3-1-2005

A Capacitated Facility Location Approach for the Tanker Employment Problem

Jeffrey R. Miller

A CAPACITATED FACILITY LOCATION APPROACH FOR THE TANKER EMPLOYMENT PROBLEM

THESIS

Jeffrey R. Miller, Second Lieutenant, USAF

AFIT/GOR/ENS/05-12

DEPARTMENT OF THE AIR FORCE
AIR UNIVERSITY

AIR FORCE INSTITUTE OF TECHNOLOGY

Wright-Patterson Air Force Base, Ohio

AFIT/GOR/ENS/05M-12

A CAPACITATED FACILITY LOCATION APPROACH FOR THE TANKER EMPLOYMENT PROBLEM

THESIS

Presented to the Faculty

Department of Operational Sciences

Graduate School of Engineering and Management

Air Force Institute of Technology

Air University

Air Education and Training Command

In Partial Fulfillment of the Requirements for the

Degree of Master of Science in Operations Research

Jeffrey R. Miller, BS

Second Lieutenant, USAF

March 2005

AFIT/GOR/ENS/05M-12

A CAPACITATED FACILITY LOCATION APPROACH FOR THE TANKER EMPLOYMENT PROBLEM

Jeffrey R. Miller, BS
Second Lieutenant, USAF

Approved:

Dr. James T. Moore (Chairman)	date
Victor D. Wiley, Maj, USAF (Member)	date

AFIT/GOR/ENS/05M-12

Abstract

Air refueling is conducted to provide rapid response, increased range, and extended airborne operations for bombers, fighters, airlift, command and control, and intelligence, surveillance, and reconnaissance aircraft. The planning and scheduling of limited tanker resources during employment operations is a major concern for Air Mobility Command (AMC). Current tools can run as long as two weeks, and most actual planning is done by hand. AMC desires a simple tool that runs in a short amount of time to aid in planning operations.

The tool developed allows AMC to input sorties consisting of various aircraft types and armaments, departing from multiple bed down locations in theater . Each sortie departs and returns to a base of origin, and is assumed to be attacking or patrolling in an engagement zone defined by the user. The user is also able to specify the locations of available military tanker aircraft. The problem is modeled as a capacitated facility location problem with sole sourcing constraints. The methodology is applied to partition the tankers and refueling points to anchor areas, surrounding the engagement zone so that all receivers can be refueled during their attack operations. Secondary goals include minimizing the number of tankers required (or maximizing the number of receivers supported), and limiting the total flight distance for the tanker aircraft. The TET tool uses the heuristic technique tabu search to find feasible allocations of tankers and sorties to anchor areas during employment.

Acknowledgments

I would like to express my sincere appreciation to my faculty advisor, Dr James T. Moore, for his guidance and support throughout the course of this thesis effort. The insight and experience was certainly appreciated. I would, also, like to thank my reader, Maj. Victor D. Wiley, for patiently reading my many thesis drafts, and answering all my questions no matter how trivial. Additional thanks go to the rest of the ENS faculty for providing me not only with the knowledge, but also confidence to complete this thesis. The instruction, mentorship, and motivation were leaps and bounds above anything I could have expected to receive at any other institution. Thank you to Maj Mark MacDonald for giving me those missing links to make my code make sense.

A special thanks goes to my friends and classmates. Frank, JD, Casey, Clint, and Kevin, I couldn't have made it without the tutoring, listening, and laughs. Dave and Tim lets keep up the support when we get to Washington.

Finally I would like to thank my wife and baby-girl for the all the sacrifices they have made over the last 18 months. Every day you show me the meaning and purpose in all that I do. I will never be able to put my feelings into words. My wife, without your love and support I would not have made it. You make me a better man. My daughter, your smiles bring me more joy than I could have ever imagined, daddy loves you.

Jeffrey R Miller.

Table of Contents

List of Figures

Page

List of Tables

Page

A CAPACITATED FACILITY LOCATION APPROACH FOR THE TANKER EMPLOYMENT PROBLEM

I. INTRODUCTION

1.1 Background

Air refueling, the in-flight transfer of fuel from a tanker to a receiver aircraft, “supports the national military strategy across the spectrum of conflict, from peacetime operations for American global interest to major regional contingencies” (Ianuzzi, 1997b:15). Air refueling not only provides all types of military aircraft the extended range to reach any corner of the globe; it is equally applicable to their employment and sustainment after they get there.

> Air refueling allows airpower forces to increase levels of mass, surprise, economy of force, flexibility, versatility, and maneuverability and can concentrate more assets for offensive operations. The overall effect of these applications is a force enabler and force multiplier in airpower employment (AFDD 2-6.2 1999).

In 1923, then Major Henry “Hap” Arnold first demonstrated air refueling with an in-flight hose contact between two De Havilland DH-4B aircraft (Ianuzzi, 1997a: 22). The Army Air corps continued experimenting throughout the 1920s, and in 1929 the flight of the “Question Mark” demonstrated air refueling’s true potential. The “Question Mark,” a modified Fokker C-2A commanded by Major Carl Spaatz, made 143 contacts with two modified Douglas C-1 biplanes and remained airborne for over 150 hours and an equivalent of 11,000 miles (Iannuzzi, 1997a:22). This flight established new records for both air refueling and endurance.

An aircraft's ability to remain airborne is limited by its capacity to carry fuel. Air refueling removes this restriction. Air refueling has two main roles, "force-enhancement" and "force-multiplication." Force-enhancement, the traditional tanker role, is achieved through deployment of aircraft to any part of the world previously unreachable without enroute landings to refuel. Capehart (2000), Wiley (2001), Tekelioglu (2001), and Annaballi (2002) developed models to help efficiently utilize tankers in their deployment missions. The second role, that of a force-multiplier, is used in short-range theater operations. The ability to aerially refuel allows aircraft to strike multiple targets, to be stationed beyond the effective range of enemy weapons, to increase payload, and to take multiple routes into an area of responsibility thus achieving surprise. Nearly all military aircraft can be refueled in-flight today. Air Mobility Command (AMC) based at Scott Air Force Base, Illinois controls all aerial refueling operations. The KC-10 extender and KC-135 Stratotanker are the primary refuelers in the Air Force Fleet.

The KC-135 entered the Air Force inventory in 1956 to extend the range of Strategic Air Command's B-52 fleet (Capehart, 2000:1). The KC-135 is a modified Boeing 707, boasting a range of 1,500 miles and able to carry a payload of 150,000 pounds of transfer fuel (Air Force Magazine, 2000:143). Table 1 outlines the deliverable fuel capacities for different tanker types with a radius of 2500 nautical miles (nm). The amount of deliverable fuel is inversely proportional to the range; as range increases, deliverable fuel decreases (see Figure 1). Introduced in the 1980s to supplement the KC-135 fleet was the KC-10, a modified Douglas DC-10. The KC-10 has the additional ability to carry cargo along with or instead of fuel. As of January 2005, the Air Force has a total of 530 KC-135s in its Active Duty, Reserve and Guard fleet along with 59 KC-10s in its Active Duty fleet (Air Force Link, 2003).

Table 1. Deliverable Fuel Capacities (radius of 2500NM)

Tanker Type	Fuel (pounds)
KC-135A	63,0000
KC-135R	94,500
KC-135E	75,600
KC-10	162,000

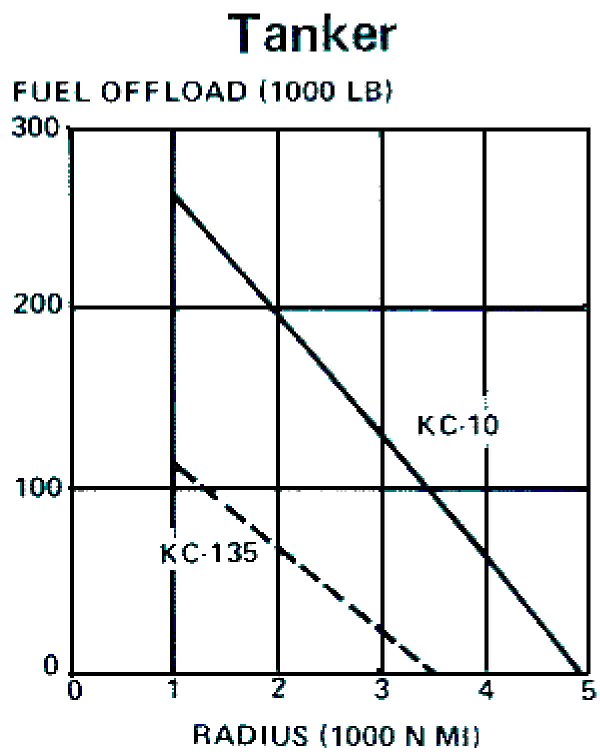

Figure 1: Deliverable Fuel vs. Radius

Air refueling has demonstrated its crucial role in supporting air operations during recent conflicts. During Operation Desert Shield and Desert Storm, 400 tankers flew over 30,000 sorties and logged over 140,000 hours of flight time according to the *Gulf War Air Power Survey*. 80,000 aircraft received over 1.2 billion pounds of fuel from tanker aircraft (Wiley, 2001:2). Tanker aircraft were also very active during Operation Allied Force, offloading 356 million pounds of fuel (Simpson, 2000:10). The lack of forward basing surrounding Afghanistan during Operation Enduring Freedom confirmed the need for air refueling. Seventy percent of the Air Force's active duty tanker fleet was utilized supporting the air campaign, despite the fact that strike sorties rarely exceeded 100 per day (Newman, 2002:57). And with the most recent conflict in Iraq, Operation Iraqi Freedom, tankers flew over 9,000 sorties and offloaded over 400 million pounds of fuel in the first month (Moseley, 2003).

1.2 Problem Statement

AMC's Tanker Airlift Control Center (TACC) is responsible for planning and scheduling all tanker operations in support of air operations. As the lead command for the Air Force's air refueling fleet, AMC frequently examines the capability of current and proposed tanker fleets along with their supporting manpower and resources to meet wartime requirements. Analysts in the command currently use a very detailed tanker planning tool called the Combined Mating and Ranging Planning System (CMARPS). This tool provides actual tanker/receiver aircraft schedules and flight plans which take into account numerous system constraints. Unfortunately, this tool takes a long time, a lot of data, and a lot of operational expertise to set-up properly for a given scenario and is not well suited to quick turn analysis or gross feasibility checks. AMC needs a more efficient tool for analysis of tanker capabilities during the employment phase of military operations.

