

- Software
- Preprocessor
- Problem builder
- Integer programming package

The preprocessor takes the inputs from all of the files and computes all of the possible refuelings that can take place between the tankers and requests. When the preprocessor is finished, a problem builder program takes the preprocessor's output and puts together the problem to be solved by the integer programming package. The integer programming package solves the problem and produces the schedule. A diagram of the scheduling process as it is envisioned is shown in Figure 3.1.

Problem Formulation

The formation of the problem used in the scheduling of tankers to requests is as follows:

Lexicographically Minimize $\{(P_1),(P_2),(P_3)\}$

$$P_1 = \sum_{i \in A \cup B} d_i \qquad (1)$$

A = (set of all category A refuelings)
B = (set of all category B refuelings)

$$P_2 = \sum_{i \in B} d_i \qquad (2)$$

$$P_3 = C_{ij} \sum_{j=1}^{m} \sum_{i=1}^{n} T_j R_i + C_{ijk} \sum_{i=1}^{n} \sum_{j=1}^{m} \sum_{k=1}^{n} T_j R_i R_k$$

$$+ C_{ij} \sum_{j=1}^{q} \sum_{i=1}^{n} P_j R_i + C_{ij} \sum_{i=1}^{n} \sum_{j=1}^{q} R_i P_j \qquad (3)$$

$i = 1,2,\ldots,n$ (requests)
$j = 1,2,\ldots,m$ (available tankers)
$j = 1,2,\ldots,q$ (precoordinated tankers)
$k = 1,2,\ldots,n$ (requests)

Subject to:

Requests:

$$d_i + \sum_{j=1}^{m} T_j R_i + \sum_{k=1}^{n} \sum_{j=1}^{m} T_j R_i R_k + \sum_{j=1}^{m} \sum_{k=1}^{n} T_j R_k R_i + \sum_{j=1}^{q} P_j R_i + \sum_{j=1}^{q} R_i P_j = N_i \quad (4)$$

for all $i \in R$

R = {set of all requests}

Available Tankers:

$$\sum_{i=1}^{n} T_j R_i + \sum_{k=1}^{n} \sum_{i=1}^{n} T_j R_i R_k \leq 1 \quad \text{for all } j \in T \quad (5)$$

T = {set of all available tankers}

Precoordinated Tankers:

$$P_j + \sum_{j=1}^{q} P_j R_i + \sum_{j=1}^{q} R_i P_j = 1 \quad \text{for all } j \in P \quad (6)$$

P = {set of all precoordinated tankers}

d_i = deviation variable for request i = $0,1,2,\ldots,N_i$ — number of tankers short of what was asked for by request i

$$T_j R_i = \begin{cases} 1 & \text{if request i is satisfied by tanker j} \\ 0 & \text{otherwise} \end{cases}$$

$$T_j R_i R_k = \begin{cases} 1 & \text{if tanker j satisfies request i then request k} \\ 0 & \text{otherwise} \end{cases}$$

$$P_j = \begin{cases} 1 & \text{if precoordinated tanker j only performs its precoordinated refueling and does not satisfy any other request} \\ 0 & \text{otherwise} \end{cases}$$

$$P_j R_i = \begin{cases} 1 & \text{if precoordinated tanker j performs its precoordinated refueling and then satisfies a request} \\ 0 & \text{otherwise} \end{cases}$$

$$R_i P_j = \begin{cases} 1 & \text{if Precoordinated tanker j satisfies a request and then performs its precoordinated refueling} \\ 0 & \text{otherwise} \end{cases}$$

N_i = number of tankers asked for by request i

C_{ij} = flight time required to perform request i by tanker j

C_{ijk} = flight time required for tanker j to satisfy request i and then satisfy request k

HQ SAC/DO8A has identified three objectives that they desire the tanker scheduling program to accomplish. In order of priority beginning with the highest priority, the objectives are:

- Maximize the number of requests satisfied
- Maximize the number of category B requests satisfied
- Minimize the total flight time to perform the refuelings

A preemptive goal programming approach is used to solve the scheduling problem. This means that the desired goals to be achieved, when scheduling the tankers to the requests, are identified and prioritized according to their perceived importance. When solving the goal program, each of the objectives is optimized in order of priority with the highest priority objective being optimized first.

A goal programming approach seems to better lend itself to perform the scheduling task than a single-objective approach. In the single-objective approach, all of the goals would have to be expressed in terms of a single measure. Since the number of requests, and flight time, are two different measures, a weight or utility would be needed to establish a single measure. Using the goal programming approach requires the goals to be rank ordered.

The problem is formulated as a generalized assignment problem instead of a regular assignment problem. In the generalized assignment problem, each job, (e.g., request), can be assigned to more than one resource, (e.g., tanker). Not only can a request have more than one tanker assigned to it, a tanker can also satisfy more than one request.

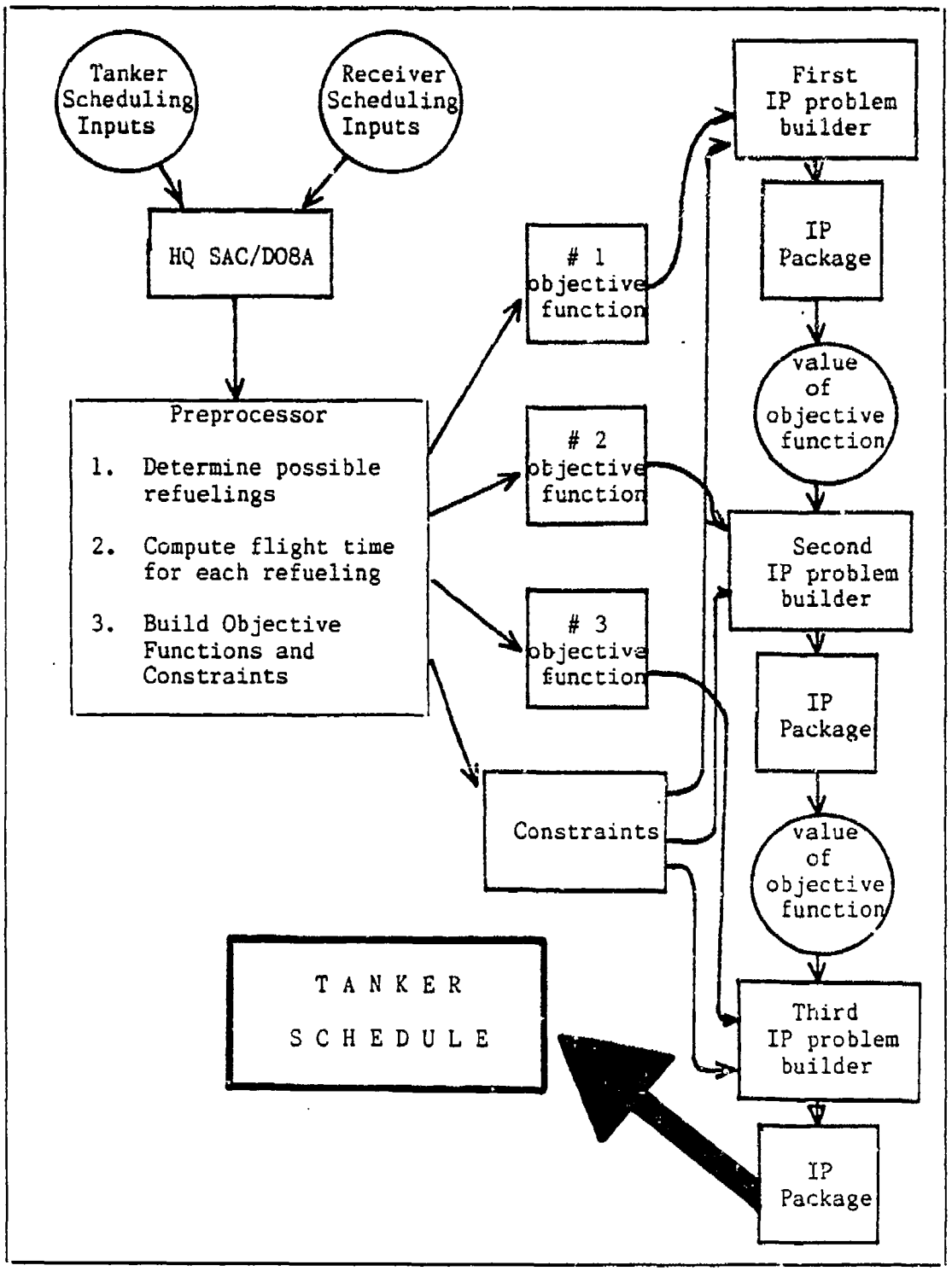


Figure 3.1 Tanker Scheduling Process

Two types of variables are used in the formulation of the objective functions and the constraints, a refueling variable (decision variable) and a deviation variable.

Each of the possible refueling that can take place between a tanker and request(s) is represented by a refueling variable. Each of the refueling variables is a binary, zero or one, variable. If a refueling is scheduled in the final solution, its associated refueling variable will be assigned a value of one.

Each request has associated with it a unique deviation variable (d_i). The value of the deviation variable reflects the achievement of each goal. It is an integer variable and can be assigned any value from zero up to a maximum value equal to the number of tankers asked for by the request (N_i). The value assigned to a request's deviation variable in the final solution indicates to what degree the desired goal of being satisfied has not been met. If the deviation variable is assigned a value of zero, the goal is achieved and the request is assigned the desired number of tankers.

Each request, available tanker, and precoordinated tanker has a constraint associated with it. The constraint associated with a request, constraint (4), is made up of the sum of all of the refueling variables that are associated with the request and also the request's deviation variable (d_i). This mathematical equation is then set equal to the number of tankers desired by the request (N_i).

An available tanker's constraint, constraint (5), is made up of the sum of all the refueling variables associated with the tanker. The equation is then set to less than or equal to one. Setting a tanker's

constraint to less than or equal to one means that the tanker can either be assigned to a maximum of one refueling or none at all. If at a later date, it is decided to schedule as many available tankers as possible, another goal would be added to the list of goals and prioritized. Also, a deviation variable would be assigned to each tanker and included in the tanker's constraint. The constraint would be changed from an inequality to an equality constraint.

A precoordinated tanker's constraint, constraint (6), requires that each precoordinated tanker either perform its precoordinated refueling alone (P_j=1), or perform its precoordinated refueling in conjunction with another request (P_jR_i=1 or R_iP_j=1). The constraint is set equal to one. The precoordinate's constraint can be set equal to one, if the precoordinated tanker is not assigned to satisfy a request, it can still perform its precoordinated refueling. The refueling variable, P_j, in constraint (6) represents the case where the precoordinated tanker only performs its precoordinated refueling.

A refueling variable that appears in a request's constraint will also appear in the associated tanker's constraint. A variable for an available tanker performing two refuelings will be found in three constraints, the tanker's, and each of the requests. Figure 3.2 shows an example of a common variable appearing in the different constraints.

Each of the goals is represented by its own objective function. The first objective, objective (1), is to maximize the number of requests satisfied; which is the same as minimizing the number of requests unsatisfied. To achieve this goal the objective function minimizes the sum of all of the deviation variables.

Example: Request R1 and Available Tanker A2

Constraint for R1: $D1 + A1R1 + A2R1 + A3R1 + \ldots = 3$

Constraint for A2: $A2 + A2R1 + A2R2 + A2R3 + \ldots \leq 1$

Request R1 requires three tankers. Notice that the variable "A2R1" appears in the request's constraint and also in the tanker's constraint.

"D1" is used as the deviation variable for R1. If D1 equals zero in the final solution, the request is satisfied. If D1 is assigned a value of two, than R1 will be scheduled two tankers less than desired.

Figure 3.2 Example of a Common Variable between Constraints

The second objective, objective (2), is to maximize the number of category B requests that are satisfied, which is the same as minimizing the number of category B requests that are unsatisfied. The objective function is to minimize the sum of all of the deviation variables associated with category B requests.

The third and final objective, objective (3), is to minimize the total flight time. The objective function is made up of the sum of all of the refueling variables. Each variable is preceded by a coefficient that is equal to the flight time to perform the refueling.

Solution Technique

Unlike the classical assignment problem, the generalized assignment problem, formulated above, does not automatically result in an integer solution (Ignizio:333). Thus, an integer programming (IP) computer package is used rather than a standard linear programming package to find the optimal solution.

For each of the goals a new integer programming problem is

formulated and then solved by the IP package. With three goals, three separate integer programming problems are solved. The goals are solved in order of priority starting with the highest priority. As each goal is solved, its objective function becomes a constraint for any succeeding goals and is set equal to its optimum value. This constraint insures that the value of the objective function at higher priority levels is not degraded in trying to achieve lower priorities. Also, the only change in the constraints from one problem formulation to the next is the addition of the objective functions for all higher priority goals. In each of the goals, the values of the individual variables are not important until the last objective is solved. It is in the final solution that the values of each variable determines which refuelings are scheduled.

It appears to be a lot of work creating and solving several problems, however, it offers the advantage of flexibility. If there is an addition or deletion of goal, or if the order of priority of the goals should change, it would not require extensive programming to incorporate the change.

Figures 3.3 shows a brief formulation of the first integer programming problem to minimize the number of requests not satisfied. The integer program package solves the problem. The value of each of the variables are not important. The objective function and its optimal value now become a constraint for the second programming problem.

The second programming problem is to minimize the number of category B refuelings that are unsatisfied. The formulation of the problem is similar to the first problem. The second formulation shares all of

the same constraints as the first problem with the addition of the objective function of the first goal. Figure 3.4 shows an example of the problem formulation. The second integer programming problem is solved by the integer programming package. Again, the values of each of the variables is not important. The only important piece of information is the solution to the objective function. The objective function and its solution now become a constraint for the third goal.

Objective Function:
Minimize R1 + R2 + R3 + R4 + ... + Rn (deviation variables for all of the requests)

Constraints:
Deviation variable + refueling variables for request 1 = 2
.
.
Deviation variable + refueling variables for request n = 1

Refueling variables for tanker 1 $\leq$ 1
.
.
Refueling variables for tanker m $\leq$ 1

Refueling variables for precoordinated tanker 1 = 1
.
.
Refueling variables for precoordinated tanker k = 1

Figure 3.3 Integer Programming Problem Formulation for the First Goal Minimize the number of Requests that are unsatisfied

The third and last programming problem is to minimizes the total flight time. The objective functions and solutions for the first two problems are included in the constraints. Because this is the last problem, the solution is the actual schedule. The values assigned to the variables determines which refuelings are scheduled. An example of the problem formulation for the third programming problem is given in Figure 3.5.

Objective Function:
Minimize R1 + R2 + R3 + ... (deviation variables for category B requests)

Constraints:

R1 + R2 + R3 + R4 + ... + Rn = number of unsatisfied requests

Deviation variable + refueling variables for request 1 = 2

.

Deviation variable + refueling variables for request n = 1

Refueling variables for tanker 1 $\leq$ 1

.

Refueling variables for tanker m $\leq$ 1

Refueling variables for precoordinated tanker 1 = 1

.

Refueling variables for precoordinated tanker k = 1

Figure 3.4 Integer Programming Problem Formulation for the Second Goal to Minimize the number of Category B refuelings unsatisfied

Objective Function:
Minimize 3.2 R1A1 + 5.4 B3R2 + 4.5 B3R2R3 + 5.1 A1R7 + ...
(Refueling variables and their associated mission length)

Constraints:

R1 + R2 + R3 + R4 + ... + Rn = number of unsatisfied requests

R1 + R2 + R3 +... = number of unsatisfied Category B requests

Deviation variable + refueling variables for request 1 = 2

.

Deviation variable + refueling variables for request n = 1

Refueling variables for tanker 1 $\leq$ 1

.

Refueling variables for tanker m $\leq$ 1

Refueling variables for precoordinated tanker 1 = 1

.

Refueling variables for precoordinated tanker k = 1

Figure 3.5 Integer Programming Problem Formulation for the Third Goal to Minimize the Total Flight Time

Scheduling

A change is made to the original scheduling inputs that are submit-

ted by the tanker and receiver units to increase the possibility of a request being satisfied. Often a receiver unit will desire a tanker to perform a refueling sometime during the day with no specific time in mind. Under the current scheduling system a specific air refueling control time (ARCT), the time at which the refueling is to begin, must be stated when requesting a tanker. To provide the receiver unit with the flexibility to request a refueling for a general period of time the new request format would allow the receiver unit to input the desired earliest and latest ARCTs rather than having to pinpoint a specific time when making the request. To the schedulers, this ARCT window indicates that any tanker can be scheduled to satisfy the request as long as the refueling begins any time between the earliest and latest ARCT. Providing an ARCT window instead of a single point in time increases the likelihood of the request being satisfied. If, on the other hand, a receiver unit must have the requested refueling at a specific time, all that has to be done is to give the earliest and latest ARCT the exact same time. Figure 3.6 shows the current inputs submitted by each receiver and the change in inputs to permit the requesting of an ARCT window.

There are two possible ways for a precoordinated tanker to perform its precoordinated refueling and satisfy a request:

- Precoordinated refueling first then the request
- Request first then the precoordinated refueling

No rules or guidelines exist on how to schedule the precoordinated tankers to perform more than their precoordinated refueling. Since precoordinated tankers have never been scheduled to satisfy requests,

many of the factors that would impact the schedule have not been resolved. To aid in the scheduling of the precoordinated tankers it is assumed that all of the precoordinated tankers are available to satisfy a request either before or after its precoordinated refueling. An addition is made to the scheduling inputs to allow the precoordinated tankers to state the earliest and latest ARCT of the precoordinated refueling. The precoordinated tanker inputs for the current system and the change in the inputs to specify an ARCT window is show in Figure 3.7.

Current Request's Inputs	New Request's Inputs
- Refueling Area	- Refueling Area
- Receiver Unit	- Receiver Unit
- Date/ARCT	- Date
- Number of tankers requested	- Earliest ARCT
- Refueling time	- Latest ARCT
- Refueling Category	- Number of tankers requested
- Receiver aircraft type and numbers	- Refueling time
- Offload	- Refueling Category
	- Receiver aircraft type and numbers
	- Offload

Figure 3.6 Current Request's Inputs versus New Request's Inputs

Programs and Procedures to Create the Integer Programming Problm

The integer programming problem to be solved by the integer programming package is created by two programs, a preprocessor and a program builder. Neither the preprocessor nor the program builder have been automated on a computer yet. Both programs were performed manually for the toy problem.

Current Precoordinated Inputs	New Precoordinated Inputs
- Tanker base	- Tanker base
- Refueling Area	- Refueling Area
- Receiver Unit	- Receiver Unit
- Date/ARCT	- Date
- Offload	- Earliest ARCT for Precoord
- Type and number of receiver aircraft	- Latest ARCT for Precoord
	- Offload
	- Type and number of aircraft

Figure 3.7 Current Precoordinated Tanker Inputs versus New Precoordinated Tanker Inputs

The preprocessor takes the tanker and receiver scheduling inputs and determines all of the possible refuelings. It assigns variables to each of the refuelings and computes their flight times. After the variables have been identified, the preprocessor builds each of the objective functions and the list of constraints.

The program builder takes the objective functions and constraints produced by the preprocessor and builds each of the integer programming problems. When the integer programming package is finished solving one of the problems, the program builder takes the solution to the problem to build the next integer problem.

Incorporated in the preprocessor is the assumption that a request can have a time window in which the refueling can take place. Rather than having a fixed ARCT, an earliest and latest ARCT is associated with the request. This scheduling change will be explained prior to the explanation of the preprocessor.

To help in developing the methodology of the preprocessor a sample problem is used. This sample problem, or toy problem, is a miniature scheduling problem with tanker and receiver scheduling inputs that exercise all possible types of tanker/receiver combinations.

Toy Problem. A toy problem is used to develop the algorithm used in the preprocessor for computing the feasible refuelings between a tanker and receiver(s). The toy problem is an abbreviated version of a day's request and tanker inputs that are to be scheduled.

The bases and refueling areas used in the toy problem were chosen because of their close geographical proximity to each other. The available tankers, precoordinated tankers, and requests were selected to cover the different refueling possibilities. There are more requests than available tankers. This is done to place the integer program package into a situation where it must consider using tankers for more than one refueling. Even if there were more requests than tankers, it may not be possible to satisfy all requests.

The toy problem is made up of three tanker bases; Altus, Barksdale, and Carswell. Each base has both available tankers and precoordinated tankers. There are a total of nine available tankers at the bases, and eleven precoordinated tankers. The available tankers and the precoordinated tankers are listed in Figure 3.8.

Fourteen requests are made that would require 16 tankers to completely satisfy all of the requests. A listing of the requests can be found in Figure 3.9.

All of the times are in ZULU time. ZULU is a common acronym for Greenwich Mean Time. One days schedule is from 0800 ZULU, 0200 Eastern Standard Time, to 0800 Zulu of the following day. The refueling times are in hours and minutes.

Available Tankers:

Base	Earliest ARCT	Latest ARCT
ALTUS	1700	2400
ALTUS	1400	2000
BARKSDALE	1600	2000
BARKSDALE	1200	1600
BARKSDALE	1800	2200
BARKSDALE	0800	2400
CARSWELL	1100	1400
CARSWELL	2330	0730
CARSWELL	1500	2000

Precoordinated Tankers:

Precoord Base	Refuel Track	Refuel Time	ARCT Window
ALTUS	13E	0 30	0800-1000
ALTUS	330W	0 35	2100-2100
ALTUS	112W	0 30	1230-1430
BARKSDALE	101S	0 44	1500-1500
BARKSDALE	313N	0 20	2400-0200
BARKSDALE	112W	0 30	2400-0200
CARSWELL	104E	0 30	1210-1610
CARSWELL	102A	0 35	1100-1100
CARSWELL	13W	0 30	2300-0100
CARSWELL	112E	0 30	0500-0700
CARSWELL	650	2 30	1400-1600

Figure 3.8. Available and Precoordinated Tankers

The variable used to represent each of the available tankers is the first letter of the base plus the tanker number at the base. For the first tanker at Altus it would be "A1." The request's variable is an "R" plus the request number. A precoordinated tanker's variable is a "P" plus the base initial plus the tanker number at the base. For the Altus precoordinated tanker refueling on track 330W the variable would be "PA2."

Refuel Track	Refuel Time	Number Tankers	Category	ARCT Window
613	2 10	2	A	1300-1500
104E	0 30	1	A	2300-0100
650	3 00	1	B	0100-0300
13E	0 30	1	A	1200-1200
112W	0 30	1	A	1600-1800
330W	0 35	1	A	1230-1430
112E	0 30	2	B	1400-1600
101N	0 55	1	A	1400-1600
11BW	0 40	1	B	1200-1400
310W	0 15	1	A	2200-0100
330E	0 30	1	B	1630-1830
112W	0 30	1	A	1430-1630
102A	0 35	1	B	0830-1030
110E	0 30	1	A	1300-1500

Figure 3.9 List of Requests

As each tanker is matched to a request the variables for the tanker and receiver are added together to form the refueling variable. The refueling variable for the first available tanker at Altus refueling the second tanker request would be "A1R2." If the same tanker can refuel the second request and then refuel the fifth request the variable would be "A1R2R5." The order in which the requests are listed in the variable indicates the order in which the tanker is to satisfy the request. For the first precoordinated tanker at Barksdale to perform its precoordinated refueling and then satisfy the third request the variable would be "PB1R3." If instead, the precoordinated tanker satisfies the third request and then its precoordinated refueling, the variable would be "R3PB1."

The flight time from the base to the refueling area is measured as a straight line distance from the airfield coordinates listed in the Flight Information Publication (FLIP) Enroute Supplement to the ARCP of

the refueling area (AP). If the flight time to the ARCP is less than thirty minutes, it is set to thirty minutes to account for the climb time needed by the tanker to reach the air refueling altitude prior to the start of refueling. The flight time from the refueling area to the base is measured from the end air refueling point to the airfield coordinates plus an additional twenty minutes to account for penetration and landing time. The coordinates for the ARCP and air refueling point for the air refueling areas can be found in the DOD FLIP Area Planning AP/1B. A 15 minute orbit time is used at all ARCPs.

The time constraints used to determine the possible refueling variables for an available tanker satisfying one request are as follows:

1. Total mission time $\leq$ 6 hours
2. Flight time from the base to the ARCP $\leq$ 1 hour 30 minutes
3. Flight time from the end refueling point to the base $\leq$ 1 hour 30 minutes
4. Flight time from the base to the ARCP plus the flight time from the end refueling point to the base $\leq$ 2 hours 30 minutes

For the available tanker satisfying a single request scenario, the only non-refueling flight time is spent flying to and from the refueling area. However, in addition to flying to and from the base, when an available tanker satisfies two requests or a precoordinated tanker performs its precoordinated refueling and satisfies a request, an additional amount of non-refueling time is spent in flying between the refueling areas. It is assumed that one would be willing to allow for a greater amount of time to be spent in non-refueling portions of flight when two refuelings are performed than for the case of a single refueling. Therefore, the non-refueling flight time for the tanker satisfying two requests is constrained to 3 hours 45 minutes rather than

the 2 hours and 30 minutes of non-refueling time for the tanker satisfying a single request. The time constraints for the available tanker satisfying two requests and for the precoordinated tanker performing its precoordinated refueling and satisfying another request are as follows:

1. Total mission time $\leq$ 6 hours
2. Flight time from the base to the first ARCP $\leq$ 1 hour 30 minutes
3. Flight time from the end refueling point of the first refueling to the second ARCP $\leq$ 2 hours 10 minutes
4. Flight time from the end refueling point of the second refueling to the base $\leq$ 1 hour 30 minutes
5. Flight time from the base to the first ARCP plus flight time from the first request's end refueling point to the second ARCP plus the flight time from the second request's end refueling point to the base $\leq$ 3 hours 45 minutes

MIP83

MIP83 is an integer program package for use on a microcomputer. It is used to solve the toy problem to determine the feasibility of performing the scheduling task with an integer programming package and the practicality of using a microcomputer.

MIP83 has the capability of solving a problem up to 1200 units in size. The units are calculated as follows (15:10-2):

	Units
Each variable name	1
Each less-than-or-equal constraint	2
Each greater-than-or-equal constraint	3
Each equality constraint	2
Each bound variable	0

For example, a problem with 34 requests, 33 precoordinated tankers, and 22 tankers that result in 1000 variables could be solved.

For the toy problem the units used are:

174 variables	=	174
27 equality constraints	=	54
9 less-than-or-equal constraints	=	27
		255 total units

MIP83 has the capability to read in information from a spreadsheet program (e.g., LOTUS 123) and solve it. Use of a spreadsheet is limited to 256 variables. Because of this limitation a spreadsheet would be practical only for very small integer problems.

Preprocessor

The scheduling inputs provided by the tanker and receiver units do not lend themselves for immediate execution by the integer programming (IP) package. A preprocessor is used to determine all of the necessary refueling variables to be processed by the IP package.

The preprocessor is an algorithm/procedure that determines all of the possible refuelings that can occur between the tankers and requests, computes the flight time and assigns a variable to each of the possible refuelings, and builds the objective functions and constraints to be solved by the IP package. One objective of the preprocessor is to minimize the amount of work that must be performed by the IP package. To decrease the workload, the preprocessor tries to minimize the number of variables that it creates. It does this by filtering out any refuelings that are not possible.

The preprocessor matches the requests and tankers to determine all possible refuelings. There are five types of variables that can be used to represent the possible outcomes when comparing the tankers to the requests:

- One tanker satisfying one request
- One tanker satisfying two requests
- Precoordinated tanker performing its precoordinated refueling and satisfying one request
- Precoordinated tanker performing only its precoordinated refueling
- Request being unsatisfied

A major step in formulating the problem is to determine all possible refuelings. A possible refueling is one in which a tanker can satisfy a request(s) within the limits of the established constraints for flight times and ARCT time windows. For each request, the preprocessor determines all of the tankers that can satisfy the request. Also, for each tanker, the preprocessor must determine all of the requests the tanker can satisfy. By performing these matches to determine the possible refueling combinations, requests and tankers that do not comply with the set of constraints can be sifted out. This sifting decreases the number of variables and possibly the number of constraints the IP package must process.

Three permanent files and three scheduling files are accessed by the preprocessor. The three permanent files consist of a file containing the flight times from each refueling area to all other refueling areas, a file of flight times from each base to all of the refueling areas, and a file of flight times from each refueling area to all of the bases. The three scheduling files are the day's inputs from the tanker and receiver units. These scheduling files consist of a file of the available tankers, tanker requests, and the precoordinated tankers. From the input files the preprocessor creates the variables that represent the various tanker/request(s) combinations.

Figure 3.10 shows the desired contents of each of the scheduling

files submitted by the tanker and receiver units. Each of the files contains the minimal information necessary to accomplish the scheduling task. If special reports are to be generated after the tankers and requests have been scheduled, additional information may have to be added to the input files.

Precoordinated Tankers	Available Tankers	Requests
- tanker base - refueling area - earliest ARCT - latest ARCT - refueling time - offload - type and number of receiver aircraft - receiver unit	- tanker base - earliest ARCT - latest ARCT - date	- receiver Unit - refueling area - date - earliest ARCT - latest ARCT - number of tankers requesting - refueling category - offload - type and number of receiver aircraft - refueling time

Figure 3.10 Contents of the scheduling files submitted by the tanker and receiver units.

Constraints. Before a tanker and request(s) combination is determined to be a possible refueling, they must first demonstrate adherence to a list of established flight time constraints. When the preprocessor is considering a tanker and request combination, if the combination should fail to satisfy any one of a list of constraints it is dropped from further consideration. Most of these constraints are established to limit the amount of time a tanker expends in non-refueling flight.

The constraints used in the preprocessor represent a general consensus among HQ SAC/DO8A of what might be important when scheduling a tanker to two refuelings. When the idea first surfaced for performing

two refuelings with one tanker, the schedulers at HQ SAC were not certain of what constraints would and would not be important. The point to be made here is that even if all of the constraints are not incorporated in the preprocessor, they can be added and deleted as necessary.

One Available Tanker Satisfying One Request. When considering the possible refueling of an available tanker satisfying one request, the constraints that must be satisfied are:

- total mission length
- flight time from the base to the ARCP
- flight time from the end refueling point to the base
- non-refueling time

The constraints on flight time from the end refueling point to the base and the constraint on the non-refueling time are not considered in the current scheduling program operated by HQ SAC/DO8A. By not considering the flight time from the end refueling point to the base, a tanker could be scheduled to perform a refueling on a track that takes the tanker directly away from the base. When the tanker reaches the end refueling point it may be left with an enormous amount of time to fly back to the base. To decrease the non-refueling phases of flight, a ceiling is placed on the flight time to return to the base.

A tanker scheduler may not be willing to spend the necessary flight time to send a tanker to a refueling area that requires both the maximum time to get to that area and also requires the maximum time to return. To satisfy this constraint, a limit has been placed on the total amount of non-refueling time a tanker needs to fly to the area and return.

One Available Tanker Satisfying Two Requests. Constraints for the one tanker satisfying two requests share similar constraints with the

one available tanker satisfying one request. However, the time to fly from the first refueling area to the second is considered as part of the non-refueling time. The constraints to be considered are:

- total mission length
- flight time from the base to the ARCP
- flight time from the first end refueling point to the ARCP of the second refueling
- flight time from the end refueling point of the second refueling to the base
- non-refueling time

The total mission time and the non-refueling time computation is shown in Figure 3.11.