1.3 Research Objectives

Previous theses by Capehart (2000), Tekelioglu (2001), Annaballi (2002), as well as a dissertation by Wiley (2001) have developed MS Excel and java-based tools, which can be used to solve the tanker assignment or deployment problem. They have used tabu search and ant colony techniques to develop solutions to assignment and scheduling problem formulations.

The objective of this research is to develop a tool similar to the Excel-based tool, called TAP, used for the deployment problem, capable of analyzing the tanker employment problem. The new tool, called Tanker Employment Tool (TET), builds upon the functionality of the TAP tool while applying it to a similar yet different class of problem.

Additional functionality of guided input menus and help screens allows the user to be able to add additional airbases, receiver aircraft, and/or tanker aircraft to the database.

1.4 Scope

A complete tanker planning model would address both the force-enhancement as well as the force-multiplier roles discussed previously. The previously developed TAP tool addresses the deployment phase of the overall model. TET is a compliment to the previously developed TAP tool by addressing the employment planning process. This project focuses on answering the following questions:

1. Given system constraints/capacities, and information on receiver employment missions, how many tankers will it take to meet receiver air refueling requirements?
2. Given system constraints/capacities, and a fixed number of tankers, how many receiver employment missions can be supported?

This problem involves non-homogeneous vehicles, at multiple locations, assigned to specific anchor points that service receiver groups. In this sense the problem can be viewed as a multi-depot vehicle routing problem. Anchor points are two stationary points, around which an elliptical refueling tract is flown. Since tankers and receiver groups need to be assigned to anchor points, it can also be viewed as an assignment problem. The introduction of time windows adds an additional scheduling problem. If we consider anchor areas to be facilities that service demand, the problem can be modeled as a facility location problem.

The preferred air refueling method for intratheater operations (employment) is to use anchor areas. The tanker flies a racetrack pattern within defined airspace while waiting for receiver aircraft to arrive. Once joined with the receiver, the tanker then flies in an expanded racetrack pattern while refueling the receiver (AFDD2-6.2). Anchor areas can be viewed as refueling facilities, with the receiver groups as demand points. In this sense the problem can be viewed as a capacitated facility location problem with single source constraints.

Factors affecting this problem include tanker and receiver aircraft fuel capacities and burn rates, number of aircraft to be supported, number of available tankers, distance to engagement area, time frames, formation size, offload rates, and weapons load. Other factors such as wind, altitude, and crew duty limitations are not explicitly modeled.

1.5 Assumptions

Planning operations for intratheater operations is a multi-tiered decision process. It is assumed, for the purpose of this research, that decisions on aircraft bed down locations, sorties, and area of engagement have already been made. The effects of these decisions are not considered and are left for future research.

1.6 Contribution of Research

The goal of the research is to provide AMC with a quick look tanker employment planning tool. This tool adapts previous efforts made in the deployment phase to be applicable to employment. This tool provides AMC with the ability to perform gross feasibility checks for current or proposed tanker assets. The algorithm proposed in this research can serve as a starting point for future research in tanker-scheduling for theater operations.

1.7 Thesis Overview

The remainder of this thesis is organized as follows: Chapter 2 reviews the literature pertinent to this topic. Chapter 3 presents the proposed methodology and description of model. Chapter 4 presents the results and comparisons of the model and test scenarios. Finally, Chapter 5 concludes this research and discusses opportunities for further improvement and research in this topic area.

II. Literature Review

2.1 Tanker Scheduling Tools

2.1.1 Combined Mating and Ranging Planning System (CMARPS)

The current tool used by AMC to determine how much, when and where air refueling is required for mission aircraft is CMARPS. Originally developed and introduced in 1982, CMARPS functions with several other AMC tools to provide optimized assignment of tanker resources to meet refueling requirements. CMARPS determines the refueling requirements for each receiver group (a single cell of similar aircraft types) individually, and assigns them specific tankers. Efforts to minimize the number of tanker aircraft as well as sorties flown are made in the development of the schedule, while still meeting time requirements.

Flight routes for receiver aircraft are determined by considering the following criteria: avoid restricted airspace, threat exposure, deconflict routes, and time over target. After the routes are determined, fuel requirements are calculated. CMARPS then assigns tankers considering all the following: minimize tanker usage, minimize tanker fuel consumption, air refuelable tankers, tanker reuse, and satisfy abort base requirements (Logicon, 1996).

CMARPS can run for an extremely long period of time, sometimes in excess of two weeks, and is a complex program to understand, even for the experienced user. Extensive computing resources are required which “limits its efficiency, mobility, and versatility” (Wiley, 2001:22).

2.1.2 Consolidated Air Mobility Planning System (CAMPS)

CAMPS is designed to support the rapid deployment of tankers. By combining the functionality of the Airlift Deployment Analysis System (ADANS) and CMARPS, CAMPS is used for scheduling, executing, and monitoring airlift operations to carry out the global deployment of U. S. forces (Boukhtouta, 2004:18). CAMPS, also developed by LOGICON, provides the interoperability of tanker resource and airlift mission requirements. Unfortunately, this tool also takes a long time, a lot of data, and a lot of operational expertise to set-up properly for a given scenario and is not well suited to quick turn analysis or gross feasibility checks.

2.1.3 Quick Look Tool for Tanker Deployment

Russina and Ruthsatz (1999) developed an Excel-Based tool to develop tanker deployment schedules on a day-by-day basis. The Quick Look tool determined the number of tankers required to support a desired deployment as well as how quickly the deployment could be achieved (Wiley, 2001:24).

Although the tool was developed for tanker deployments, several of the assumptions and issues are directly applicable to tanker employment. The following is a review of these issues and the assumptions made.

Time, air speed and travel distance are obviously important factors to include in the model. For the employment problem, the only known value is the great circle distance between the base of origin and the area of responsibility. Increasing the rate at which receiver aircraft travel changes time requirements and increases fuel consumption.

Varying receiver's ingress routes also affects the complexity of the problem. For these reasons, assumptions must be made and clearly stated prior to beginning an analysis of the problem.

The geographical position of the airbases involved in theater operations is also an important factor to consider. If tanker resources are located on the opposite side of the engagement zone, either the tanker or receiver aircraft must fly around the strike area to avoid possible threats. For this reason, it is desirable to have some proximity between tanker and receiver airbases. Tanker recycling, or reuse must also be considered. If multiple sorties have similar refueling requirements that can be "stacked" in succession, it is possible for a single tanker to service many receiver groups. This situation brings another situation to the forefront, that of timing. The Quick Look Tool determined schedules in terms of days. In the employment scenario, schedules need to be in terms of hours and minutes.

Air Force regulations limit the length of time a crew may fly without a rest period. These regulations may be stretched in a war time environment, but they still need to be considered in determining flight durations.

Another factor taken into consideration in deployment models that can be adapted to employment is the amount of take off fuel. Deployment scenarios must consider origin airbase weather requirements for takeoff, which in turn affect air refueling requirements. In employment scenarios, weapons load, runway length, and origin weather affect take off fuel. It is not uncommon for military aircraft to takeoff with limited fuel in order to achieve flight, and immediately require air refueling before beginning their ingress route.

Fuel consumption is also very important to the model. Russina and Ruthsatz (1999) do an adequate job identifying and modeling the factors that determine fuel consumption rates. True air speed directly affects fuel consumption as the higher the rate of speed, the higher the rate of fuel consumption. Ground speed determines the amount of time required to traverse a specific distance, and is affected by wind conditions, which in turn affects fuel consumption. Fuel consumption rates are also inversely proportional to the aircrafts altitude (lower altitude equates to higher fuel consumption). Gross weight, alluded to earlier with weapons load, also directly affects fuel consumption rates (higher weight leads to higher consumption).

2.1.5 TAP Tool

Three AFIT masters students, Capehart (2000), Tekelioglu (2001), and Annaballi (2002), built upon the QLT and previous TAP efforts. TAP is also an Excel-based spreadsheet model, built in Visual Basic for Applications (VBA). Multiple worksheets provide input data to the model regarding tanker resources and receiver data.

Both Capehart and Tekelioglu viewed the problem as an assignment problem and used a heuristic method to speed up computing. Tekelioglu extended the tabu search methodology of Capehart by adapting the heuristic to include a relatively simple reactive tabu tenure. All refueling points are calculated up front, based on maximum flight distance of the receiver groups. Annaballi updated the TAP tool by making it more operationally realistic and applying an ant colony heuristic to a vehicle routing problem formulation.

2.1.6 Wiley's Group Theoretic Tabu Search Tool (GTTS)

As part of his doctoral research, Wiley (2001) developed a JAVA-based tool for the tanker assignment problem. GTTS was shown to improve upon the results of the TAP tool, although it was still limited to deployment. He eliminated the requirement for tankers to return to their base of origin, calculated refueling points to reduce the distance tankers had to travel, rather than the maximum range of the receiver aircraft, and allowed a tanker to service more than one refueling point (Wiley, 2001:108).

Although the results of GTTS were an improvement, runs could take upward of 30 minutes to produce results while the TAP tool would take about half as long (Wiley, 2001:120). Also Wiley's tool is not spreadsheet based (Annaballi, 2002: 16).

2.2 Scheduling Theory

Scheduling exists in practically all settings. Every person practices scheduling theory almost everyday during their lives. Have you ever tried to figure out the best way to get all your errands done before lunch? Scheduling concerns the allocation of a limited pool of resources to a finite set of tasks over time. It is the decision making process that has as a goal the optimization of one or more objectives (Pinedo, 1995: 11). Scheduling is crucial in the manufacturing arena, as well as military applications. In the military, scheduling is widely used in weapons system development and flight scheduling (Calhoun, 2000: 22).

Schedules consist of resources, tasks, and objectives. Examples of resources include engineers, drills, tanker aircraft, runways, or anything else that is needed to complete a task. Tasks can be a specific operation in an assembly process, analysis of an intelligence report, take-offs or landings at an airport, or refueling a group of aircraft. Objectives are the measurement of success, or goals. They include minimization of time to completion in an assembly process, minimization of the amount of total lateness, minimization of cost, maximization of profits, or minimization of total resources required.