Precoordinated Tanker Satisfying One Request. Constraints for the refueling combination of a precoordinated tanker satisfying a request are similar to those for the one available tanker satisfying two requests. The precoordinated refueling is treated as a request when applying the constraints.

There are additional considerations when scheduling a precoordinated tanker to a request versus an available tanker satisfying a request. One of the major problems with the precoordinated tanker performing an additional refueling with a request is the actual availability times of the tanker. Is the precoordinated tanker available to meet a request before it performs its precoordinated refueling, or, is the tanker available to meet the request after its precoordinated refueling? HQ SAC/DO8A would like to consider the possibility of the precoordinated tanker satisfying the request either before or after its precoordinated refueling. To place restrictions on when a tanker can satisfy a request would require additional information to be included in the precoordinated tanker's scheduling file.

One available tanker satisfying one request

Non-refueling time = (flight time from the base to the ARCP) + (flight time from the end refueling point to the base)

Totnl mission time = (flight time from the base to the ARCP) + (orbit time) + (refueling time) + (flight time from the end refueling point to the base)

One available tanker satisfying two requests and Precoordinated tanker performing its precoordinated refueling and satisfying a request

Non-refueling time = (flight time from the base to the first ARCP) (flight time from end refueling point of the first refueling to the ARCP of the second refueling) + (flight time from the end refueling point to the base)

Total mission time = (flight time from the base to the first ARCP) + (orbit time) + (refueling time of the first refueling) + (flight time from end refueling point of the first refueling to ARCP of the second refueling) + (orbit time) + (refueling time of the second refueling) + (flight time from the second end refueling point to the base)

Figure 3.11 Determining mission length

General Procedure. The procedure is designed to decrease the number of mixing and matching comparisons that take place between the tankers and requests to find the possible refuelings. What the preprocessor is intended to do is avoid comparing every tanker to every request. The algorithm tries to limit the number of tanker and receiver comparison tests. However, the algorithm is not designed with the sole purpose of finding a way of making the fewest comparisons possible while neglecting "good" structured programming practices of being understandable and maintainable. Some sacrifices had to be made in efficiency to exercise acceptable structured programming techniques.

Organization. Each of the different types of possible refuelings are computed separately as shown in Figure 3.12. The computation of the precoordinated tanker performing its precoordinated refueling and then a request, and the precoordinated tanker satisfying a request first and then performing its precoordinated refueling share information back and forth to determine the possible refuelings. However, it would be easy to remove one of the precoordinated tanker's scenarios.

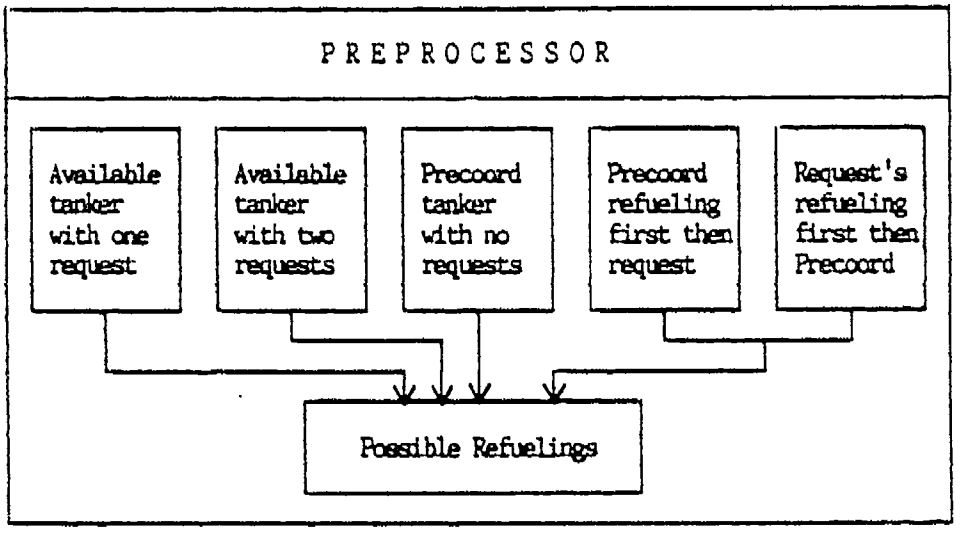


Figure 3.12 Possible refuelings as computed in the Preprocessor

By having the preprocessor compute the different possible refuelings independently, the workload required to add or delete a possible refueling type is simplified over mixing the different refuelings together. If at a later date it is decided that it would be desirable for an available tanker to be used to satisfy three requests, all that needs to be done is the addition of a module to the preprocessor that determines the possible refuelings with a tanker satisfying three requests. Figure 3.13 is an example of the addition.

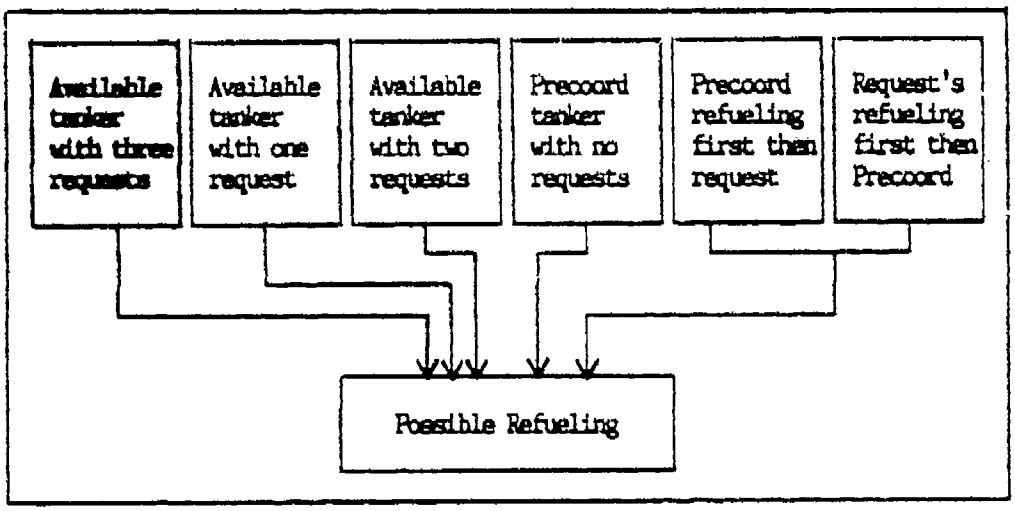


Figure 3.13 Example of adding a new option to the preprocessor

File Sorting. As a first step in processing the tanker and requests information, the records in each of the scheduling files are sorted to improve the efficiency of the preprocessor. The reason for this sorting is to have the files in a format that will lead to a decrease in the number of comparisons between tankers and receivers.

The available tanker's file and the precoordinated tanker's file are sorted by the base where the tanker is located.

The request file is sorted by the air refueling area. Any request that asks for more than one tanker has its earliest and latest ARCTs set to the same value. Since the request asked for more than one tanker, all of the tankers satisfying the request must perform the refueling at the same time. If the earliest and latest ARCTs are not equal, additional constraints would have to be added to the problem to ensure that all of the scheduled tankers satisfy the request at the same time. In the toy problem, when a multiple tanker request is encountered the

latest ARCT is set to the value of the earliest ARCT.

Searching for Possible Refueling. After the sorting of the files is complete, the files are compared to determine all of the possible refuelings.

One Available Tanker Satisfying One Request. Rather than comparing each request to each available tanker, the comparisons are first performed between refueling areas and bases. If each available tanker is compared to each request, the number of comparisons is equal to the number of available tankers multiplied by the number of requests. The number of comparisons increase rapidly as the number of tankers and requests increase. Instead, each request's refueling area is compared to each available tanker base. If a base cannot meet the flight time constraints for flying either to or from the request's refueling area it is not necessary to compare each of the requests for the refueling area with each available tanker at the base. Figure 3.14 shows a flow diagram for determining the case of an available tanker satisfying one request.

First, each requested refueling area is compared to an available tanker base. An example of the comparisons is shown in Figures 3.15 and 3.16. If the base cannot satisfy a request at the air refueling area, the next refueling area is compared to the same base. To determine if a base can perform the refueling, the flight times from the base to the ARCP, from the end refueling point to the base, and the non-refueling time are compared to their maximum allowable values.

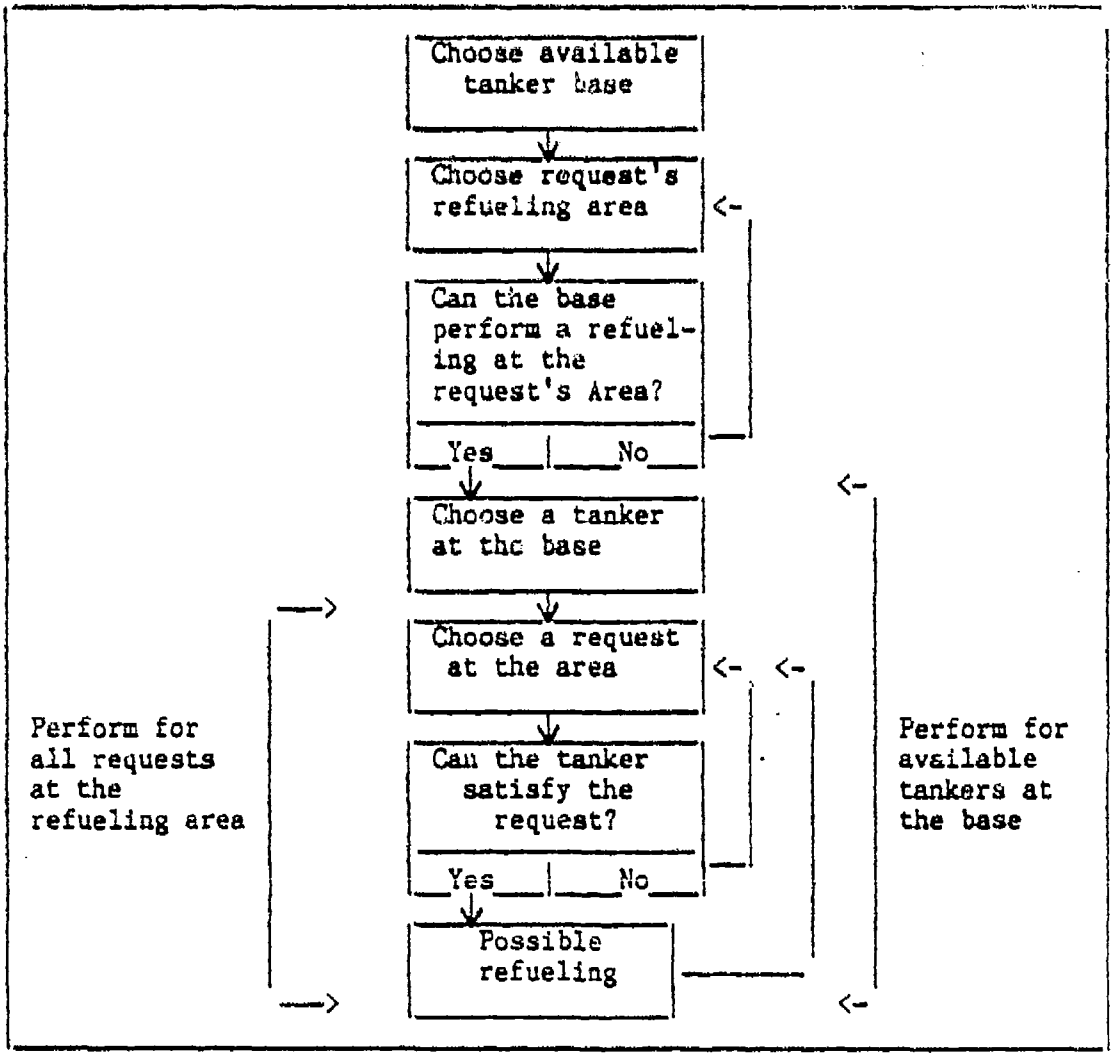


Figure 3.14 Flow diagram for determining the available tankers satisfying a request

If the flight time constraints are met by the base, each request for the refueling area is compared to each of the available tankers at the base as shown in Figure 3.11b. This is done by taking an available tanker at the base and comparing it to each of the requests for the refueling area. The ARCTs of the tanker are compared to the ARCTs of the request and the total mission time is computed to determine if the

tanker can satisfy the request. If the tanker and the request meet the constraints they are considered as a possible refueling combination. After the comparisons are complete for the tanker, then the next available tanker at the base is compared to all of the requests for the given refueling area. This continues until all of the base's available tankers have been compared to all of the area's requests.

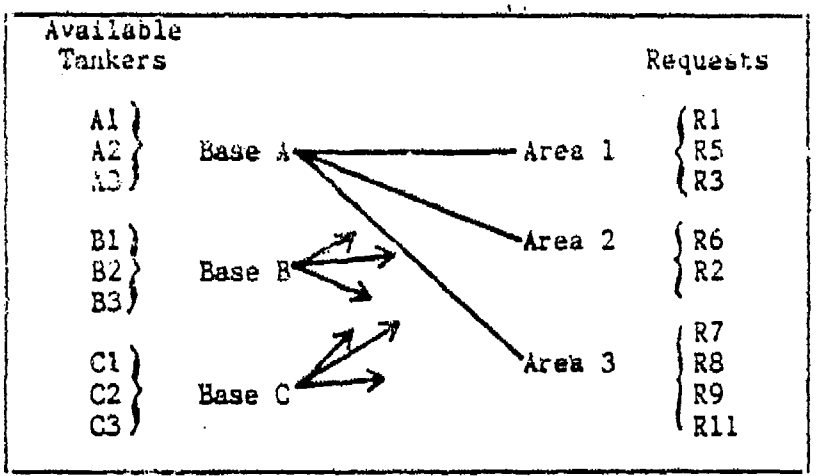


Figure 3.15 Comparing Bases to air refueling areas

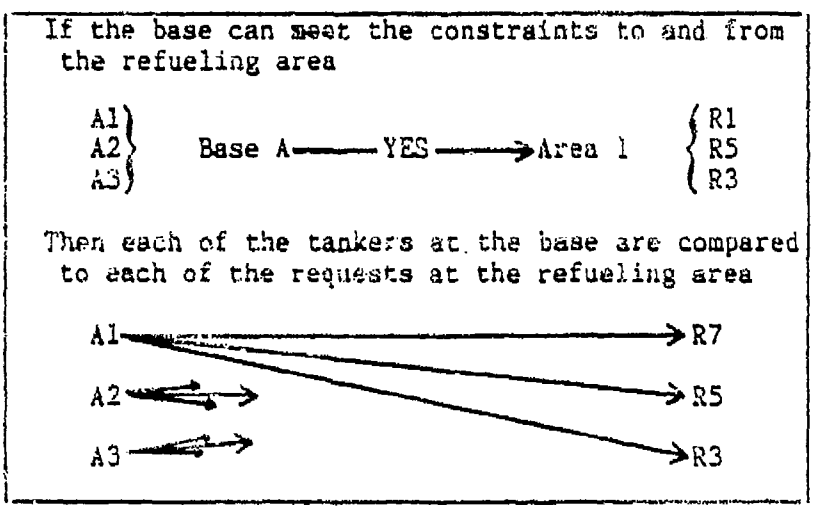


Figure 3.16 Comparing available tankers to requests

One Available Tanker Satisfying Two Requests. One possible means of satisfying requests is to use an available tanker to satisfy two requests. Again, an attempt is made to minimize the number of comparisons between available tankers and requests. To reduce the number of comparisons the first step is to determine all of the requests that can possibly be paired together. Once this is determined, then the possible pairs are compared to the available tankers.

Determining Request Pairs. When two requests are paired together, the order of the requests in each of the pairs is important. An acceptable pair with request "X" being followed by request "Y" does not imply that the request pair of "Y" then "X" will be acceptable. Another important reason for ordering the pairs is that the flight time required to perform pair (X,Y) will most likely not equal the flight time for the pair (Y,X). To determine the request pairs, each requested refueling area is compared to the other refueling areas. Under consideration is the possibility of a refueling taking place on the first refueling area and a subsequent refueling taking place at the second refueling area.