2.3 Parallel Machine Models (PMMs)

PMMs are common scheduling problem formulations. A machine is a finite resource required for completing a task, such as a plasma cutter in a machine shop, an editor of a news story, a cashier in a checkout line, or tanker aircraft in theater. A parallel machine model consists of *m* machines which operate in parallel to process a set of jobs. Each job *j* requires processing on a single machine, and may be served by any of the *m* machines in parallel. For example, a four ship of F-16s (job *j)* needs to be refueled before crossing the border for a Close Air Support (CAS) mission. They can be refueled by any of the available tankers (*m* machines) in the area. Parallel machines can be either identical or unrelated. If the machines are identical, they process an identical job *j* at the same speed, while unrelated machines will process job *j* at different rates.

The most common objective of PMMs is to minimize the makespan, or the completion time of the last job. In this application, the primary focus is balancing the load across all machines while still meeting time requirements. Load balance is ensured by minimizing the makespan (Pinedo, 1995: 61). The problem can be decomposed into a two step process, allocate jobs to each machine, and then determine the sequence of those jobs on each machine subject to any precedence constraints (Pinedo, 1995: 62).

A precedence constraint is a restriction to the problem which defines specific timing requirements between activities in both single and parallel machine models. The most common precedence (finish to start) states that job *j* must be completed before processing can begin on job *k*.

Scheduling is critical to the actual employment of tankers in theater. Particular sorties may have time sensitive targets which would require them to be refueled before any other. Also we must be able to account for the time it takes to transfer fuel to receivers. Precedence constraints also come into play when scheduling the initial, mid-mission and egress refueling points for a particular mission. Not only is there the necessary sequential nature of these refuelings to consider, but there are also specific timing requirements for the mid-mission and egress refueling points. These are not explicitly modeled in the TET tool and are left for future research.

2.4 Multi-Depot (MD) Vehicle Routing Problem (VRP)

Another approach to the Tanker Employment Problem is to model it as a VRP. The tankers are represented as finite capacity non-homogeneous vehicles with route length restrictions. They deliver to customers with a finite (>0) delivery time. Using the notation of Barnes and Carlton (1996) the TEP, like the AFRP, is a variation of a problem known to be NP-hard (Carlton 1995, Gendreau, Laporte & Potvin 1997).

$$\text{Problem } \alpha = (MV\overline{H}\text{, MD, VRP, RL})$$

where $MV\overline{H}$ ≡ Multi-Vehicle Non-Homogenous, MD ≡ Multi-Depot, VRP ≡Vehicle Routing Problem, and RL ≡ Route Length. (Wiley, 2001: 14).

The following is an outline of additional considerations applicable to the TEP problem adapted from those considerations outlined by Wiley (2001):

1. In problem α, the customers' locations are fixed in space, like a warehouse, and the decision relies only on the sequence for the assigned vehicle. Further, the demand for each customer is an a priori stipulated deterministic amount and there is a single delivery to any customer. Finally, the route length restriction is given only in terms of a total travel distance that may not be exceeded. Problem α has no explicit accounting for the timing of events. In the TEP, we know only the total amount of fuel that must be provided to a RG before entering the engagement. We have the choice of fixing the customers location in 3-dimensional space, or allowing that location to be determined by the model to allow the RG and tanker to "meet in the middle" so as to improve efficiency. As in problem α, we must stipulate the responsible vehicle (tanker) and the ordering of any delivery. We also must designate the start time of each fuel delivery and the number of possibly multiple deliveries and the amount of product (fuel) to be provided in each delivery.
2. All customers must be supplied with fuel in a timely manner that will assure that no receiving aircraft has its available fuel fall below a prespecified "minimal reserve."
3. Directly associated with the waypoint (WPT) decisions are the decisions on the takeoff time of each receiver group (RG) and the possibly multiple takeoff times of each tanker.

We could relax many of these requirements and make the TEP problem emulate α. The introduction of time windows for "deliveries" (refuelings) leads to an adapted formulation by Barnes and Carlton (1996).

Problem β = (MV $\overline{H}$, MD, VRP, RL, TW)

using the notation of problem α with TW $\equiv$ Time Windows

Wiley (2001) states this formulation reintroduces some of the time-based considerations, i.e., time ordered precedence relations between events, but would still require the spatial location and fuel requirement of each WPT to be fixed known constants.

Finding a feasible solution to a TSP with TW, a relaxation of the above formulation has been shown to be NP-hard (Savelsbergh 1992). Every increase in computing power has been offset by the expandability of the problem. For this reason, the most promising research in obtaining good feasible solutions in a reasonable amount of time has been through the use of heuristic and meta-heuristic techniques (Wiley 2001: 27).

2.5 Capacitated Facility Location Problem (CFLP) with Sole Sourcing (SS)

A third approach to modeling the TEP is to treat the anchor areas as capacitated facilities. The CFLP describes a wide variety of planning problems. Additional applications include lot sizing decisions in production planning; telecommunications network design; machine replacement; vehicle routing when capacities are not equal (Conuejols et. al., 1991); optimal stationing of Units at Bases (Jackson, 1995); stochastic transportation problem; and discrete network design (Holmberg, 1990).

Receivers must be completely refueled by a single tanker. This restriction to the problem is modeled in the CFLP as a sole source constraint. The addition of sole sourcing constraints is an important variant to the CFLP. All customer demand must be met by a single facility. In our example, all fuel for a particular refueling point must be received from a single anchor area. The CFLPSS is a pure integer linear program with a high number of binary variables. The following is the binary formulation of the CFLPSS:

Indices:

i	facilities
j	demand points

Data:

f_i	fixed cost to operate facility i
$c_{i,j}$	cost to supply all demand at j by facility i
d_j	total demand at j
s_i	capacity of facility i

Binary Variables:

y_i = 1 if facility i is open and 0 if it is closed.

$x_{i,j}$ = 1 if demand at j (d_j) is provided by facility i.

Formulation:

$$\text{minimize} \qquad \sum_i f_i \cdot y + \sum_i \sum_j c_{i,j} \cdot x_{i,j} \qquad (1)$$

s.t.

$$\sum_i x_{i,j} = 1 \qquad \forall j \qquad (2)$$

$$x_{i,j} \le y_i \qquad \forall i, j \qquad (3)$$

$$\sum_j d_j \cdot x_{i,j} \le s_i \qquad \forall i \qquad (4)$$

$$\sum_i s_i \cdot y_i \ge \sum_j d_j \qquad (5)$$

$$x_{i,j} \in \{0,1\} \qquad \forall i, j \qquad (6)$$

$$y_i \in \{0,1\} \qquad \forall i \qquad (7)$$

The objective is to minimize total cost, and the objective function (1) contains two distinct sets of binary decision variables. The first set of decision variables (y_i) determine which facilities to open. The second set of decision variables ($x_{i,j}$) allocate customer demand to the open facilities. Constraints (2) ensure all demand is met. Constraints (3) ensure demand is only allocated to an open facility. Constraints (4) enforce capacity restrictions on the facilities. Constraint (5) ensures the total capacity can meet the total demand. The final two sets of constraints (6 and 7) enforce the binary restriction.

This formulation has to be modified in order to fit the tanker employment problem. In order for a facility (anchor area) to be open, a tanker must be assigned to this route. The distance of the anchor area from the tanker base affects the available offload fuel, or capacity, as well as the cost of operating the facility. Receiver groups, or customers, also have range restrictions which limits the facilities that can meet their demands. Among the facilities that are within range, the amount of fuel demanded will also fluctuate. Assuming average offload available for tankers, as well as maximum onload (completely refueling from reserve to max fuel levels) for receivers could relax these restrictions.

Instances of the CFLPSS with relatively low numbers of available facilities and customers are difficult to solve; it is a combinatorial optimization problem that belongs to the class of NP-hard problems (Cortinhal and Captivo, 1993: pp 334). Several authors have studied this problem. The problem has been formulated as a set partitioning problem and solved by a column generation branch and bound procedure (Neebe and Rao, 1983). It has also been attacked using a Lagrangean heuristic with the customer assignments relaxed. This heuristic consists of two stages, one for plant selection and another for client assignment (Barceló and Casanovas, 1984). Other researchers have combined a Lagrangean relaxation with a search procedure. Cortinhal and Captivo (2004) use a tabu search procedure to find feasible solutions which give upper bounds along with a Lagrangean relaxation to generate lower bounds. All of these previous studies for the capacitated facility location problem use a form of a heuristic to find feasible solutions in a short period of time. Recent work in the area has focused on the performance of these heuristics in obtaining feasible solutions (Cortinhal and Captivo, 2004).

2.6 Heuristics

In *How to Solve It: Modern Heuristics*, Michalewicz and Fogel (2004) give four reasons problem solving in real life is difficult: (i) complex problems often pose an enormous number of possible solutions. (ii) we often have to simplify problems to make them tractable, (iii.) conditions of problems change over time, and (iv.) problems often have constraints that require special operations to generate feasible solutions.

We can enumerate all possible solutions and simply select the best one. However, if we are solving a tanker employment problem with only 30 flights, there are $30! = 2.6 \times 10^{32}$ possible solutions. A computer evaluating 1 trillion solutions per second would take 8.4×10^{12} years to evaluate all possible solutions. Obviously, we need a set of rules to guide our search quickly to a good solution; this is where heuristics enter. The word heuristic is defined by Webster as “involving or serving as an aid to learning, discovery, or problem-solving by experimental and especially trial-and-error methods” (Merriam-Webster 1986). In practice, good heuristics provide near optimal solutions to complex problems in a short amount of time. Heuristics are not rules of thumb, as the term is traditionally used; in search procedures, the word heuristic designates a particular set of rules to follow to only enumerate portions of the feasible region in order to find a good solution.

Why would we settle for a good solution? There are algorithms such as the simplex method, or large scale decomposition methods that use special structures inherent in the problem to find the global optimum without having to completely enumerate all solutions. However many of these methods may still take too long or be too complex for the decision maker to understand. As aforementioned, we may have had to make simplifying assumptions in order to model the problem, or the inputs to the problem may have been estimated. In either case, the “optimal” solution is purely academic and a heuristic solution is good enough. A fast near-optimal solution is much more useful than a time consuming exact answer to an inexact problem (Zanakis 1981; 85).

Good heuristics have several features in common: (i) they are simple and easy to understand, (ii) they require reasonable core storage requirements, (iii) they have fast computation speed, computation times do not grow exponentially as problem size increases, (iv) the solutions generated are accurate, as determined by the user, (v) they are robust, as size and parameters change the method still performs well, (vi) they allow for multiple starting points including infeasible solutions, (vii) they output multiple solutions allowing the user to make the final selection, (viii) they contain a set of stopping criteria, and (ix) user interaction is allowed (Zanakis 1981; 85-86). Most modern heuristics are actually meta-heuristic techniques in that they consist of a construction heuristic which generates initial solutions, a local search/improvement procedure which improves upon the current solution, and a managing heuristic which ensures ample search area coverage.

> The best methods appear to be those encompassing hybrid systems such as local search techniques embedded within a meta-strategy that employ a simple neighborhood structure and transcend poor local optimality by allowing non-improving moves (Jain and Meeran 1999).

The most promising methods include ant colony, genetic algorithms, simulated annealing, and tabu search.

2.7 TABU SEARCH (TS)

2.7.1 Introduction

"Tabu Search is a meta-heuristic that guides a local heuristic search procedure to explore the solution space beyond local optimality" (Glover and Laguna, 1997: 2). All meta-heuristics include three main components: a construction heuristic, a search heuristic, and stopping criteria. TS contains additional components (Ben-Daya and Al-Fawzan, 1998: 90) that are described in more detail in the sections that follow:

Tabu list. Contains attributes associated with recent moves. The tabu list helps prevent cycling.

Aspiration criterion. A tabu move is allowed if it satisfies some criterion.

Intensification scheme. Used to intensify the search in promising regions of the solution space.

Diversification scheme. Used to move to an unexplored area.

Tabu search is applicable to a number of different problem types; these include planning and scheduling problems, vehicle routing problems, integer programming, assignment problems, and others (Glover and Laguna, 1997: 267-303).

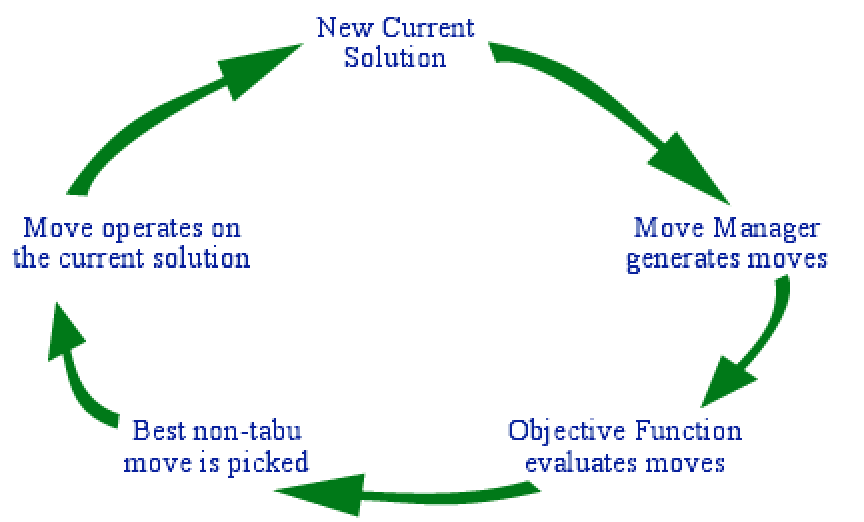

Figure 2. One Iteration of TS (Harder, 2001)

TS systematically explores the solution space by moving from one solution to another. Each move is made to some best solution in the neighborhood of the current solution; this new solution need not be an improving solution. The neighborhood is defined as all possible moves from the current solution. The determination of a neighborhood for a particular solution x is accomplished through a simple move operation σ. The interchange of two objects in a solution is a common move in sequencing problems (Glover and Laguna, 1997: 5). Other examples of moves are changing the value assigned to a variable, adding or deleting an element from a set, and so on (Glover, 1990: 78). Normally neighborhoods are assumed to be symmetric; i.e. x' is a neighbor of x if and only if x is a neighbor of x' (Glover and Laguna, 1995: 153).

After constructing an initial solution, which may or may not be feasible, TS iteratively moves through the solution space searching for a better solution. In order to prevent cycling and direct the search to other regions of the solution space that have not yet been explored, operations that move to solutions with particular attributes are forbidden, or tabu. The concepts of short and long term memory are heavily relied on to control the search. The size of the tabu list determines how long a solution will remain tabu. The number of iterations that attributes remain on the list, or the length of the list, is called the tabu tenure. An alternative to tabu lists based on attributes is a list based on solutions. Storing complete solutions usually consumes a large amount of memory. In order to avoid this costly "baggage" in a solution-based tabu list, a mapping function, called a *hash function*, is used to map each solution to a unique integer. This list of integers, *hash list*, then contains the function values for recent solutions (Glover and Laguna, 1997: 246). Woodruff and Zemel (1993) describe three desired properties of the hashing function: easy to update and compute, integers within a range that is easy to store, and a low probability of collision. A collision is defined as two different solutions returning the same hash function value (Woodruff and Zemel, 1993: 124). The memory required to carry a vector of integers is negligible to the computational effort used in evaluating the neighborhood.

2.7.2 Aspiration Criteria

Aspiration criteria determine when tabu restrictions can be overridden. The appropriate use of this type of criteria can be very important for enabling a TS method to perform at its best and find the best solution in the least possible time (Glover and Laguna, 1995: 178). The simplest aspiration criterion is to allow a tabu move when it yields a solution with an objective function value better than the best so far. This criterion is still widely used; however, other criteria can also prove effective for improving the search. The following outlines possible aspiration criteria (Glover and Laguna, 1995: 160).

Aspiration by default. If all moves are tabu, the “least” tabu is selected.

Aspiration by objective. Tabu move is selected if its objective value is better than best found so far.

Aspiration by search direction. If the direction of the move is the same direction as a previously improving move, and the new solution is an improving solution, the move is allowed.

Aspiration by influence. Moves of high influence change the solution structure. We would allow a tabu move that had high influence after a series of non improving moves.

2.7.3 Intensification

Strategies for intensification drive the search to thoroughly examine a particular area of the solution space. “Intensification strategies undertake to create solutions by aggressively encouraging the incorporation of good attributes” (Glover and Laguna, 1995: 159). This is accomplished in two ways; in the short term by modifying choice rules to incorporate attributes of solutions that were historically good, and in the long term by modifying choice rules to incorporate attributes from the elite candidate list (a collection of best solutions).

2.7.4 Diversification

“Diversification strategies instead seek to generate solutions that embody compositions of attributes significantly different from those encountered previously during the search” (Glover and Laguna, 1995: 159). This is usually accomplished by modifying choice rules to incorporate attributes that are used infrequently, creating new candidate lists, or by partially or fully restarting the search from a new solution. These strategies help the search avoid settling at a local optimum. Although randomization of a new initial solution and restarting the process could achieve diversification, it is not desired. A diversified collection is very different from a random collection. Diversification purposefully moves the solution to a specific region in the solution space that has not been explored. Randomization on the other hand would simply move to any area of the solution space.

2.7.5 Candidate Lists

For any non-trivial problem, the number of combinatorial choices is extremely large. Building a neighborhood of all possible moves then becomes very computationally expensive. In order to reduce the neighborhood being considered, a candidate list is constructed.

A candidate list is a subset of the neighborhood that contains solutions with certain properties. These lists can be generated by random sampling, the foundation for Monte Carlo studies, but a more purposeful construction yields better results. One approach is to decompose the neighborhood into critical subsets using some set of rules that ensure other subsets will be examined on subsequent iterations (Glover and Laguna, 1995: 170). This type of approach directly lends itself to parallel processing, with each processor searching a different subset of the neighborhood. Candidate list strategies also lend directly to diversification, and coordinating these two strategies may yield better performance of the TS. Candidate list strategies include: successive filter, aspiration plus, elite candidate list, sequential fan candidate list, and bounded change candidate list.

Elite candidate lists are a form of long term memory within TS. The list contains certain good solutions encountered during TS. Occasionally the best move from the elite list is selected and the search is intensified around this neighborhood. Elite candidate lists create a sort of diversification to and intensification around good solutions. Good solutions may be revisited and their neighborhoods, searched more thoroughly, may yield better solutions (Glover and Laguna, 1997: 63).

2.7.6 Strategic Oscillation

Another method to achieve effective interplay between intensification and diversification is strategic oscillation. This is achieved by temporarily relaxing problem constraints in a specified fashion. “Strategic oscillation operates by orienting moves in relation to a critical level, as identified by a stage of construction or a chosen interval of function values” (Glover and Laguna, 1997: 102). This critical level could be the solution’s feasibility, a particular value, different evaluation functions, or relaxing particular constraints. The approach operates until hitting a boundary, e.g. feasibility, where normally it would stop. Instead of stopping, rules are modified and the neighborhood definition extended to allow the approach to keep moving past the boundary for a specified depth. Then the approach turns around and re-crosses the boundary. This approach is repeated, creating an oscillation across the boundary. Glover and Laguna give a simple example, the multidimensional knapsack problem. Values of 0-1 are changed from 0 to 1 until reaching the boundary of feasibility. It then continues into the region of infeasibility with the same type of changes but with a modified evaluator. After a specified number of steps, it then “turns around” and starts changing values from 1 to 0 to move back towards feasibility (Glover and Laguna, 1995: 166).

2.7.7 Reactive Tabu Search

Reactive tabu search (RTS) is another method to improve the balance between intensification and diversification. RTS achieves the balance by reactively changing the tabu tenure through a feedback process while the search procedure is in progress. The tabu tenure, T, is set to an initial value (usually one), and increases as the need to diversify rises, and decreases as the need to intensify rises. Diversification is usually triggered by revisiting previous solutions. This basic method of increasing the tenure may not be sufficient to avoid long cycles. For this reason, use of another diversification scheme such as those described previously may be employed when too many configurations repeat too often in the search (Battiti and Tecchiolli, 1994: 131).