If the two refueling areas can be paired together, then each of the requests at the first area is compared to each of the requests at the second area. A list of the requests that can be paired together is kept. By design, this list of possible refueling pairs will automatically be sorted by the refueling areas of the first request. A flow diagram of the comparison is shown in Figure 3.17.

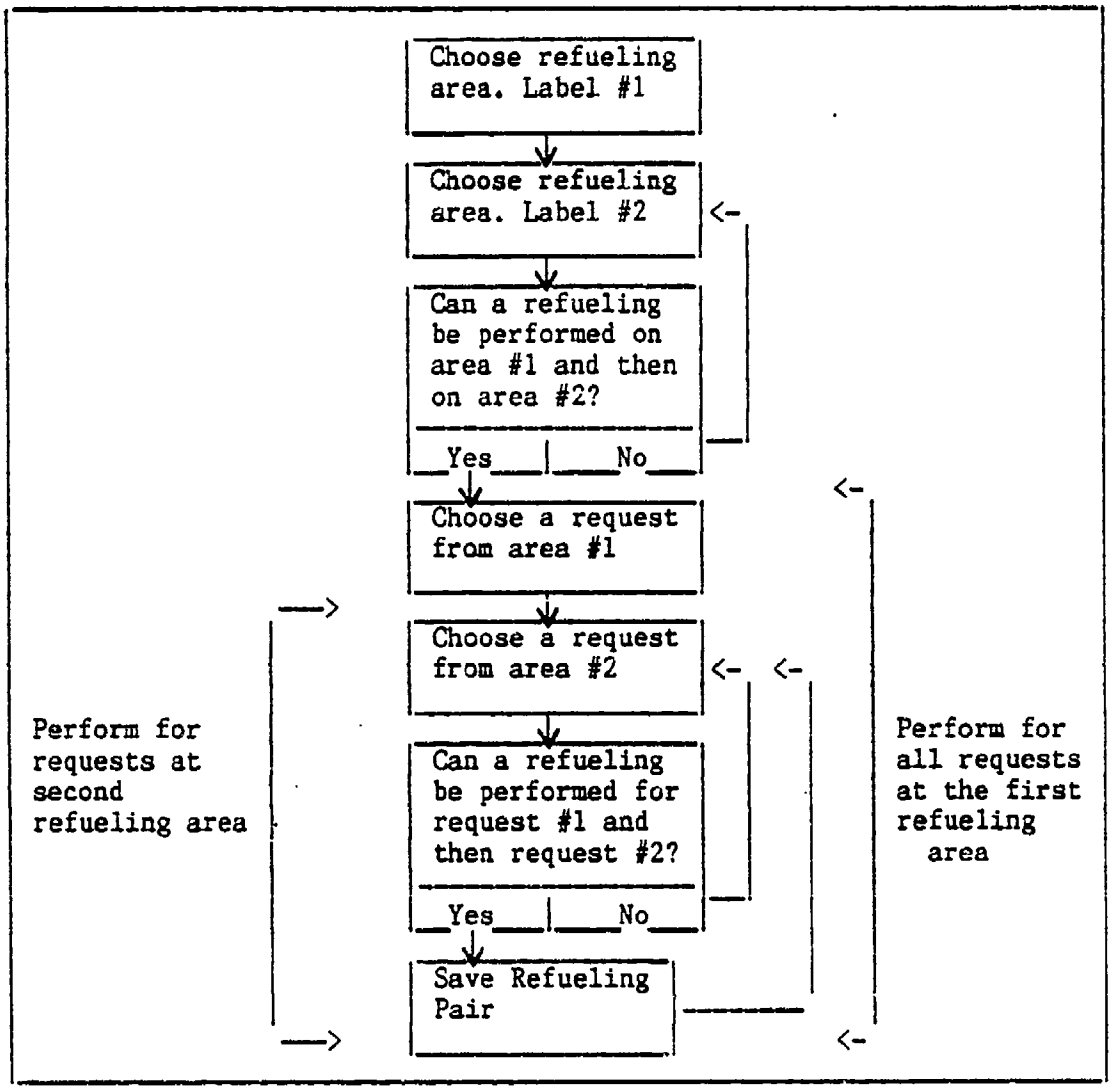


Figure 3.17 Flow diagram for determining the request pairs

After all of the refueling areas have been compared to each other and all of the requests pairs have been determined, the list of request pairs is searched. The request pairs are searched to determine all of the possible request pairs that share the same two requests but in different order. This will be used later to decrease the number of

variables. If the same tanker can perform a possible refueling with request pair (X,Y) and also a possible refueling with pair (Y,X), the refueling with the higher flight time will be discarded.

Comparing Available Tankers to the Request Pairs. To determine the available tankers that can successfully satisfy two requests, the file of available tankers is compared to the file of possible refueling pairs. Comparisons between tankers and request pairs are performed in the same manner as previous comparisons. The tanker bases are compared to the refueling area of the first request in the pair. If the tanker base can perform a refueling at the first request's refueling area, each of the tankers at the base are compared to each of the request pairs having the same refueling area for the first request to determine all of the possible refuelings. After all of the possible refuelings are determined for the base and the refueling area, the base than compares itself to the next refueling area. This continues until all of the bases have been compared to all of the refueling areas. Figure 3.18 shows a diagram for determining the possible refuelings.

Precoordinated Tanker satisfying one request. A possible refueling combination is the scheduling of a precoordinated tanker to perform its precoordinated refueling as well as satisfy a request. Of all the mix and match comparisons that are to be performed, this is probably the one that is most sensitive to efficiency since up to 75 percent of the tankers can be precoordinated. The precoordinated tanker can either perform the request prior to its precoordinated refueling or after.

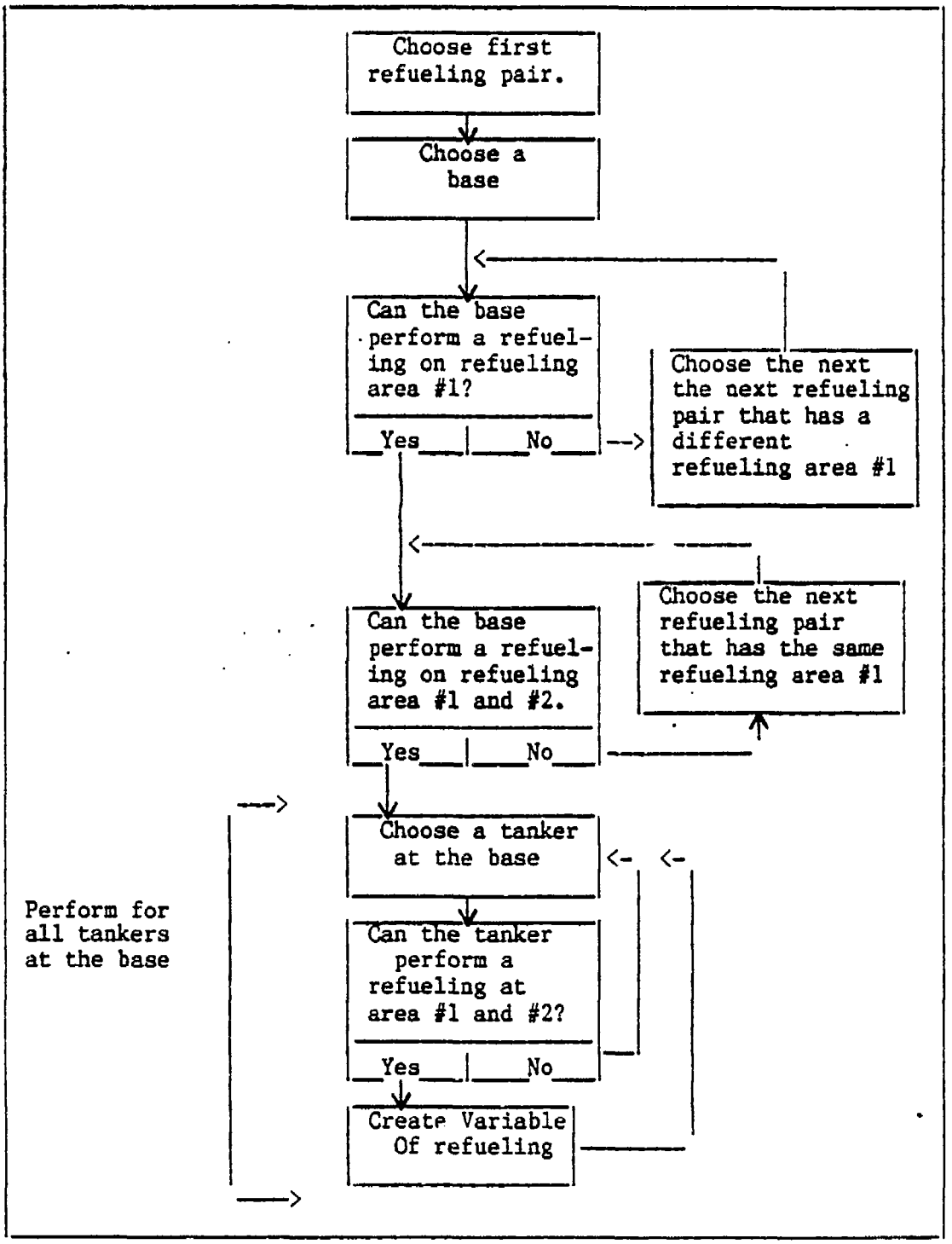


Figure 3.18 Flow diagram for comparing the available tankers to the request pairs

The first step in the process is to determine all of the precoordinated tankers that cannot perform a refueling in addition to its precoordinated refueling. If the precoordinated tanker's mission length, or, if either the time to travel from the base to the precoordinated refueling area or from the precoordinated refueling area to the base exceed the maximum allowable time, the precoordinated tanker can only perform its precoordinated refueling and is not considered for a second refueling. A flow diagram of the process is shown is Figure 3.19.

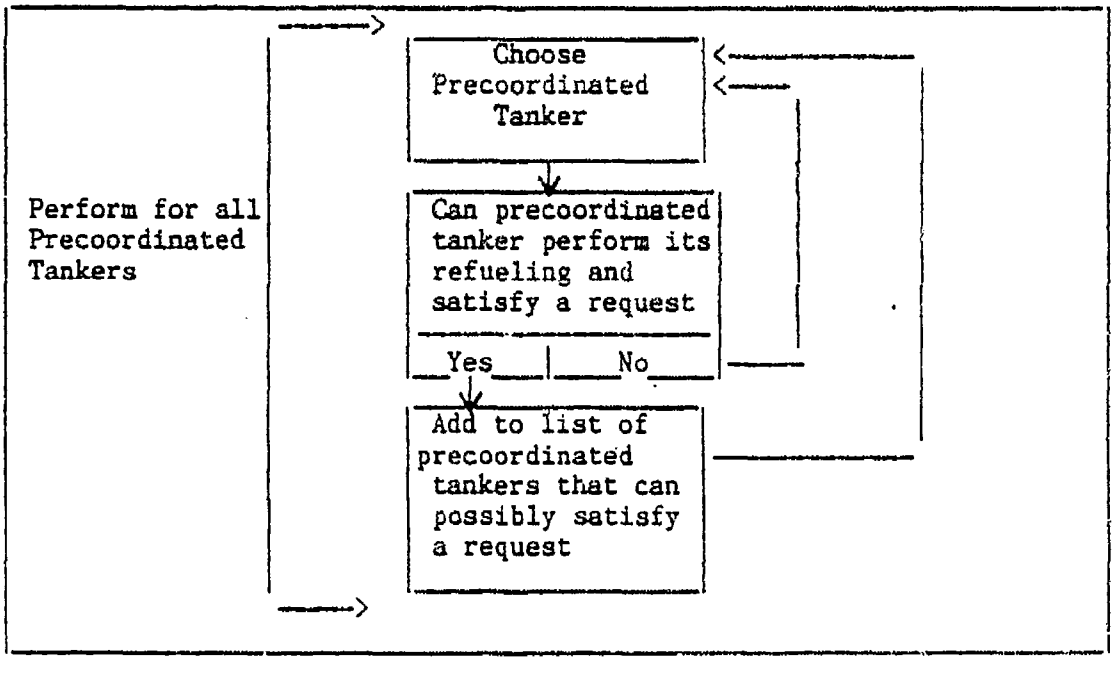


Figure 3.19 Flow diagram for determining precoordinated tankers that can possibly satisfy a request

The list of precoordinated tankers that can possibly satisfy a request is compared to the requests. The flow diagram is shown in Figures 3.20a and 3.20b.

The process begins by first comparing the precoordinated tanker bases to the request's refueling areas. A tanker base and refueling area are compared to see if the base can satisfy the request as a second refueling. If it can, then each of the precoordinated tankers at the base are compared to each of the requests at the area to determine the possible refuelings for the case of the tanker performing its precoordinated refueling first and then satisfying a request. After completing the comparison, the same base and refueling area are compared to see if the base can perform a refueling at the refueling area as its first refueling. If it can, then each of the precoordinated tankers at the base are compared to each of the requests at the area to determine all possible refuelings for the case of the tanker satisfying a request first and then performing its precoordinated refueling.

If the base could perform a refueling at the request's refueling area as its first refueling and as its second refueling, the possible refuelings that were determined are compared. The comparison is performed to find any situations in which a tanker can satisfy the same request either as its first refueling or its second refueling. If such a case exists, the two possible refuelings are compared to each other to determine which has the lower flight time. The one with the lower flight time is retained while the other refueling is discarded.

After this process is complete, the same base is compared to the next request area. This continues until all of the bases have been compared to all of the refueling areas.

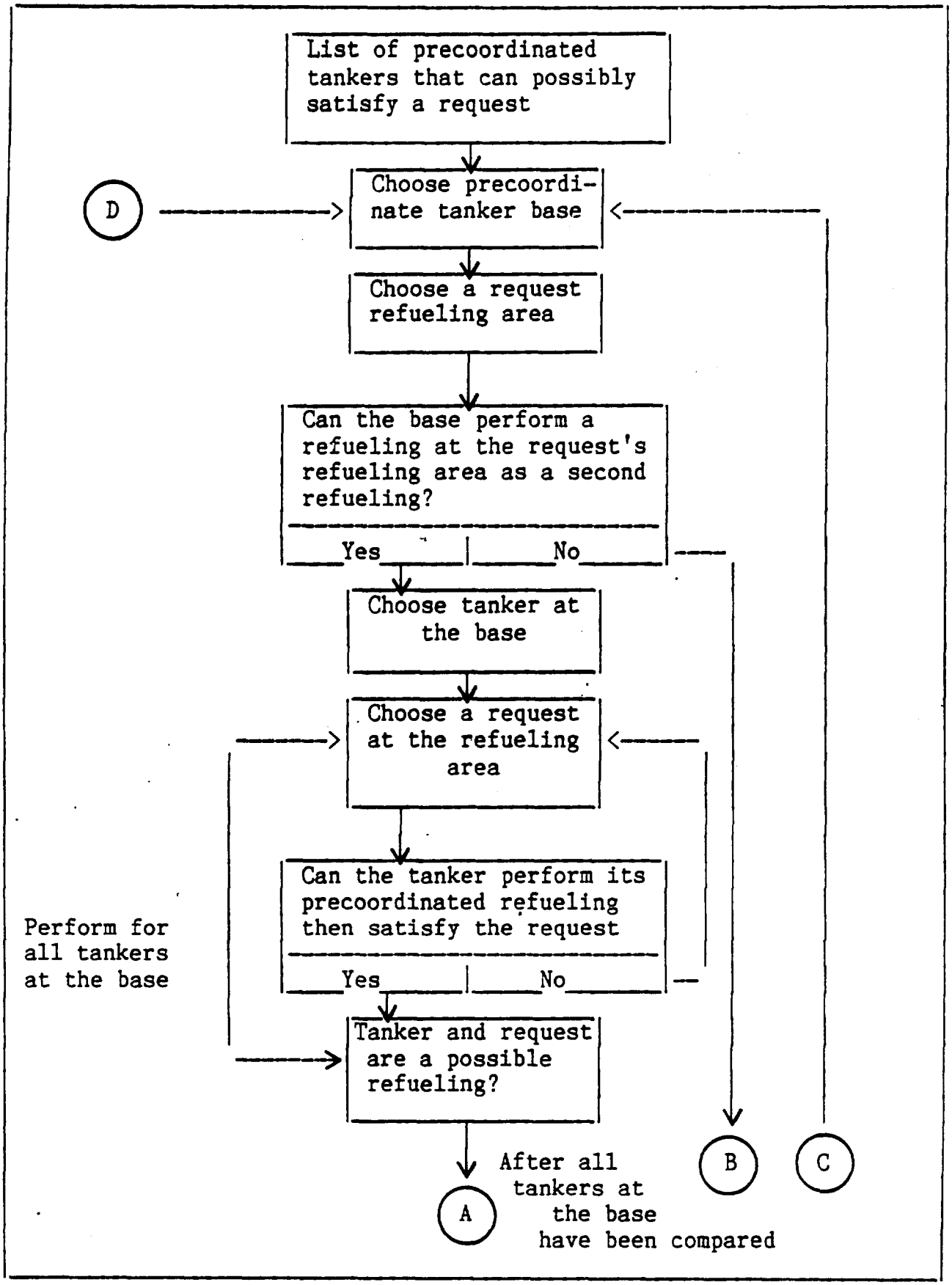


Figure 3.20a Flow diagram for a precoordinated tanker satisfying a request after its precoordinated ruefueling

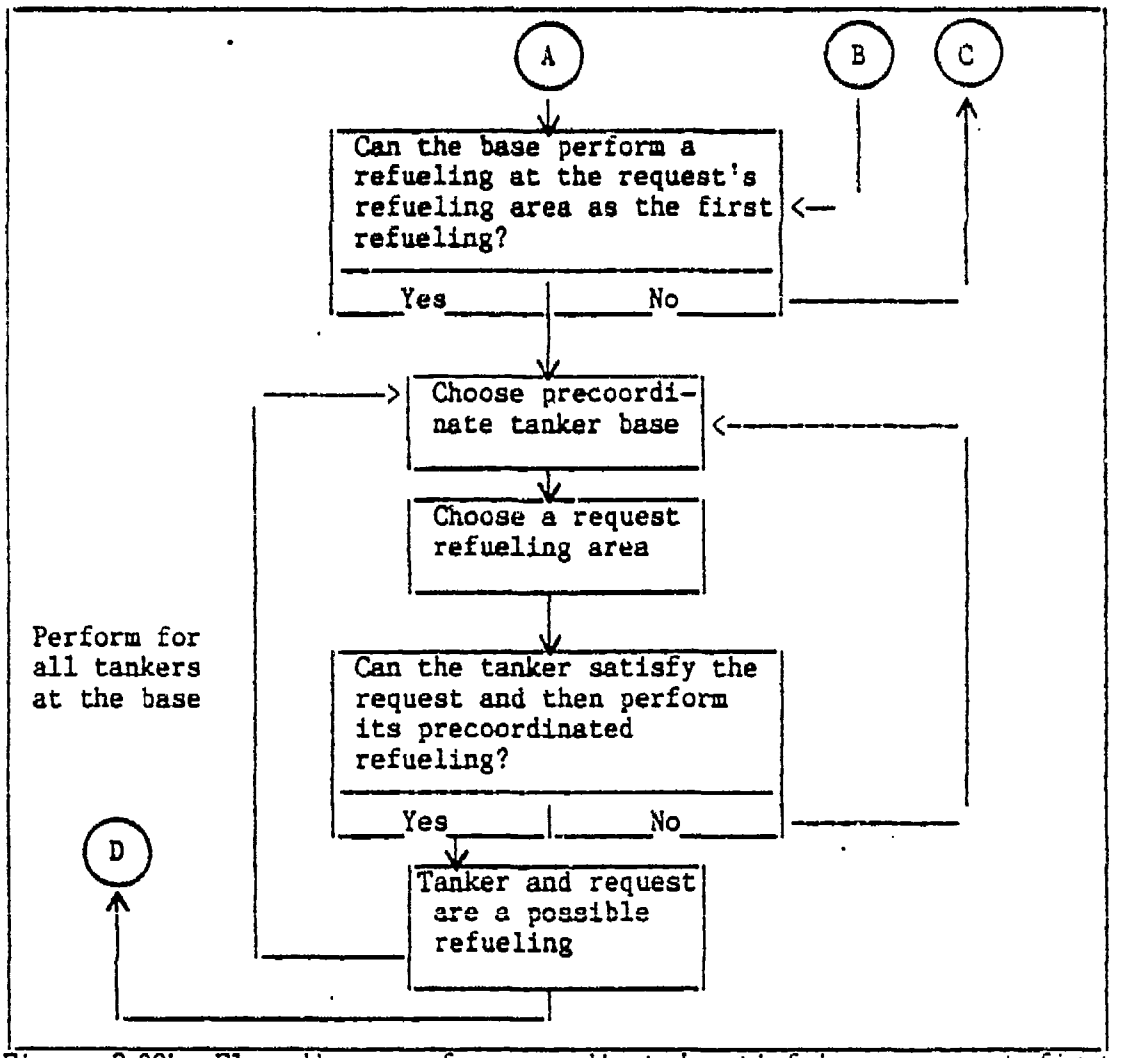


Figure 3.20b Flow diagram of precoordinated satisfying a request first and then performing its precoordinated refueling

Preprocessor Output. After the preprocessor determines and assigns variables to all of the possible refuelings it then creates its output files. One output file is created for each of the objective functions, one constraint file, and a file of the variable definitions. The preprocessor writes the applicable variables to each of the files.

Program Builder

A program builder is a software program that takes the objective

function and constraint files and combines them together to create the integer programming problem to be solved by the integer programming package. There is one program builder for each of the goals.

The first program builder program takes the first objective function file and combines it with the constraint file to form the IP problem. After the problem is built, it is solved by the IP package.

The second program builder accesses the solution file created by the IP package and reads the value of the first objective function. It then combines the second objective function file to the constraint file and includes the first objective function and its solution as a constraint. It is then solved by the IP package.

The third program builder accesses the solution file of the second programming problem and reads the value of the objective function. It takes the third objective function file and combines it with the constraint file. Two other constraints are include, the first objective function and its solution, and the second objective function and its solution. The IP package solves the problem and the results are the schedule.

System Description and Requirements

To perform the scheduling task the following hardware and software is used:

- Zenith Z-248 computer
  20 Megabyte hard disk drive
  640 K Random Access Memory (RAM)
- Mip83 Integer Program
- Compiled programming language

The use of an microcomputer to perform the scheduling task offers some advantages over a mainframe computer.

Advantages:

- Large availability of knowledge even among non-computer programmers.
- Don't have to worry about computer down time. If the computer is down another microcomputer can be used.
- Computer output can be mailed by diskette instead of sending a bulky report.
- Unclassified work can be performed at home on a compatible system.
- Microcomputer costs a lot less than a mainframe.
- Microcomputers are popular and readily available. By having common microcomputers at the tanker and receiver units a scheduling input program can be written and distributed to each unit to ensure that the scheduling inputs sent to HQ SAC/DO8A are in the proper format.

Disadvantage:

- Not as much memory space as a mainframe.
- Slower execution speed.
- May be too small to handle big problems.
- May not have the computer personnel support as is common with a mainframe. Programs may not have the advantage of being maintained by computer programmer.

IV. Results, Conclusions, and Recommendations

Results

A method was developed to formulate the scheduling problem as a generalized assignment problem. To formulate the generalized assignment problem, each request, available tanker, and precoordinated tanker, has a constraint associated with it. Each constraint contains all of the possible refuelings that can be performed with the request or tanker.

The scheduling objectives of maximizing the number of requests satisfied, maximizing the number of category B requests satisfied, and minimizing the total flight time can be achieved by using a preemptive goal programming approach to solving the problem. To do this, required each of the objectives to be prioritized.

Once the problem has been formulated as a generalized assignment problem, an integer programming package is used to perform the scheduling task. To employ the preemptive goal programming approach, three different executions, one for each objective, of the integer programming package are required to determine the final schedule.

A method was developed to preprocess the scheduling information received from the tanker and receiver units to assist the integer programming package. This preprocessor is used to cut down on the amount of processing to be performed by the integer programming package. Before the integer program is executed, the preprocessor determines all of the possible refuelings that can take place between the tankers and requests. By using a list of flight time constraints, all of the refuelings that are not possible are sifted out before they reach the

integer program. This sifting, allows the integer program to work with only those refuelings that are possible.

Conclusion

This thesis has demonstrated a way of scheduling a tanker to perform more than one refueling. It may not guarantee that all requests will be satisfied. However, it should at least decrease the number of unsatisfied requests.

Some important considerations are demonstrated in this thesis. A preemptive goal programming approach has added a new dimension to the scheduling process by allowing the scheduler to specify and rank order the objectives to be achieved when scheduling. As the objectives change, it should be easy to add, delete, or rearrange the objectives.

On the surface, the effort required to schedule tankers to perform two refuelings appears to be a difficult task. However, through the use of a program to determine all of the possible refuelings and an integer programming package the scheduling task is greatly simplified.

The handling of the scheduling process through the use of an off-the-shelf integer programming package and locally written software appear to present no problems. In fact, many advantages are gained through the use of an existing software package. An unforeseen advantage to be gained through the use of a commercial software program is the considerable decrease in the amount of computer code that has to be maintained. Rather than having to maintain an entire scheduling program, only the preprocessor and program builder need maintaining. Additions, deletions, and major overhauls should be easier. The preprocessor is primarily made up of the time and distance constraints

established by the schedulers. These constraints are no more than simple addition and subtraction problems. If no programmer support is available, much of the preprocessor program can be maintained by someone with a limited knowledge of programming.

Allowing the receiver units to submit their requests with an ARCT (air refueling control time) window rather than a fixed ARCT should increase the possibility of having an available tanker.

This scheduling method is not without problems. One assumption made is that all of the refuelings are not probe and drogue refuelings. A KC-10 may be able to perform a boom refueling and later a drogue refueling, while the KC-135 is limited to performing only one type of refueling without landing and changing refueling equipment. Before a plane is scheduled against two different types of refuelings, a check would have to be made to ensure that it is a KC-10 that is being scheduled.

The scheduling program does not account for the request's offload. The available offload of a KC-10 is almost three times greater than a KC-135. A check would need to be made on the tanker type before it is scheduled. To incorporate the offload as a constraint the type of tanker would have to be included in the tanker inputs. This constraint could be incorporated into portion of the preprocessor where the request pairs are identified. All request pairs with total offloads exceeding the capability of the KC-135 are only compared against the available KC-10s.

Recommendations

The system that is developed in this thesis is one that operates independent of any interaction with the scheduler. Once executed it performs the scheduling task. An upgrade to the scheduling process would be to allow the scheduler to interact with the computer as it solves the schedule. The system would allow the schedulers to target any tanker or request that requires special consideration when scheduling. If any portion of the scheduling solution needs changing, the scheduler could make those changes without exiting the program.

Another recommendation would be the actual implementation of the methodology described in this thesis on a computer. This would include computerizing the entire scheduling process from the point where the scheduling information has been received from the tanker and receiver units through the production of the completed schedule. To test the program, actual scheduling information from HQ SAC/DO8A should be used.

When tanker units submit their inputs, they may want certain conditions to apply when scheduling selected tankers. The current scheduling process assumes that a precoordinated tanker is available to satisfy a request either before or after its precoordinated refueling. However, this may not be the case. A precoordinated tanker may only be able to satisfy a request after its precoordinated refueling, or not at all. An available tanker may only be able to perform one refueling. The tanker unit may be willing to make a tanker available, but, they want the tanker back at the base at a specified time. This time may be well short of letting the tanker fly the maximum flight time.

Also, with different types of tankers, it may be desirable to have

different constraints on flight time to apply to each type of tanker. The tanker inputs would require specification of the tanker type.

Somehow, a convenient input system must be made available to the tanker units to specify their refueling conditions for each aircraft and the type of tanker. The key here is convenience, so that when the inputs are made, a lot of boxes will not have to be filled in.

Even if the goal of scheduling a tanker to perform more than one refueling is never realized, HQ SAC/DO8A can implement much of the methodology expressed in this thesis to their advantage. The use of a microcomputer, preprocessor, and integer programming package would release them from the computer system they are currently using. Also, the request format should be changed to allow receiver units to submit an ARCT window instead of a fixed ARCT. If the receiver units use the ARCT window, it will make a request more likely to be satisfied.

If a precoordinated tanker can perform two refuelings, it may be desirable to let the wing schedulers precoordinate tankers to perform more than one refueling.

Appendix A

MIP83 Output

The intent of this appendix is to demonstrate how MIP83 (Mixed Integer Programming) integer programming package is used to perform the scheduling task. A Zenith Z-248, IBM-compatible, computer is used to run the MIP83 program. To demonstrate the use of MIP83, the scheduling of the inputs in the toy problem are used. This chapter is not intended to be a substitute for the MIP83 user's manual (15), rather, it is an aid to better understand the use of MIP83 for the scheduling task.

All of the possible refuelings have been determined and assigned variables. The constraints and objective functions have been built. The variables for the tankers and receivers are listed with a brief explanation Figure A.1 and A.2.

Something unique to MIP83 is that all of the variables used in the the problem formulation must be listed in the objective function. Even if the variable is not part of the objective function it is listed and assigned a coefficient value of zero. Any variables in the objective function contained within the double brackets is a binary, zero or one, variable. The first objective to minimize the number of unscheduled requests (maximize scheduled requests). The formulation of the objective function is as follows:

Minimize Unscheduled Requests

[[R1 + R2 + R3 + R4 + R5 + R6 + R7 + R8 + R9 + R10 + R11 + R12 + R13 + R14 + 0 A1R3 + 0 A1R6 + 0 A1R6R16 + ... (the remaining variables that are in the constraints) + 0 PC4R15 + 0 PC5]]

A complete listing of the integer programming problem can be found

at the end of this appendix.

Available Tankers

Variable	Base	Earliest ARCT	Latest ARCT
A1	ALTUS	1700	2400
A2	ALTUS	1400	2000
B1	BARKSDALE	1600	2000
B2	BARKSDALE	1200	1600
B3	BARKSDALE	1800	2200
B4	BARKSDALE	0800	2400
C1	CARSWELL	1100	1400
C2	CARSWELL	2330	0730
C3	CARSWELL	1500	2000

Precoordinated Tankers

Variable	Precoord Base	Refuel Track	Refuel Time	ARCT Window
PA1	ALTUS	13E	0 30	0800-1000
PA2	ALTUS	330W	0 35	2100-2100
PA3	ALTUS	112W	0 30	1230-1430
PB4	BARKSDALE	101S	0 44	1500-1500
PB5	BARKSDALE	313N	0 20	2400-0200
PB3	BARKSDALE	112W	0 30	2400-0200
PC1	CARSWELL	104E	0 30	1210-1610
PC2	CARSWELL	102A	0 35	1100-1100
PC3	CARSWELL	13W	0 30	2300-0100
PC4	CARSWELL	112E	0 30	0500-0700
PC5	CARSWELL	650	2 30	1400-1600

Figure A.1 Tanker Units Scheduling Information

The first integer problem is solved using the following command to execute the MIP83 program:

```
C> MIP83 FIRSTOBJ  ALTERNATE 1 ACTIVITY  FIRSTOBJ.DAT COSTANALYSIS NO
   MARGINANALYSIS NO SOLUTION NO CONSTRAINT NO
```

The integer problem's file name is FIRSTOBJ and the answer is placed in file FIRSTOBJ.DAT. The use of this command to execute the MIP83 program causes the solutions to be saved to a file in a format that permits easy access by another program.