2.8 Conclusion

Tabu search methods provide good solutions with relatively small computational costs. Large complex problems involving combinatorial decision points have an inordinate number of possible solutions. These types of problems, including the Employment Tanker Assignment Problem are extremely difficult to solve. The use of heuristic approaches yields solutions in a reasonable amount of time. TS decreases the solution time without reducing the quality of the search (Cortinhal and Captivo, 1993:337). TS was chosen as the search method for the TET tool because it has been shown to yield better computational results than other search methods for the CFLPSS (Cortinhal and Captivo 1993: 338).

This chapter presented current tanker scheduling tools including their uses and drawbacks. Formulations as vehicle routing, scheduling, and facility location problems were overviewed. Basic descriptions of scheduling theory, precedence constraints and heuristics were presented to better understand possible solution methodologies. A summary of tabu search was also included. The following chapter applies some of these ideas and techniques to the Tanker Employment Problem.

III. Methodology

3.1 Introduction

This chapter details how the methods discussed in Chapter 2 were applied to the tanker employment problem. An explanation of the methods used to convert the tanker employment problem to the CFPLSS are discussed. The chapter finishes with a detailed look at the tabu search and scheduling methodology developed to schedule tanker assets.

3.2 Tanker Employment and Assignment Problem as CFLPSS

A capacitated facility location problem tries to determine the best configuration of facility locations in order to service customer demand. To interpret the tanker employment problem as a facility location problem, the anchor areas are facilities, the refueling points are customers, the costs of locating a facility are the tanker distances (distance a tanker must fly to reach an anchor area), the costs of servicing a customer are the fuel required, the capacity of facilities are the available fuel offloads, and the demand of each customer is the offload fuel required. The following is the Binary Integer Programming (BIP) formulation:

Indices:

i anchor areas (facilities)

j refueling points (demand points)

k tankers

Data:

$f_{i,k}$ fixed cost to operate anchor area i with tanker k

$c_{i,j}$ cost to supply all demand at *j* by anchor area *i*

$d_{i,j}$ total demand at *j* if serviced by facility *i*

$s_{i,k}$ available offload fuel at anchor area *i* manned by tanker *k*

Constants:

K total number of tankers available

Binary Variables:

$y_{i,k}$ = 1 if anchor area *i* is flown by tanker *k* and 0 if it is closed.

$x_{i,j}$ = 1 if demand at *j* (d_j) is provided by facility *i*.

Formulation:

$$\text{minimize} \quad \sum_i \sum_k f_{i,k} \cdot y_{i,k} + \sum_i \sum_j c_{i,j} \cdot x_{i,j} \tag{1}$$

s.t.

$$\sum_i x_{i,j} = 1 \qquad \forall j \tag{2}$$

$$\sum_k y_{i,k} \leq 1 \qquad \forall i \tag{3}$$

$$\sum_i \sum_k y_{i,k} \leq K \tag{4}$$

$$\sum_j d_{i,j} \cdot x_{i,j} \leq \sum_k s_{i,k} \cdot y_{i,k} \qquad \forall i \tag{5}$$

$$x_{i,j} \in \{0,1\} \qquad \forall i, j \tag{6}$$

$$y_{i,k} \in \{0,1\} \qquad \forall i, k \tag{7}$$

The key to this is determining the set of possible anchor areas. For modeling purposes a general anchor area is a fifty mile by twenty mile “racetrack”; however this is a parameter that can be varied. Due to the importance and high vulnerability of tanker assets, all refuelings must take place outside the range of the enemy’s anti-air assets.

First, define the engagement area (hostile fire zone) as a rhombus by latitude and longitude of the corners. Then partition the exterior of the rhombus into anchor areas, each defined by a single latitude longitude pair, along one edge of the rhombus. The set of all possible anchor areas is then contained as a numbered list of latitude and longitude pairs. These are possible facility locations (see Figure 3).

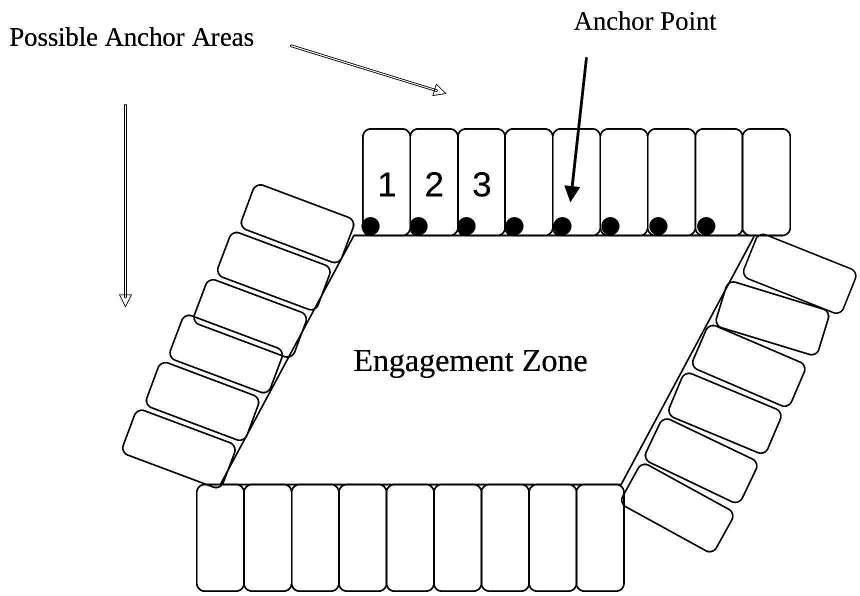

Figure 3: Possible Anchor Areas around Engagement Zone with first 3 numbered

The next step is to calculate the costs of “opening” each anchor area. This is done by calculating the great circle distance from each tanker’s base of origin to the anchor area. These costs are not “fixed” for all possible tankers, but rather are fixed for sets of tankers departing from the same base of origin. These costs are stored in a lookup matrix for use in our solution evaluation.

When calculating the distance to the anchor area, we must consider only routes that avoid traversing the interior of the engagement area (see Figure 4).

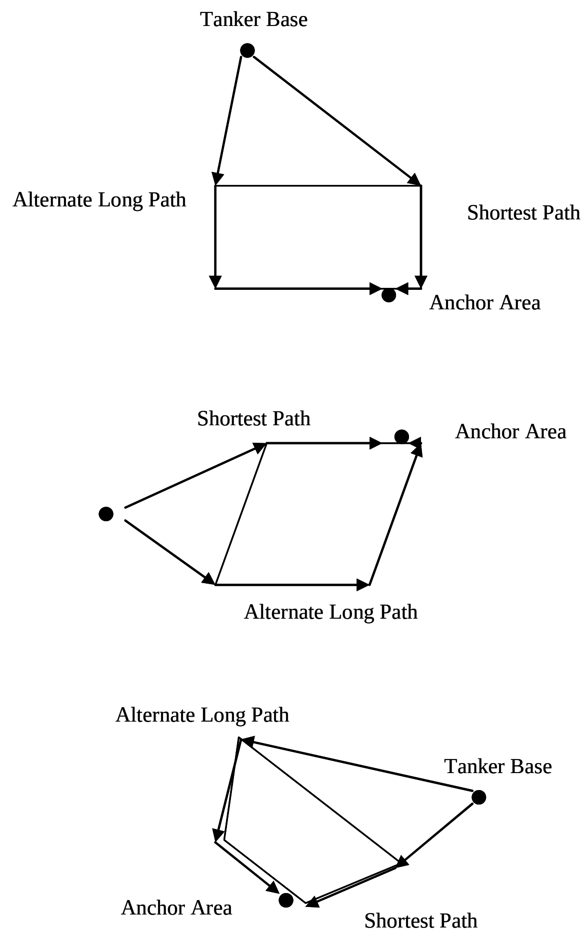

Figure 4: Engagement Area Avoidance

Since the anchor areas are touching an edge of the engagement area, we calculate the shortest path distance to the anchor area. This ensures that the tankers fly around the engagement area, if necessary, to reach anchor areas.

Using a similar approach, we calculate the cost for each refueling point to be serviced by each facility. The offload at each refueling point is equal to the amount of fuel needed to fill each aircraft to its maximum fuel load. There are three types of refueling points: initial, mid-mission, and egress. The offload required for the initial refueling point is determined simply using fuel flow, fuel burned during climb, and flight with a specific armament; it does not incorporate problem specific factors such as wind.

Then we determine the distance the receiver group can travel before it reaches its fuel reserve limit. This will determine which anchor areas are capable of servicing the initial refueling point. It is assumed that receivers will be at the fuel reserve limit for the mid-mission refuelings, and can be refueled at any anchor area. The mid-mission refueling can occur at any possible anchor area, because the airplane is already inside the engagement area, so traveling to an anchor area located on any edge of the engagement zone can be accomplished before reaching the fuel reserve limit. The amount of fuel required for the egress refueling is calculated using the same method as the initial refueling with one minor change. It is assumed that all armament will have been expended during the sortie, so the armament weight used for fuel burn calculations is zero. Additionally we assume that there is no restriction to number of refueling points an anchor area can serve. The limiting factor is the capacity of the tanker, rather than a specific number of refuelings.

Each receiver has a set service time that it takes a tanker to refuel. For simplicity, it is assumed in this problem that tankers can instantaneously pass the desired fuel offload to the receiver groups, resulting in a service time of zero. Adding the true service time would create a scheduling problem within each anchor area that could be modeled as a single machine job shop model. However, since tankers are the limiting resource in intratheater operations, available service times may be the driving factor in determining mission start times. The addition of service times and scheduling concerns is left for future research.

After determining all possible anchor areas, offload available, offload required, and associated costs, the list of decision variable (DV) alternatives represent all possible options for assigning tanker resources and refueling to anchor areas. Which of these alternatives to choose from is now the focus of the search engine.

3.3 Construction Heuristic

We begin with a construction heuristic to build an initial mission plan. Every available tanker is assigned to fly to the nearest anchor area that does not already have a tanker assigned. Then each DV, representing a refueling point, is assigned to the nearest anchor area, manned by a tanker with enough capacity to support the refueling point. If the number of tankers available is unable to meet the demand, we alert the user that the current number of available tankers, and consequently overall capacity may not be sufficient. After all DVs have been assigned, we then eliminate any unused tankers from the initial solution. This initial solution is checked for feasibility. When we begin with an infeasible solution, the first goal is to reach a feasible solution.