Refueling Requests

Variable	Refuel Track	Refuel Time	Number Tankers	Category	ARCT Window
R1	613	2 10	2	A	1300-1500
R2	104E	0 30	1	A	2300-0100
R3	650	3 00	1	B	0100-0300
R4	13E	0 30	1	A	1200-1200
R5	112W	0 30	1	A	1600-1800
R6	330W	0 35	1	A	1230-1430
R7	112E	0 30	2	B	1400-1600
R8	101N	0 55	1	A	1400-1600
R9	11BW	0 40	1	B	1200-1400
R1(	310W	0 15	1	A	2300-0100
R11	330E	0 30	1	B	1630-1830
R12	112W	0 30	1	A	1430-1630
R13	102A	0 35	1	B	0830-1030
R14	110E	0 30	1	A	1300-1500

If available tanker 1 from Altus is to satisfy request 4 the variable would be A1R4. Precoordinated tanker 3 from Barksdale is satisfy request 6 first and then perform its precoordinated refueling the variable assigned is R6PB3.

Figure A.2 Receiver units Scheduling Inputs

Upon solving the problem the solution file FIRSTOBJ appears as follows:

```
"  Minimize Unscheduled Requests "   <---------- Title
  36, 174,      2.0000     <---------- (number of constraints, number of
"R1       ",     1.0000,      1.0000      variables, value of objective
"R2       ",     1.0000,      1.0000      function)
"R3       ",     0.0000,      1.0000
"R4       ",     0.0000,      1.0000
"R5       ",     0.0000,      1.0000
"R6       ",     0.0000,      1.0000
"R7       ",     0.0000,      1.0000  <-- (variable name, value assign
"R8       ",     0.0000,      1.0000       to the variable, coefficient
"R9       ",     0.0000,      1.0000       in the objective function)
     .              .            .
     .              .            .
     .              .            .
```

The only thing of importance in the solution is the value of the objective function. For the first objective function the value is 2.0000. This 2.0000 value means that two of the requests will be

unsatisfied in the final schedule. The remaining information in the file is of no relevant importance at this time. From this solution the second objective function is formulated. The second objective function is to minimize the number of category B requests that are unsatisfied. The problem formulation is as follows:

Minimize Category "B" Refuelings

[[R3 + R7 + R9 + R11 + R13 + 0 R1 + 0 R2 + 0 R4 + 0 R5 + 0 R6 + 0 R8 + 0 R10 + 0 R12 + 0 R14 + ... (remaining variables that are found in the constraints) + ... + 0 PC4R13 + 0 PC4]]

Unassign: R1 + R2 + R3 + R4 + R5 + R6 + R7 + R8 + R9 + R10 + R11 + R12 + R13 + R14 = 2

Reqst 1: R1 = 1

Reqst 2: R2 = 1

Reqst 3: R3 + A1R3 + B4R3 + C2R3 + R3PB2 + A1R12R3 + B4R12R3 + PA2R3 + PC3R3 = 1

. . .

(All of the remaining constraints)

The formulation for the second integer problem is similar to the first. the only difference is the formulation of the objective function and the addition of an extra constraint. All of the other constraints are the same. The additional constraint in the second problem is the objective function of the first problem set equal to its computed value. The solution to the second problem is:

```
"  Minimize Category "B" Refuelings "
  36, 174,      0.0000
"R1       ",     1.0000,      0.0000
"R2       ",     1.0000,      0.0000
"R3       ",     0.0000,      0.0000
   .              .            .
   .              .            .
```

The value of the objective function is zero. A zero value means that the minimum number of category B requests that are unsatisfied is zero. In other words, all of the category B requests can be satisfied.

The final objective is to schedule the tankers and request in such a manner that it minimizes the total flight time. The objective functions of the first two goals are included as constraints and set equal to their respective values. The objective function is made up of the variables of the possible refuelings. The coefficient assigned to each variable is the amount of flight time required to perform the refueling. The problem appears as follows:

Minimize Flight time

[[0 R2 + 2.0333 A1R2 + 2.55 B4R2 + 1.7333 C2R2 + 3.8333 R2PB2 + 3.7 R2PC3 + 0 R3 + 6.0667 B4R3 + 5.2 C1R3 + 5.2 C3R3 + 0 R4 + 3.4667 B2R4 + 2.5667 C1R4 + 4.8 B2R4R7 + 4.8 B4R4R7 + 5.3333 B2R4R8 + 5.3333 B4R4R8 + 4.5333 C1R4R7 + 5.0833 C1R4R8 + 4.55 C1R4R12 + 3.2167 R4PA3 + 5.15 R4PB1 + 3.1667 R4PC1 + 0 R5 + 1.8333 A1R5 + 1.8333 A2R5 + 2.25 B1R5 + 2.25 B4R5 + 2.1833 C3R5 + 4.55 A1R5R14 + 4.55 A2R5R14 + 4.1333 B1R5R14 + 4.1333 B2R5R14 + 4.1333 B4R5R14 + 4.5 C3R5R14 + 0 R6 + 3.2667 B4R6 + 2.75 C1R6 + 4.4167 B4R6R4 + 4.7333 B4R6R7 + 4.5167 B2R6R12 + 4.5167 B4R6R12 + 4.2167 C1R6R7 + 4.2333 C1R6R12 + 3.4833 R6PA3 + 5.1 R6PB1 + 3.7333 R6PC1 + 2.45 A2R7 + 2.2167 B2R7 + 2.2167 B4R7 + 2.4 C1R7 + 3.7167 A2R7R5 + 4.85 A2R7R8 + 4.8667 A2R7R11 + 2.75 A2R7R12 + 4.25 B2R7R5 + 4.25 B4R7R5 + 4.4 B2R7R8 + 4.4 B4R7R8 + 4.9333 B2R7R11 + 4.9333 B4R7R11 + 3.2833 B2R7R12 + 3.2833 B4R7R12 + 3.9167 C1R7R5 + 4.7333 C1R7R8 + 4.9667 C1R7R11 + 2.95 C1R7R12 + 2.8667 R7PA3 + 0 R8 + 2.6667 B2R8 + 2.6667 B4R8 + 3.4167 C3R8 + 4.1333 A2R8R5 + 5.3167 A2R8R11 + 4.1333 A2R8R12 + 5.2667 A2R8R14 + 3.85 B1R8R5 + 3.85 B2R8R5 + 3.85 B4R8R5 + 4.6 B2R8R11 + 4.6 B4R8R11 + 3.8833 B2R8R12 + 3.8833 B4R8R12 + 4.1833 B2R8R14 + 4.1833 B4R8R14 + 4.0167 C3R8R5 + 4.0167 C1R8R12 + 4.9167 C3R8R14 + 0 R9 + 5.3167 A2R9R5 + 5.3167 R9PA3 +0 R10 + 2.9833 A1R10 + 4.0833 A1R10R2 + 4.1167 B4R10R2 + 4.65 R10PB2 + 0 R11 + 2.35 A2R11 + 2.9 B2R11 + 2.9 B4R11 + 2.6833 C3R11 + 3.8833 A2R11R14 + 4.0333 B1R11R14 + 4.0333 B4R11R14 + 0 R12 + 1.8333 A2R12 + 2.25 B2R12 + 2.25 B4R12 + 2.1833 C3R12 + 2.9333 A2R12R5 + 3.6333 A2R12R11 + 4.3 A2R12R14 + 3.5167 B1R12R5 + 3.5167 B2R12R5 + 3.5167 B4R12R5 + 3.5833 B2R12R11 + 3.5833 B4R12R11 + 4.1333 B1R12R14 + 4.1333 B2R12R14 + 4.1333 B4R12R14 + 0 R13 + 1.7833 B4R13 + 4.95 B4R13R4 + 4.5667 B4R13R6 + 4.7167 R13PA3 + 3.7 R13PB3 + 3.6 R13PC1 + 0 R14 + 3.2833 A1R14 + 3.2833 A2R14 + 2.8667 B1R14 + 2.8667 B3R14 + 2.8667 B4R14 + 3.3167 C3R14 + 4.45 R14PA2 + 1.9167 PA1 + 3.7667

PA1R4 + 3.5833 PA1R6 + 3.4667 PA1R13 + 1.9833 PA2 + 4.1833 PA2R2 + 4.6667 PA2R10 + 1.5833 PA3 + 3.0833 PA3R5 + 5.1667 PA3R8 + 3.6333 PA3R11 + 3.1 PA3R12 + 4.55 PA3R14 + 2.3167 PB1 + 4.45 PB1R5 + 3.9 PB1R8 + 4.5 PB1R12 + 1.7167 PB2 + 2.0 PB3 + 4.7167 PB3R4 + 3.9167 PB3R6 + 2.65 PB3R13 + 1.4833 PC1 + 3.7833 PC1R5 + 3.6833 PC1R7 + 4.65 PC1R8 + 4.6667 PC1R11 + 3.3833 PC1R12 + 1.9333 PC2 + 4.0833 PC2R6 + 3.5167 PC2R13 +2.1833 PC3 + 3.0167 PC3R2 + 3.9167 PC3R10 + 2.15 PC4 + 3.8167 PC4R13 + 4.45 PC5]] + [0 R1 + 0 R7]

Unassign: R1 + R2 + R3 + R4 + R5 + R6 + R7 + R8 + R9 + R10 + R11 + R12 + R13 + R14 = 2

Categ B : 1 R3 + 1 R7 + 1 R9 + 1 R11 + 1 R13 = 0

Reqst 1: R1 = 2

Reqst 2: R2 + A1R2 + B4R2 + C2R2 + R2PB2 + R2PC3 + A1R10R2 + B4R10R2 + PA2R2 + PC3R2 = 1

Reqst 3: R3 + B4R3 + C1R3 + C3R3 = 1

. .
. .
. .

(All of the remaining constraints)

When this integer problem is solved, the final solution is the schedule. Not only is the value of the objective important, but also the value of each of the variables. If a variable is assigned a value of one, the refueling is scheduled. A value of zero indicates it is not scheduled. The value of the objective function is the total amount of flight time to accomplish the refuelings.

The final solution is found in figure A.3a and A.3b. As part of the solution the number of constraints in the problem is listed. The last 38 entries in the solution represent the constraints. Listed is the label of the constraint, the actual value of the constraint in the final solution, and the value assigned the constraint when it was entered into the integer program. Notice (figure A.1b) that for Altus A1, Barks B1, Barks B2, and Barks B3 are assigned a value of zero in the

final solution. Each one of these labels represents a constraint for an available tanker in the integer problem. By being assigned a value of zero the tanker is unscheduled.

The tankers and requests are scheduled as follows:

Unsatisfied Requests: R1, R2

Scheduled Refueling:	Variable Identifier	Flight time
	C2R3	1.7333
	C3R4	5.2000
	R5PC1	3.1667
	R8PA3	2.8667
	C1R9R14	2.9500
	A2R11R6	5.3167
	B4R13R16	4.0333
	PA1R7	3.5833
	PA2	1.9833
	PB1R10	3.9000
	PB2	1.7167
	PB3R15	2.6500
	PC2	1.9333
	PC3R12	3.9167
	PC4	2.1500
	PC5	4.4500
	Total Flight Time:	51.55 hours

Available tankers that were not scheduled: A1, B1, B2, B3

" Minimize Flight time "			"B2R5R14 ",	0.0000,	4.1333	"C1R7R12 ",	1.0000,	2.9500	"C3R11 ",	0.0000,	2.6833
36, 174,	51.5500		"B4R5R14 ",	0.0000,	4.1333	"R7PA3 ",	1.0000,	2.8667	"A2R11R14",	0.0000,	3.8833
"R2 ",	0.0000,	0.0000	"C3R5R14 ",	0.0000,	4.5000	"R8 ",	0.0000,	0.0000	"B1R11R14",	1.0000,	4.0333
"A1R2 ",	0.0000,	2.0833	"R6 ",	0.0000,	0.0000	"B2R8 ",	0.0000,	2.6667	"B4R11R14",	0.0000,	4.0333
"B4R2 ",	0.0000,	2.5500	"B4R6 ",	0.0000,	3.2667	"B4R8 ",	0.0000,	2.6667	"R12 ",	0.0000,	0.0000
"C2R2 ",	1.0000,	1.7333	"C1R6 ",	0.0000,	2.7500	"C3R8 ",	0.0000,	3.4167	"A2R12 ",	0.0000,	1.8333
"R2PB2 ",	0.0000,	3.8333	"B4R6R4 ",	0.0000,	4.4167	"A2R8R5 ",	0.0000,	4.1333	"B2R12 ",	0.0000,	2.2500
"R2PC3 ",	0.0000,	3.7000	"B4R6R7 ",	0.0000,	4.7333	"A2R8R11 ",	0.0000,	5.3167	"B4R12 ",	0.0000,	2.2500
"R3 ",	0.0000,	0.0000	"B2R6R12 ",	0.0000,	4.5167	"A2R8R12 ",	0.0000,	4.1333	"C3R12 ",	0.0000,	2.1833
"B4R3 ",	0.0000,	6.0667	"B4R6R12 ",	0.0000,	4.5167	"A2R8R14 ",	0.0000,	5.2667	"A2R12R5 ",	0.0000,	2.9333
"C1R3 ",	0.0000,	5.2000	"C1R6R7 ",	0.0000,	4.2167	"B1R8R5 ",	0.0000,	3.8500	"A2R12R11",	0.0000,	3.6333
"C3R3 ",	1.0000,	5.2000	"C1R6R12 ",	0.0000,	4.2333	"B2R8R5 ",	0.0000,	3.8500	"A2R12R14",	0.0000,	4.3000
"R4 ",	0.0000,	0.0000	"R6PA3 ",	0.0000,	3.4833	"B4R8R5 ",	0.0000,	3.8500	"B1R12R5 ",	0.0000,	3.5167
"B2R4 ",	0.0000,	3.4667	"R6PB1 ",	0.0000,	5.1000	"B2R8R11 ",	0.0000,	4.6000	"B2R12R5 ",	0.0000,	3.5167
"C1R4 ",	0.0000,	2.5667	"R6PC1 ",	0.0000,	3.7333	"B4R8R11 ",	0.0000,	4.6000	"B4R12R5 ",	0.0000,	3.5167
"B2R4R7 ",	0.0000,	4.8000	"A2R7 ",	0.0000,	2.4500	"B2R8R12 ",	0.0000,	3.8833	"B2R12R11",	0.0000,	3.5833
"B4R4R7 ",	0.0000,	4.8000	"B2R7 ",	0.0000,	2.2167	"B4R8R12 ",	0.0000,	3.8833	"B4R12R11",	0.0000,	3.5833
"B2R4R8 ",	0.0000,	5.3333	"B4R7 ",	0.0000,	2.2167	"B2R8R14 ",	0.0000,	4.1833	"B1R12R14",	0.0000,	4.1333
"B4R4R8 ",	0.0000,	5.3333	"C1R7 ",	0.0000,	2.4000	"B4R8R14 ",	0.0000,	4.1833	"B2R12R14",	0.0000,	4.1333
"C1R4R7 ",	0.0000,	4.5333	"A2R7R5 ",	0.0000,	3.7167	"C3R8R5 ",	0.0000,	4.0167	"B4R12R14",	0.0000,	4.1333
"C1R4R8 ",	0.0000,	5.0833	"A2R7R8 ",	0.0000,	4.8500	"C1R8R12 ",	0.0000,	4.0167	"R13 ",	0.0000,	0.0000
"C1R4R12 ",	0.0000,	4.5500	"A2R7R11 ",	0.0000,	4.8667	"C3R8R14 ",	0.0000,	4.9167	"B4R13 ",	0.0000,	1.7833
"R4PA3 ",	0.0000,	3.2167	"A2R7R12 ",	0.0000,	2.7500	"R9 ",	0.0000,	0.0000	"B4R13R4 ",	0.0000,	4.9500
"R4PB1 ",	0.0000,	5.1500	"B2R7R5 ",	0.0000,	4.2500	"A2R9R5 ",	1.0000,	5.3167	"B4R13R6 ",	0.0000,	4.5667
"R4PC1 ",	1.0000,	3.1667	"B4R7R5 ",	0.0000,	4.2500	"R9PA3 ",	0.0000,	5.3167	"R13PA3 ",	0.0000,	4.7167
"R5 ",	0.0000,	0.0000	"B2R7R8 ",	0.0000,	4.4000	"R10 ",	0.0000,	0.0000	"R13PB3 ",	0.0000,	3.7000
"A1R5 ",	0.0000,	1.8333	"B4R7R8 ",	0.0000,	4.4000	"A1R10 ",	0.0000,	2.9833	"R13PC1 ",	0.0000,	3.6000
"A2R5 ",	0.0000,	1.8333	"B2R7R11 ",	0.0000,	4.9333	"A1R10R2 ",	0.0000,	4.0833	"R14 ",	0.0000,	0.0000
"B1R5 ",	0.0000,	2.2500	"B4R7R11 ",	0.0000,	4.9333	"B4R10R2 ",	0.0000,	4.1167	"A1R14 ",	0.0000,	3.2833
"B4R5 ",	0.0000,	2.2500	"B2R7R12 ",	0.0000,	3.2833	"R10PB2 ",	0.0000,	4.6500	"A2R14 ",	0.0000,	3.2833
"C3R5 ",	0.0000,	2.1833	"B4R7R12 ",	0.0000,	3.2833	"R11 ",	0.0000,	0.0000	"B1R14 ",	0.0000,	2.8667
"A1R5R14 ",	0.0000,	4.5500	"C1R7R5 ",	0.0000,	3.9167	"A2R11 ",	0.0000,	2.3500	"B3R14 ",	0.0000,	2.8667
"A2R5R14 ",	0.0000,	4.5500	"C1R7R8 ",	0.0000,	4.7333	"B2R11 ",	0.0000,	2.9000	"B4R14 ",	0.0000,	2.8667
"B1R5R14 ",	0.0000,	4.1333	"C1R7R11 ",	0.0000,	4.9667	"B4R11 ",	0.0000,	2.9000	"C3R14 ",	0.0000,	3.3167