3.4 Solution Representation

A solution is represented by a mission plan. Each tanker is assigned to fly to a particular anchor area. Each sortie's refueling requirements are assigned to a particular anchor area. In order to determine the best allocation of resources, we must have a method for evaluating a mission plan.

3.5 Mission Evaluation

There are several goals for this problem. The first goal is to minimize the number of tankers used. This is embedded in the evaluation function to minimize cost, by assigning a fixed cost for using a tanker to “open” a facility (anchor area). The second goal is to minimize the amount of offload required. This is also embedded in the objective function as a variable cost of servicing a particular refueling point at a specific facility. Another goal is to meet all customer demand. This is represented by a hard constraint, resulting in an infeasible mission plan if any refueling point is assigned to an anchor area that does not have the capacity to service it. An infeasible initial mission plan assigns artificial tankers to service uncovered refueling points to allow the TS heuristic to proceed. The weighted objective function is outlined below:

Mission Evaluation = Facility Cost + Refueling Cost + Infeasibility Penalties

Facility Cost: The distance flown by the tanker in order to serve the anchor area

Refueling Cost: The amount of fuel offloaded.

Infeasibility Penalty: A penalty imposed if a tanker capacity constraint is violated, a tanker is assigned to more than one anchor area, an anchor area is manned by more than one tanker, or if a refueling point is not assigned to be refueled at an anchor area.

3.6 Tabu Implementation

The search engine employed in TET is tabu search. The search explores the solution space by swapping refueling points assigned to anchor areas. If the refueling point is swapped to an anchor area not currently "open," a tanker is assigned to the new anchor area. A *move* is defined as removing a customer assigned to a particular anchor area and assigning it to another anchor area.

A particular refueling point is selected for evaluation. The refueling point is removed from the current anchor area, and assigned to another anchor area within range. This new mission plan, or *neighbor*, is then evaluated. This process is repeated for every anchor area within range of the refueling point. Every refueling point in the mission plan is evaluated in a similar manner. The resulting list of mission plans and evaluations defines the *candidate list*.

TS now chooses the best mission from the candidate list, executing the *move.* After executing this *move*, TS puts this refueling point on the tabu list. TS will not change the anchor area assigned to this refueling point on this list unless it is better than any feasible mission thus far (*aspiration criteria*). The refueling point will remain on the list for a set number of iterations; this number is the *tabu tenure*. This is to regulate the short-term memory characteristic of the search. Changing the length of the *tabu tenure* will affect the behavior of the search. The default length of this tenure is set to 7; however, this can be changed by the user at the beginning of the search routine; humans have a short-term memory capacity of 7 and empirical results have shown this to be a reasonable starting value (Glover, 1997).

3.7 Tabu Search Methods

The TAP tool, Capehart (2000), allowed the user to modify three parameters regarding the tabu search process: tabu tenure, candidate list size, and size of tabu restriction. The TET tool allows for these changes as well as selecting whether or not to allow the tabu tenure to be reactive, changing the search mechanism to reactive tabu search.

Due to the problem complexity, computation time increases greatly as the number of DVs increase (Capehart, 2000). For this reason the user may select a *skip number*. This number partitions the candidate list. With a skip number of one, the entire candidate list is considered; this is the default value. With a skip number of two, every other DV alternative is considered. With the "skipped" DV alternatives evaluated on the following iteration. Any skip number greater than two, results in a similar partitioning, i.e. every third, every fourth, etc. (Capehart 2000).

There are two choices on the size of tabu restriction. A *large* restriction results in placing the refueling point in the tabu list. This refueling point cannot be assigned to another anchor area unless it satisfies the aspiration criterion. A *small* restriction allows for the refueling point to be assigned to an anchor area that is already "open," but not to an anchor area that would require the use of an additional tanker.

We employ a single version of reactive tabu search. RTS executes a minimum of 100 iterations and continues (in increments of 50 iterations) until 50 iterations fail to find a new best solution. The tabu tenure is initially set to 25% of the number of refueling points. The tenure is then adjusted depending on the performance of the search. If 25 iterations pass without finding a new best solution, we then increase the length of the tabu list by one. We continue increasing the tenure by one until a new best solution is found or the tabu list reaches a length of 75% of the customers. When either of these two conditions are reached, we reset the tenure to 25% the number of customers.

Varying these parameters forms different TS approaches. Experimenting with different combinations may yield better performance on specific problems. Table 2 outlines the basic steps of the tabu search.

Table 2: Basic Tabu Search Steps

	Step	Description
1	Generate Move	Create next solution (neighbor)
2	Evaluate Move	Determine worth of move if move is Tabu check Aspiration
3	Repeat through entire neighborhood	If entire neighborhood complete go to step 4 else repeat step 1
4	Select Best Move	Best Non-Tabu Move or Tabu move that meets aspiration criteria
5	Update Tabu list	Add move to tabu list to prevent cycling
6	Evaluate Solution	Determine objective function value
7	Continue until stopping criteria	Repeat step 1 until maximum number of iterations reached or solution is within tolerance of known lower bound

3.8 TET Model Description

Once the user inputs the data for the employment scenario, the TET tool uses three phases to arrive at an output of initial feasible and best feasible mission plans. Providing each of these as output gives the user different plans from which to make a selection. Figure 5 depicts the three phases. These phases consist of DV and cost generation, initial solution generation, and tabu search. Explanation of the DV and cost generation and initial solution phases are depicted in Figures 6 and 7, respectively.

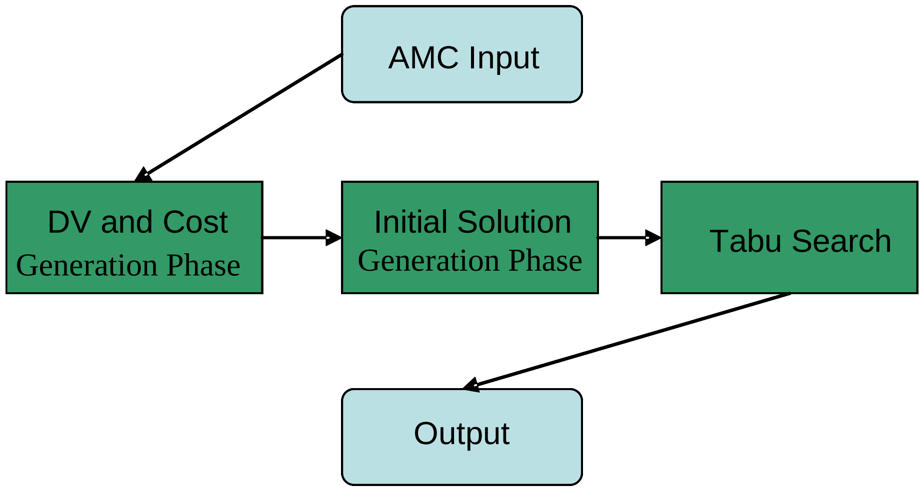

Figure 5: TET Flow Chart

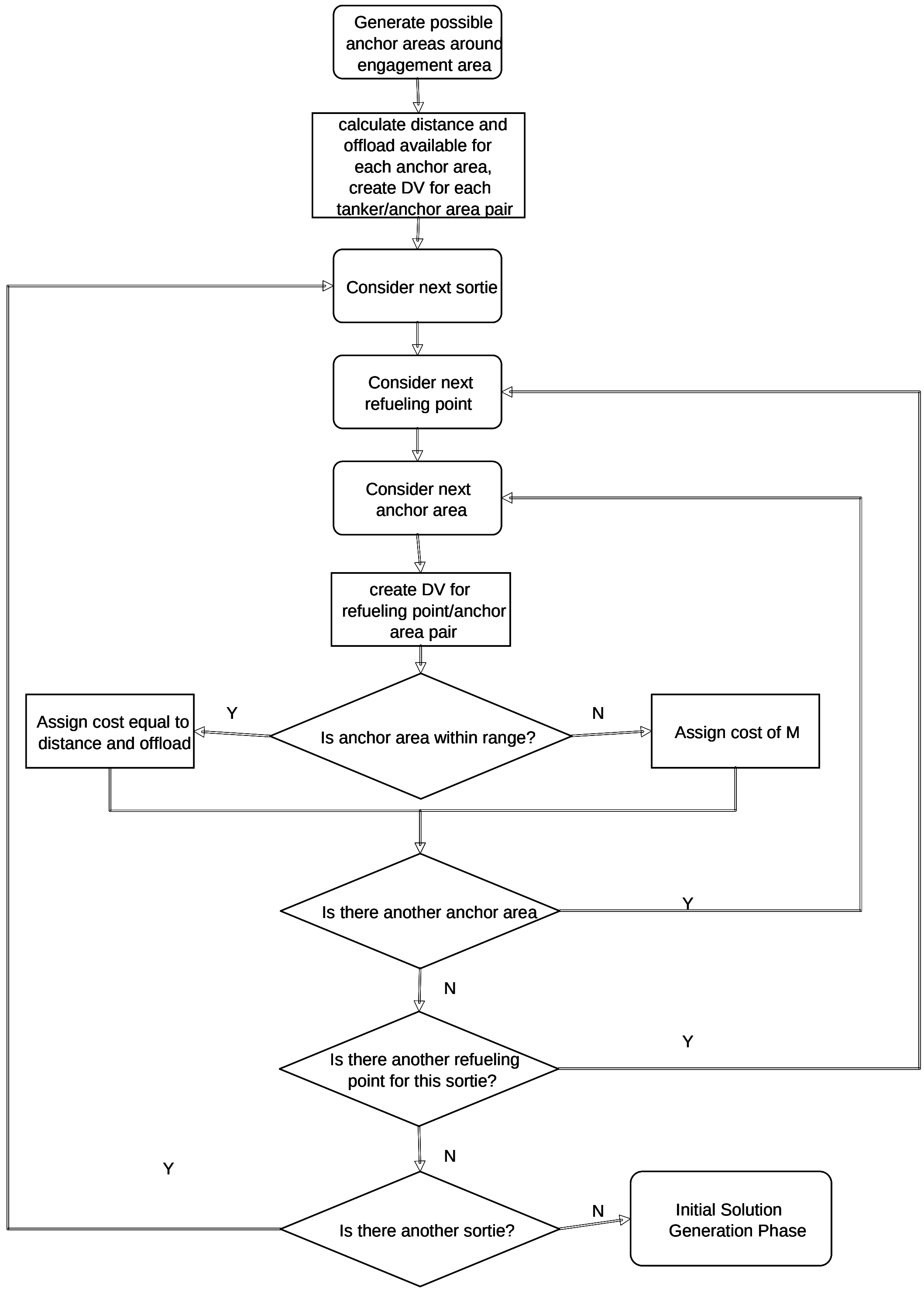

Figure 6: DV and Cost Generation Phase

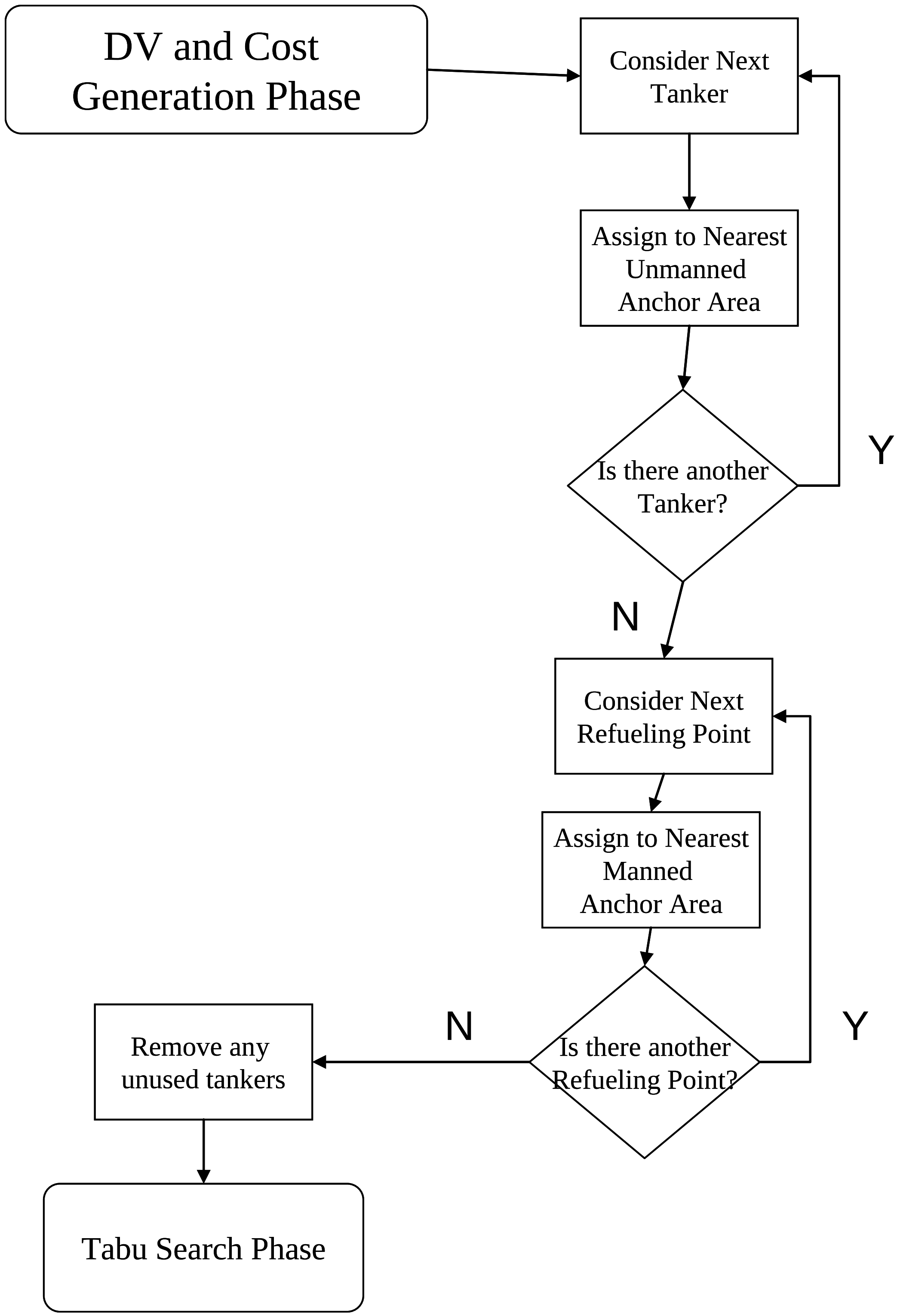

Figure 7: Initial Solution Generation Phase

3.8 Memory Usage

This research uses explicit memory, the same memory structure used for the TAP tool by Tekelioglu (2001). Explicit memory stores complete solutions visited during the search, stored in an array. These solutions are used for measuring cycling in the short-term. Objective function values are recorded in an additional array. The objective function values are used to check for long term cycling.

Tabu search also has a memory component called attributive memory. This memory records information about solution attributes that change in moving from one solution to another. This memory structure is not used in this research.

3.9 Visual Basic for Applications

Visual Basic for Applications (VBA) is the common scripting language created by Microsoft. It is included in all applications included in the Microsoft Office suite as well as many applications from other vendors. Using the VBA language allows the user to create structured programs directly within Microsoft Excel.

AMC desires an easy to use tool. Most members of the Air Force have a working familiarity with Excel and Excel based spreadsheets, including the TAP tool. Since VBA is the embedded scripting language, it is an obvious choice. Additionally, the aforementioned TAP tool was built using VBA. Finally, Excel's built in functions provide tools to analyze the results.

3.10 Measurement of Results

For completeness, it would be ideal to compare the results obtained with this tool with those obtained by a commercial or AMC application. However, no such model or program exists to draw a comparison. For this reason we compare the results of the model while varying both search and anchor area definition parameters. These results are also analyzed by a knowledgeable source for their reasonableness and usefulness. The next chapter reports the results of model testing.

IV. Results and Analysis

4.1 Introduction

An Excel-based tool was developed to input a number of sorties, an engagement area, and available tankers, and output a mission plan consisting of tanker and sortie assignments to anchor areas. Two sample employments were created for testing this new tool.

4.2 Caribbean Employment Scenario

4.2.1 Description of Caribbean Scenario

The first employment we test involves sorties departing from the south eastern United States and engaging over Cuba. This is a notional scenario designed to demonstrate the ability of the TET tool to select good anchor areas, and partition the tankers and sorties among them. It consists of 34 aircraft departing from 4 different bases located in Texas, Louisiana and Florida. The tankers available for this scenario are located in Mississippi. Table 3 provides a list of the 10 sorties and Table 4 provides the available tanker list. Tables 5 - 7 provide aircraft totals and sortie breakdown for the scenario. The scenario is designed to depict air operations in support of the air war, with priority being placed on strike missions. The following figure gives a visual representation of the Caribbean employment scenario.

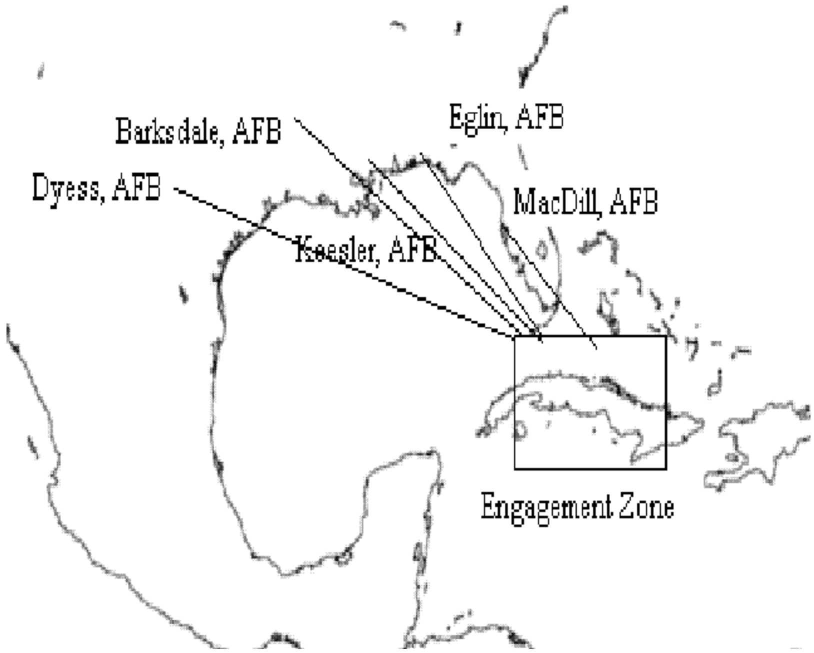

Figure 8: Caribbean Employment Scenario

Table 3: Sorties Caribbean Scenario

Sortie ID	Receiver Type	# of Receivers	Origin Base	Mission Type
Eagle1	F15	4	MacDill	CAP (Combat Air Patrol)
Eagle2	F15	4	MacDill	CAP (Combat Air Patrol)
Eagle3	F15	4	MacDill	CAP (Combat Air Patrol)
Falcon1	F16	4	Eglin	Strike
Falcon2	F16	4	Eglin	Strike
Falcon3	F16	4	Eglin	Strike
NightHawk1	F117	1	Barksdale	Strike
NightHawk2	F117	1	Barksdale	Strike
Hog1	A/OA10	4	Dyess	CAS (Close Air Support)
Hog2	A/OA10	4	Dyess	CAS (Close Air Support)

Table 4: Tankers Caribbean Scenario

Tanker Base	Number Available	Type
Keesler	10	KC-135E

Table 5 : Fighter Aircraft Totals Caribbean Scenario

Type	Number
A-10	8
F-14	0
F-15	12
F-15E	0
F-16	12
F-18	0
F-117	2
Total	**34**

Table 6: Tanker Aircraft Totals Caribbean Scenario

Type	Number
KC-135E	10
KC-135R	0
KC-10	0
Total	**10**

Table 7: Mission Type Breakdown Caribbean Scenario

A/C									
Missions	A-10	F-14	F-15	F-15E	F-16	F-18	F-117	**Total**	**Percent**
CAP			12					**12**	**35%**
CAS	8							**8**	**24%**
STRIKE					12		2	**14**	**41%**
Total	**8**	**0**	**12**	**0**	**12**	**0**	**2**	**34**	

For this employment, we assume there are 10 KC-135E tankers located at Keesler AFB. Tankers located at this base are capable of satisfying all the sorties' fuel requirements during the employment. None of the receivers have waypoints for this scenario, although our code allows the user to input up to two waypoints for each sortie.