Figure A.3a Scheduling Output file

"R14PA2 ", 0.0000, 4.4500	"PC3R10 ", 1.0000, 3.9167	"Precrd B", 1.0000, 1.0000
"PA1 ", 0.0000, 1.9167	"PC4 ", 1.0000, 2.1500	"Precrd B", 1.0000, 1.0000
"PA1R4 ", 0.0000, 3.7667	"PC4R13 ", 0.0000, 3.8167	"Precrd B", 1.0000, 1.0000
"PA1R6 ", 1.0000, 3.5833	"PC5 ", 1.0000, 4.4500	"Precrd C", 1.0000, 1.0000
"PA1R13 ", 0.0000, 3.4667	"R1 ", 2.0000, 0.0000	"Precrd C", 1.0000, 1.0000
"PA2 ", 1.0000, 1.9833	"R7 ", 0.0000, 0.0000	"Precrd C", 1.0000, 1.0000
"PA2R2 ", 0.0000, 4.1833	"Unassign", 2.0000, 2.0000	"Precrd C", 1.0000, 1.0000
"PA2R10 ", 0.0000, 4.6667	"Categ B ", 0.0000, 0.0000	"Precrd C", 1.0000, 1.0000
"PA3 ", 0.0000, 1.5833	"Reqst 1 ", 2.0000, 2.0000	
"PA3R5 ", 0.0000, 3.0833	"Reqst 2 ", 1.0000, 1.0000	
"PA3R8 ", 0.0000, 5.1667	"Reqst 3 ", 1.0000, 1.0000	
"PA3R11 ", 0.0000, 3.6333	"Reqst 4 ", 1.0000, 1.0000	
"PA3R12 ", 0.0000, 3.1000	"Reqst 5 ", 1.0000, 1.0000	
"PA3R14 ", 0.0000, 4.5500	"Reqst 6 ", 1.0000, 1.0000	
"PB1 ", 0.0000, 2.3167	"Reqst 7 ", 2.0000, 2.0000	
"PB1R5 ", 0.0000, 4.4500	"Reqst 8 ", 1.0000, 1.0000	
"PB1R8 ", 1.0000, 3.9000	"Reqst 9 ", 1.0000, 1.0000	
"PB1R12 ", 0.0000, 4.5000	"Reqst 10", 1.0000, 1.0000	
"PB2 ", 1.0000, 1.7167	"Reqst 11", 1.0000, 1.0000	
"PB3 ", 0.0000, 2.0000	"Reqst 12", 1.0000, 1.0000	
"PB3R4 ", 0.0000, 4.7167	"Reqst 13", 1.0000, 1.0000	
"PB3R6 ", 0.0000, 3.9167	"Reqst 14", 1.0000, 1.0000	
"PB3R13 ", 1.0000, 2.6500	"Altus A1", 0.0000, 1.0000	
"PC1 ", 0.0000, 1.4833	"Altus A2", 1.0000, 1.0000	
"PC1R5 ", 0.0000, 3.7833	"Barks B1", 1.0000, 1.0000	
"PC1R7 ", 0.0000, 3.6833	"Barks B2", 0.0000, 1.0000	
"PC1R8 ", 0.0000, 4.6500	"Barks B3", 0.0000, 1.0000	
"PC1R11 ", 0.0000, 4.6667	"Barks B4", 0.0000, 1.0000	
"PC1R12 ", 0.0000, 3.3833	"Carsw C1", 1.0000, 1.0000	
"PC2 ", 1.0000, 1.9333	"Carsw C2", 1.0000, 1.0000	
"PC2R6 ", 0.0000, 4.0833	"Carsw C3", 1.0000, 1.0000	
"PC2R13 ", 0.0000, 3.5167	"Precrd A", 1.0000, 1.0000	
"PC3 ", 0.0000, 2.1833	"Precrd A", 1.0000, 1.0000	
"PC3R2 ", 0.0000, 3.0167	"Precrd A", 1.0000, 1.0000	

Figure A.3b Scheduling Output file

This is the entire integer programming problem for the third objective, to minimize the total flight time:

Reqst 4: R4 + B2R4 + C1R4 + B2R4R7 + B4R4R7 + B2R4R8 + B4R4R8 + C1R4R7 + C1R4R8 + C1R4R12 + R4PA3 + R4PB1 + R4PC1 + B4R6R4 + B4R13R4 + PA1R4 + PB3R4 = 1

Reqst 5: R5 + A1R5 + A2R5 + B1R5 + B4R5 + C3R5 + A1R5R14 + A2R5R14 + B1R5R14 + B2R5R14 + B4R5R14 + C3R5R14 + A2R7R5 + A2R8R5 + A2R9R5 + A2R12R5 + B2R7R5 + B4R7R5 + B1R8R5 + B2R8R5 + B4R8R5 + B1R12R5 + B2R12R5 + B4R12R5 + C1R7R5 + C3R8R5 + PA3R5 + PB1R5 + PC1R5 = 1

Reqst 6: R6 + B4R6 + C1R6 + B4R6R4 + B4R6R7 + B2R6R12 + B4R6R12 + C1R6R7 + C1R6R12 + R6PA3 + R6PB1 + R6PC1 + B4R13R6 + PA1R6 + PB3R6 + PC2R6 = 1

Reqst 7: R7 + A2R7 + B2R7 + B4R7 + C1R7 + A2R7R5 + A2R7R8 + A2R7R11 + A2R7R12 + B2R7R5 + B4R7R5 + B2R7R8 + B4R7R8 + B2R7R11 + B4R7R11 + B2R7R12 + B4R7R12 + C1R7R5 + C1R7R8 + C1R7R11 + C1R7R12 + R7PA3 + B2R4R7 + B4R4R7 + B4R6R7 + C1R4R7 + C1R6R7 + PC1R7 = 2

Reqst 8: R8 + B2R8 + B4R8 + C3R8 + A2R8R5 + A2R8R11 + A2R8R12 + A2R8R14 + B1R8R5 + B2R8R5 + B4R8R5 + B2R8R11 + B4R8R11 + B2R8R12 + B4R8R12 + B2R8R14 + B4R8R14 + C3R8R5 + C1R8R12 + C3R8R14 + A2R7R8 + B2R4R8 + B4R4R8 + B2R7R8 + B4R7R8 + C1R4R8 + C1R7R8 + PA3R8 + PB1R8 + PC1R8 = 1

Reqst 9: R9 + A2R9R5 + R9PA3 = 1

Reqst 10: R10 + A1R10 + A1R10R2 + B4R10R2 + R10PB2 + PA2R10 + PC3R10 = 1

Reqst 11: R11 + A2R11 + B2R11 + B4R11 + C3R11 + A2R11R14 + B1R11R14 + B4R11R14 + A2R7R11 + A2R8R11 + A2R12R11 + B2R7R11 + B4R7R11 + B2R8R11 + B4R8R11 + B2R12R11 + B4R12R11 + C1R7R11 + PA3R11 + PC1R11 = 1

Reqst 12: R12 + A2R12 + B2R12 + B4R12 + C3R12 + A2R12R5 + A2R12R11 + A2R12R14 + B1R12R5 + B2R12R5 + B4R12R5 + B2R12R11 + B4R12R11 + B1R12R14 + B2R12R14 + B4R12R14 + A2R7R12 + A2R8R12 + B2R6R12 + B4R6R12 + B2R7R12 + B4R7R12 + B2R8R12 + B4R8R12 + C1R4R12 + C1R6R12 + C1R7R12 + C1R8R12 + PA3R12 + PB1R12 + PC1R12 = 1

Reqst 13: R13 + B4R13 + B4R13R4 + B4R13R6 + R13PA3 + R13PB3 + R13PC1 + PA1R13 + PB3R13 + PC2R13 + PC4R13 = 1

```
Reqst 14: R14 + A1R14 + A2R14 + B1R14 + B3R14 + B4R14 + C3R14 + R14PA2 +
          A1R5R14 + A2R5R14 + A2R8R14 + A2R11R14 + A2R12R14 + B1R5R14 +
          B2R5R14 + B4R5R14 + B2R8R14 + B4R8R14 + B1R11R14 + B4R11R14 +
          B1R12R14 + B2R12R14 + B4R12R14 + C3R5R14 + C3R8R14 + PA3R14
          = 1

Altus A1: A1R2 + A1R5 + A1R10 + A1R10R2 + A1R14 + A1R5R14  <=  1

Altus A2: A2R5 + A2R5R14 + A2R7 + A2R7R5 + A2R7R8 + A2R7R11 + A2R7R12 +
          A2R8R5 + A2R8R11 + A2R8R12 + A2R8R14 + A2R9R5 + A2R11 +
          A2R11R14 + A2R12R5 + A2R12R11 + A2R12R14 + A2R14 <=  1

Barks B1: B1R5 + B1R5R14 + B1R8R5 + B1R11R14 + B1R12R14 + B1R14  <=  1

Barks B2: B2R4 + B2R4R7 + B2R4R8 + B2R5R14 + B2R6R12 + B2R7 + B2R7R5 +
          B2R7R8 + B2R7R11 + B2R7R12 + B2R8 + B2R8R5 + B2R8R11 + B2R8R12
          +B2R8R14 + B2R11 + B2R12 + B2R12R5 + B2R12R11 + B2R12R14<= 1

Barks B3: B3R14  <=  1

Barks B4: B4R2 + B4R3 + B4R4R7 + B4R4R8 + B4R5 + B4R5R14 + B4R6 + B4R6R4
          + B4R6R7 + B4R6R12 + B4R7 + B4R7R5 + B4R7R8 + B4R7R11 +
          B4R7R12 B4R8 + B4R8R5 + B4R8R11 + B4R8R12 + B4R8R14 + B4R10R2
          + B4R11 + B4R11R14 + B4R12 + B4R12R5 + B4R12R11 + B4R12R14 +
          B4R13 + B4R13R4 + B4R13R6 + B4R14  <=  1

Carsw C1: C1R3 + C1R4 + C1R4R7 + C1R4R8 + C1R4R12 + C1R6 + C1R6R7 +
          C1R6R12 + C1R7 + C1R7R5 + C1R7R8 + C1R7R11 + C1R7R12 + C1R8R12
          <=  1

Carsw C2: C2R2  <=  1

Carsw C3: C3R3 + C3R5 + C3R5R14 + C3R8 + C3R8R5 + C3R8R14 + C3R11 +
          C3R12 + C3R14  <=  1

Precrd A1: PA1 + PA1R4 + PA1R6 + PA1R13  =  1

Precrd A2: PA2 + PA2R2 + PA2R10 + R14PA2  =  1

Precrd A3: PA3 + PA3R5 + PA3R8 + PA3R11 + PA3R12 + PA3R14 + R4PA3 +
           R6PA3 + R7PA3 + R9PA3 + R13PA3  =  1

Precrd B1: PB1 + PB1R5 + PB1R8 + PB1R12 + R4PB1 + R6PB1 = 1

Precrd B2: PB2 + R2PB2 + R10PB2  =  1

Precrd B3: PB3 + PB3R4 + PB3R6 + PB3R13 + R13PB3  =  1

Precrd C1: PC1 + PC1R5 + PC1R7 + PC1R8 + PC1R11 + PC1R12 + R4PC1 + R6PC1
           + R13PC1  =  1

Precrd C2: PC2 + PC2R6 + PC2R13  =  1
```

Precrd C3: PC3 + PC3R2 + PC3R10 + R2PC3 = 1

Precrd C4: PC4 + PC4R13 = 1

Precrd C5: PC5 = 1

Appendix B

Preprocessor Algorithm

Presented in this appendix is an algorithm for developing the preprocessor.

Available Tanker Satisfying One Request

This algorithm determines the possible refuelings for an available tanker satisfying a single request.

1. Take first available tanker's base.
2. Go to step 4.
3. Take next available tanker's base. If there are no more bases STOP.
4. Take first request's refueling area.
5. Go to step 7.
6. Take next request's refueling area. If there are no more refueling areas go to step 3.
7. Can base perform a refueling at the refueling area?

Flight time from the base to ARCP	$\leq$	Maximum allowable flight time to the ARCP
Flight time from area's end refueling point to the base	$\leq$	Maximum allowable flight time from the end refueling point to the base
Flight time from the base to the ARCP plus flight time from area's end refueling point to base	$\leq$	Maximum allowable flight time from the sum of the flight times to the ARCP plus the flight time to return from the end refueling point

If any of these conditions is not met, the tanker's at the base cannot perform a refueling at the request's area, go to step 6. If all the conditions are met, compare each tanker at the base to each request at the refueling area.

8. Take first available tanker at the base.
9. Go to step 11.

10. Take next available tanker. If there are no more tankers return to step 6.

11. Take first request at the area. go to step 13.

12. Take next request. If there are no more requests, return to step 10.

13. Can tanker satisfy the request?

    Available tanker's earliest ARCT $\leq$ Request's latest ARCT

    Available tanker's latest ARCT $\geq$ Request's earliest ARCT

    Total mission length $\leq$ Maximum allowable mission length

    If any of the constraints are not met, the tanker cannot satisfy the request, go to step 12.

    If all constraints are met, the tanker and request are a possible refueling. Go to step 12.

Available Tanker Satisfying Two Requests

This algorithm determines all the requests that can be paired together for later use in determining the possible refuelings for an available tanker satisfying two requests.

Reference is made to a first list and a second list. Both lists are identical copies of the request's file.

1. Take first refueling area in first list.

2. Go to step 4.

3. Take next refueling area in first list. If there are no more areas, go to step 17.

4. Take first refueling area in second list.

5. Go to step 7.

6. Take next refueling area in second list. If there are no more areas in the second list, go to step 3.

7. Can a refueling be performed at area and then at area 2?

    Flight time from the end refueling point of the first area to the ARCP of the second refueling area $-$ Maximum allowable flight time between refueling area

If the refueling areas satisfy the constraint than continue. If not, go to step 6.

8. Take first request from the first refueling area.

9. Go to step 11.

10. Take next request from the first refueling area. If there are no more requests, go to step 6.

11. Take first request from the second refueling area.

12. Go to step 14.

13. Take next request from the second refueling area. If there are no more request, go to step 10.

14. Compare air refueling control times.

(first request's earliest ARCT) + (first request's refueling time) + (flight time to second request's ARCP) + (orbit time) $\leq$ Second requests latest ARCP

(first request's latest ARCT) + (first request's refueling time) + (maximum allowable flight time between refueling areas) + (orbit time) $\geq$ Second request's earliest ARCT

If the constraints are satisfied, than the requests are a request pair. If either of these constraints are not met, go to step 13.

15. Determine new ARCT windows for the first request.

Earliest Possible ARCT for the first request = second request's earliest ARCT + [(orbit time at second ARCP) + (maximum allowable flight time between refueling areas) + (first request's refueling time)]

If the first request's earliest ARCT is earlier than the computed earliest possible ARCT, the request's earliest ARCT is set equal to the earliest possible ARCT. If it is not earlier, it is left unchanged.

Latest possible ARCT for the first request = Second request's latest ARCT - [(orbit time at second ARCP) + (flight time from first request's end refueling point to second request's ARCP) + (first request's refueling time)]

If the first request's latest ARCT is later than the latest possible ARCT, the request's latest ARCT is set equal to the latest possible ARCT. If it is not later, it is left unchanged.

These newly computed ARCTs for the first request only apply to the request pair. If the same request is compared to another request the original ARCTs are used.

16. Go to step 13.

17. Go through the list of all of the possible refueling pairs and identify any pairs in which the requests in both pair are identical but in different order (e.g.,. pair (1,2) and pair (2,1)). Record each of the requests in the pairs.

18. STOP

This algorithm determines the refuelings in which an available tanker can satisfy two requests. This is done by comparing the available tankers to the computed request pairs.

1. Take first available tanker's base.

2. Go to step 5.

3. Take next available tanker's base. If there are no more tanker bases, go to 16

4. Take first first request pair.

5. Go to step 7.

6. Take next request pair having a different refueling area for the first request than the previous request pair. If there are no more request pairs, go to step 3.

7. Can the base perform a refueling at the refueling area of the first request in the pair?

   flight time from base to the ARCP of the first refueling area $\leq$ maximum allowable flight time from the base to the ARCP

   If the base satisfies the constraint, then, the base will be compared to each of the request pairs having the same first refueling area. If the constraint is not satisfied, go to step 6.

8. Go to step 10.

9. Take next request pair that has the same refueling area for the first request as the previous request pair. If there are no other request pairs having the same refueling for the first request, go to step 6.

10. Can the base perform a refueling at both refueling area for the first and second request in the request pair?

flight time from second end refueling point to the base $\leq$ maximum allowable time from the end refueling point to the base

total non-refueling time $\leq$ maximum allowable non-refueling time

total mission length $\leq$ maximum mission length

If the base satisfies these constraints, then each of the tankers at the base are compared to the refueling pair. If the base does not satisfy the constraint, go to step 9.

11. Take the first available tanker at the base.

12. Go to step 14.

13. Take next available tanker at the base. If there are no more tankers at the base, go to step 9.

14. Compare ARCTs between the tanker and the first request.

available tanker's earliest ARCT $\leq$ request's latest ARCT
available tanker's latest ARCT $\geq$ request's earliest ARCT

If the constraints are satisfied, than, the tanker and the request pair are a possible refueling.

15. Go to step 13.

16. Take the list of request pairs, identified in the algorithm for finding request pairs, having the same requests in the pair but in different order. Determine if there are any available tankers that a possible refueling has been determined for one of these request pairs and its opposite request pair. If the case exits determine which possible refueling has the lowest total mission flight time and discard the other possible refueling.

17. STOP

Precoordinated Tanker Performing Its Precoordinated Refueling and Satisfying a Request

Before the precoordinated tankers are compared to the request the first thin to be determined is all of the precoordinated tankers that can not satisfy a request.

This algorithm determines all of the precoordinated tankers that can perform their precoordinated refueling and satisfy a request.

1. Take first precoordinated tanker.

2. Go to step 4.

3. Take next precoordinated tanker. If there are no more precoordinated tankers, go to step 6.

4. Compare flight time to and from precoordinated area.

flight time from base to precoordinated refueling area	$\leq$	maximum allowable flight time from base to ARCP
flight time from precoordinate's end refueling point to base	$\leq$	maximum allowable flight time from end refueling point to the base

The maximum allowable flight times are those for a double refueling.

If either of these constraints are not adhered to, the precoordinated tanker can only perform its precoordinated refueling. If it does meet the constraints, then it is added to a list of precoordinated tankers that can satisfy a request. It is this list that will be used to compared with the list of requests.

5. Go to step 3.

6. STOP.

This algorithm computes all of the possible refueling combinations of a precoordinated tanker performing its precoordinated refueling and then satisfying a request. When the term "tanker" is used, it refers to the precoordinated tanker and its precoordinated refueling.

1. Take first precoordinate tanker's base.

2. Go to step 4.

3. Take next precoordinate's base. If there are no other precoordinate tanker bases, go to step 28.

4. Take first refueling area.

5. Go to step 7.

6. Take next refueling area. If there are no other refueling areas, go to step 3.

7. Set precoord-first-flag to "on."

8. Can base perform a refueling at the request's refueling area?

flight time from the request's refueling area to the base	$\leq$	maximum allowable flight time from the end refueling point to the base

If this constraint is satisfied then each tanker at the precoordinated base is compared to each request for the area. If the constraint is not satisfied, set precoord-first-flag to "off", and go to step 17.

9. Take first tanker at the base.

10. Go to step 12.

11. Take next precoordinated tanker at the base. If there are no other tankers at the base, go to step 17.

12. Take first request at the refueling area.

13. Go to step 15.

14. Take next request at the refueling area. If there are no more requests at the area, go to step 11.

15. Can the tanker perform its precoordinated refueling first and then satisfy the request?

a. ARCT compatibility between the tanker and the request?

(tanker's earliest ARCT) + (tanker's refueling time) + (flight time to request's ARCP) + (orbit time) $\leq$ request's latest ARCT

(tanker's latest ARCT) + (tanker's refueling time) + (maximum allowable flight time between refueling areas) + (orbit time) $\geq$ request's earliest ARCT

If both constraints are not satisfied, go to step 14.

b. Any flight time limitations exceeded?

flight time from the end refueling point of the tanker's area to the request's ARCP	$\leq$	Maximum allowable flight time between refueling areas

If the constraint is not satisfied, go to step 14.

total non-refueling flight time	$\leq$	maximum allowable non-refueling flight time

If this constraint is not satisfied, go to step 14.

total mission length $\leq$ maximum mission length

If this constraint is not satisfied, go to step 14.

16. Tanker and request are considered to be a possible refueling, go to step 14.

The following algorithm determines the precoordinated tankers that can satisfy a request first and then perform its precoordinated refueling. Many of the steps here are similar to those for the precoordinated tanker performing its precoordinated refueling first.

17. Can base perform a refueling at the request's refueling area?

flight time from the base to the request's ARCP $\leq$ maximum allowable flight time from the base to the first ARCP

If this constraint is satisfied then each tanker at the precoordinated base is compared to each request for the area. If the constraint is not satisfied, go to step 3.

18. Take first tanker at the base.

19. Go to step 21.

20. Take next precoordinated tanker at the base. If there are no other tankers at the base, go to step 26.

21. Take first request at the refueling area.

22. Go to step 24.

23. Take next request at the refueling area. If there are no more requests at the area, go to step 20.

24. Can the tanker satisfy the request first and then perform its precoordinated refueling?

a. ARCT compatibility between the tanker and the request?

(request's earliest ARCT) + (request's refueling time) + (flight time to tanker's ARCP) + (orbit time) $\leq$ tankers latest ARCT

(request's latest ARCT) + (request's refueling time) + (maximum allowable flight time between refueling areas) + (orbit time) $\geq$ tanker's earliest ARCT

If both constraints are not satisfied, go to step 14.

b. Any flight time limitations exceeded?

flight time from the end refueling point for the request to the ARCP for the tanker $\leq$ Maximum allowable flight time between refueling areas

If the constraint is not satisfied, go to step 23.

total non-refueling flight time $\leq$ maximum allowable non-refueling flight time

If this constraint is not satisfied, go to step 23.

total mission length $\leq$ maximum mission length

If this constraint is not satisfied, go to step 23.

25. Tanker and request are considered to be a possible refueling, go to step 23.

26. If the precoord-first-flag is "off", go to step 3. If it is "on", compare all of the possible refuelings for the base and the refueling area. Determine if there are any possible refuelings where a precoordinated tanker can satisfy the same request either before or after it performs its precoordinated refueling. If there are, determine which of the scenarios has the longest mission length and discard it. It is no longer considered to be a possible refueling.

27. Go to step 3.

28. STOP.

Bibliography

1. Congress of the United States Congressional Budget Office. Modernizing the Aerial Tanker Fleet: Prospects for Capacity, Timing, Cost. Washington: Congressional Budget Office, Sept 1985.

2. Department of Defense. Flight Information Publication, AP/1B. St. Louis: Defense Mapping Agency Aerospace Center, 28 August 1986.

3. Department of the Air Force. Air Refueling Management (KC-135 and KC-10). AFR 55-47. Washington: HQ USAF, 03 Aug 1984.

4. Department of the Air Force. USAF Series Air Refueling Manual. T.O. 1-1C-1-3. Oklahoma City: HQ USAF, 10 Aug 1986.

5. Ford, L. R. and Fulkerson, D. R. "Solving the Transportation Problem", Management Science, 3: 24 - 32 (October 1956).

6. Goicoechea, A. and others. Multiobjective Decision Analysis with Engineering and Business Applications. New York, John Wiley and Sons, 1982.

7. Headquarters Strategic Air Command, HQ SAC/SIC, Scheduler. Computer Program Documentation. Offutt AFB, 1964.

8. Hillier, F. S. and Lieberman, G. J. "Introduction to Operations Research". Oakland: Holden-Day, Inc., 1980.

9. Ignizio, J. P. Linear Programming in Single- & Multi-objective Systems. Englewood Cliffs: Prentice-Hall, Inc, 1982.

10. Martin, Maj Elmer, Applications Division of Tanker Operations. Telephone interview. HQ SAC/DO8A, Offutt AFB NEB, 15 August 1986.

11. Martin, Maj Elmer, Applications Division of Tanker Operations. Personal interview. HQ SAC/DO8A, Offutt AFB NEB, 29 - 30 September 1986.

12. McElhinney, Lt Col Robert E., Chief, Applications Division of Tanker Operations. Personal interview. HQ SAC/DO8A, Offutt AFB NEB, 29 - 30 September 1986.

13. McElhinney, Lt Col Robert E., Chief, Applications Division of Tanker Operations. HQ SAC/DO8A. Point Paper on FQ 1-87 AR Scheduling Results. Unpublished report for record. HQ Strategic Air Command, Offutt AFB NEB, 04 September 1986.

14. Rosenthal, R. E., "Principles of Multiobjective Optimization," Decision Sciences, 16: 133-149 (Spring 1985)

15. Sunset Software. A Professional Linear and Mixed Integer Programming System. Computer Software Documentation. San Marino: Sunset Software, June 1985.

16. Taylor, John W. R. and Susan H. H. Young. "Gallery of USAF Weapons," Air Force Magazine, 157: 152 (May 1986).

Vita

Captain Harry C. Hostler was born on 19 October 1955 in Texarkana, Arkansas. He graduated in 1973 from Waianae High School in Waianae, Hawaii. He immediately attended the University of Hawaii from which he received a Bachelor of Arts in Mathematics in May 1977. Upon graduation he received a commission in the USAF through the two year ROTC program. He entered the Air Force in November 1977 to attend Undergraduate Navigational Training at Mather AFB, California. Upon earning his wings in June 1978 he was assigned to Kadena AB, Japan as a KC-135 navigator. After an 18 month tour he was reassigned to Fairchild AFB, Washington where he served as an instructor navigator. In March 1982, he was assigned to PACAF as an EC-135J navigator. While there he held the positions of chief instructor navigator and also senior navigational flight examiner. While stationed in Hawaii he earned a second bachelors degree in computer science. He left Hawaii in July 1985 to enter the School of Engineering, Strategic and Tactical Science Program, AFIT, Wright-Patterson AFB, Ohio.

Permanent Address: 100 Heritage St.
Jacksonville, ARK 72076

UNCLASSIFIED
SECURITY CLASSIFICATION OF THIS PAGE

AD-A182 227

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

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/GST/ENS/87M-7	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 Wright-Patterson AFB, Ohio 45424	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 NUMBERS PROGRAM ELEMENT NO. / PROJECT NO. / TASK NO / WORK UNIT ACCESSION NO.
11. TITLE (Include Security Classification) See Box 19	
12. PERSONAL AUTHOR(S) Harry C. Hostler B.A., Capt USAF	
13a. TYPE OF REPORT MS Thesis 13b. TIME COVERED FROM _____ TO _____	14. DATE OF REPORT (Year, Month, Day) 1987 March 15. PAGE COUNT 93
16. SUPPLEMENTARY NOTATION	

17. COSATI CODES			18. SUBJECT TERMS (Continue on reverse if necessary and identify by block number)
FIELD	GROUP	SUB-GROUP	
01	02		Refueling, Scheduling, Tanker Aircraft, Operations Research; Network Flows

19. ABSTRACT (Continue on reverse if necessary and identify by block number)

Title: AIR REFUELING TANKER SCHEDULING

Thesis Advisors: Thomas F. Shuppe, Lt Col, USAF
Instructor in Operations Research

William F. Rowell, Lt Col, USAF
Assistant Professor of Operations Research

Approved for public release: IAW AFR 190-1.
5 May 87
JOHN E. WOLAVER
Dean for Research and Professional Development
Air Force Institute of Technology (AU)
Wright-Patterson AFB OH 45433

20. DISTRIBUTION/AVAILABILITY OF ABSTRACT ☒ UNCLASSIFIED/UNLIMITED ☐ SAME AS RPT ☐ DTIC USERS	21 ABSTRACT SECURITY CLASSIFICATION UNCLASSIFIED
22a. NAME OF RESPONSIBLE INDIVIDUAL Thomas F. Shuppe, Lt Col, USAF	22b. TELEPHONE (Include Area Code) (513) 255-3362 22c. OFFICE SYMBOL AFIT/ENS

DD Form 1473, JUN 86 — Previous editions are obsolete. — SECURITY CLASSIFICATION OF THIS PAGE

UNCLASSIFIED

This thesis determined a way to schedule SAC's air refueling tanker fleet to perform, if necessary, more than one refueling mission during a flight. A preemptive goal programming approach was adopted using three priority levels. The basic formulation was that of the generalized assignment problem. Three objectives had to be considered when performing the task, maximize the number tanker requests satisfied, maximize the number of category B requests satisfied, and minimize the total flight time to perform all of the missions.

A preprocessor was developed to transform the inputs from the tanker and receiver scheduling units into a usable format to be executed by a mixed integer programming package. This preprocessor determined all of the possible refuelings that could take place, computed the flight times of the missions, and determined all of the variables to be used in the constraints and objective functions.