4.2.2 Analysis of Caribbean Scenario

We apply TS with tenure of 7; skip number of 1; and large tabu restriction. Tables 5 and 6 show the initial feasible mission plan reached during the search routine, the search was run on a Gateway dual processor Pentium 4 and both processor speeds are 2.8 GHz. Tables 6 and 7 show the best mission plan along with the mean total computation time for the search. Complete computation time statistics are contained in Appendix B.

Table 8: Initial Feasible Mission Plan Caribbean Scenario

Sortie ID	Receiver Type	Num Aircraft	Sortie RP	RP Number	Origin Base	Anchor Area Assigned	Tanker Assigned
Eagle1	F15	4	1	1	MacDill	4	8
Eagle1	F15	4	2	2	MacDill	71	2
Eagle1	F15	4	3	3	MacDill	4	8
Eagle2	F15	4	1	4	MacDill	67	9
Eagle2	F15	4	2	5	MacDill	70	3
Eagle2	F15	4	3	6	MacDill	1	1
Eagle3	F15	4	1	7	MacDill	1	1
Eagle3	F15	4	2	8	MacDill	69	4
Eagle3	F15	4	3	9	MacDill	1	1
Falcon1	F16	4	1	10	Eglin	71	2
Falcon1	F16	4	2	11	Eglin	3	7
Falcon1	F16	4	3	12	Eglin	1	1
Falcon2	F16	4	1	13	Eglin	70	3
Falcon2	F16	4	2	14	Eglin	3	7
Falcon2	F16	4	3	15	Eglin	1	1
Falcon3	F16	4	1	16	Eglin	69	4
Falcon3	F16	4	2	17	Eglin	4	8
Falcon3	F16	4	3	18	Eglin	1	1
NightHawk1	F117	1	1	19	Barksdale	2	5
NightHawk1	F117	1	2	20	Barksdale	4	8
NightHawk1	F117	1	3	21	Barksdale	1	1
NightHawk2	F117	1	1	22	Barksdale	68	6
NightHawk2	F117	1	2	23	Barksdale	4	8
NightHawk2	F117	1	3	24	Barksdale	1	1
Hog1	A/OA10	4	1	25	Dyess	1	1
Hog1	A/OA10	4	2	26	Dyess	2	5
Hog1	A/OA10	4	3	27	Dyess	3	7
Hog2	A/OA10	4	1	28	Dyess	1	1
Hog2	A/OA10	4	2	29	Dyess	68	6
Hog2	A/OA10	4	3	30	Dyess	1	1

Table 9: Initial Feasible Mission Evaluation Caribbean Scenario

Total Tanker Distance (nm)	4383.267902
Total Fuel Offload (1000 lbs)	739.5812456
Number of Tankers Used	9
Mean Computation Time (s)	5.7212

Table 10: Best Mission Plan Caribbean Scenario

Sortie ID	Receiver Type	Num Aircraft	Sortie RP	RP Number	Origin Base	Anchor Area Assigned	Tanker Assigned
Eagle1	F15	4	1	1	MacDill	4	8
Eagle1	F15	4	2	2	MacDill	71	2
Eagle1	F15	4	3	3	MacDill	4	8
Eagle2	F15	4	1	4	MacDill	4	8
Eagle2	F15	4	2	5	MacDill	70	3
Eagle2	F15	4	3	6	MacDill	4	8
Eagle3	F15	4	1	7	MacDill	4	8
Eagle3	F15	4	2	8	MacDill	69	4
Eagle3	F15	4	3	9	MacDill	4	8
Falcon1	F16	4	1	10	Eglin	70	3
Falcon1	F16	4	2	11	Eglin	67	9
Falcon1	F16	4	3	12	Eglin	1	1
Falcon2	F16	4	1	13	Eglin	71	2
Falcon2	F16	4	2	14	Eglin	3	7
Falcon2	F16	4	3	15	Eglin	1	1
Falcon3	F16	4	1	16	Eglin	1	1
Falcon3	F16	4	2	17	Eglin	67	9
Falcon3	F16	4	3	18	Eglin	1	1
NightHawk1	F117	1	1	19	Barksdale	1	1
NightHawk1	F117	1	2	20	Barksdale	3	7
NightHawk1	F117	1	3	21	Barksdale	1	1
NightHawk2	F117	1	1	22	Barksdale	2	5
NightHawk2	F117	1	2	23	Barksdale	3	7
NightHawk2	F117	1	3	24	Barksdale	1	1
Hog1	A/OA10	4	1	25	Dyess	68	6
Hog1	A/OA10	4	2	26	Dyess	2	5
Hog1	A/OA10	4	3	27	Dyess	1	1
Hog2	A/OA10	4	1	28	Dyess	1	1
Hog2	A/OA10	4	2	29	Dyess	68	6
Hog2	A/OA10	4	3	30	Dyess	1	1

Table 11: Best Mission Evaluation Caribbean Scenario

Total Tanker Distance (nm)	4383.2679
Total Fuel Offload (1000 lbs)	735.0483
Number of Tankers Used	9
Mean Computation Time (s)	67.8972

Although the total tanker distance is the same for both solutions, this would not necessarily be true for all feasible solutions; using additional tankers would also be feasible and would increase the total tanker distance. Both mission plans utilize 9 tankers. This is consistent with the 4:1 ratio, that is the current rule of thumb for the number of tankers required. For every four aircraft that need tanker support, there needs to be one tanker in theater. There is gain from the best solution over the initial feasible solution. The total fuel offload is lowered by approximately 4000 lbs. Even though this is relatively small in terms of the scale of the model, saving 4000 lbs of fuel may be of high importance to AMC. Figure 9 displays the mission evaluations during the 100 iterations.

Table 12: Lower Bound Caribbean Scenario

Total Tanker Distance (nm)	3934.9296
Total Fuel Offload (1000 lbs)	727.7905
Number of Tankers Used	8

Tanker distance is within 11% of the calculated lower bound and total fuel offload is within 0.9% of the calculated lower bound. The lower bounds were achieved by relaxing the capacity constraints and solving the facility location subproblem.

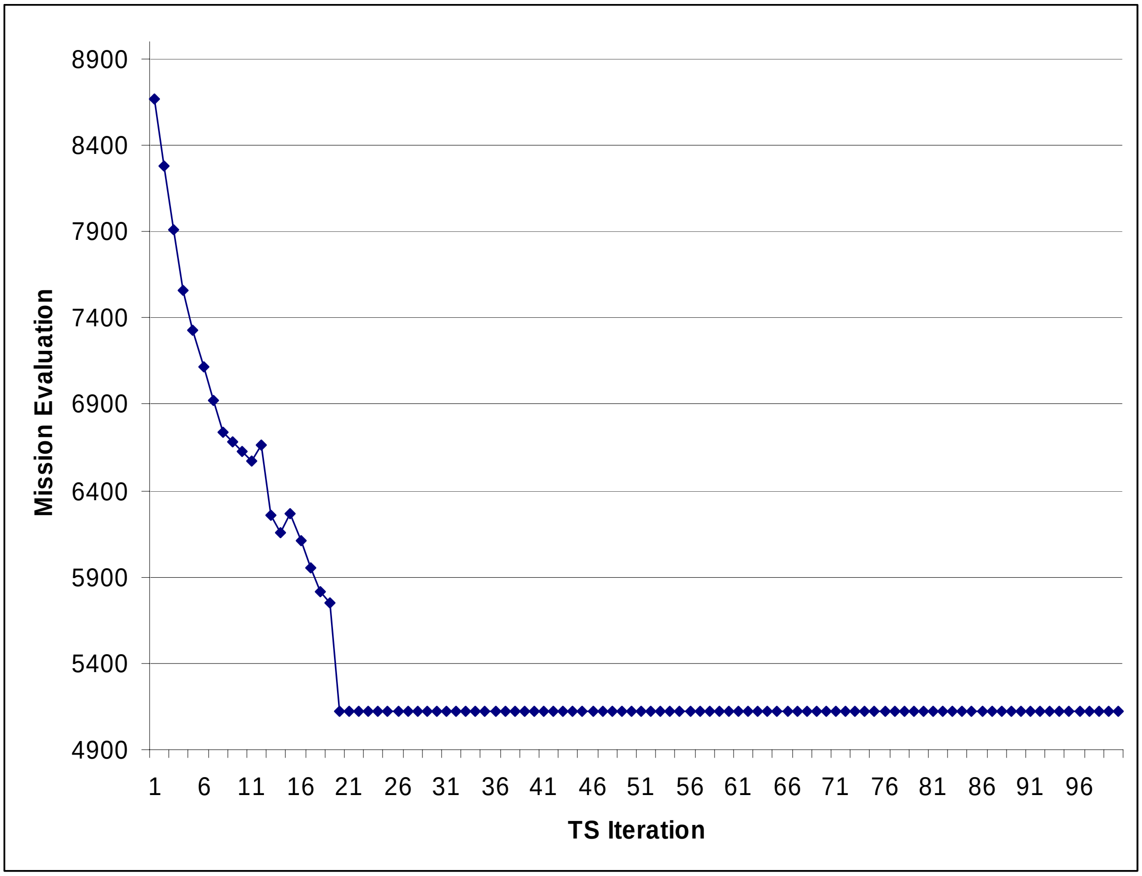

Figure 9: Caribbean Scenario TS Results

4.2.3 TS Tenure Comparison for Caribbean Scenario

We compare the results of different TS tenures based on 100 iterations. We test the following methods {Tenure, Skip Number, Restriction Size}: {7, 1, Large}, {9, 1, Large}, {11, 1, Large}. Figure 10 shows the mission evaluations during these 3 runs.

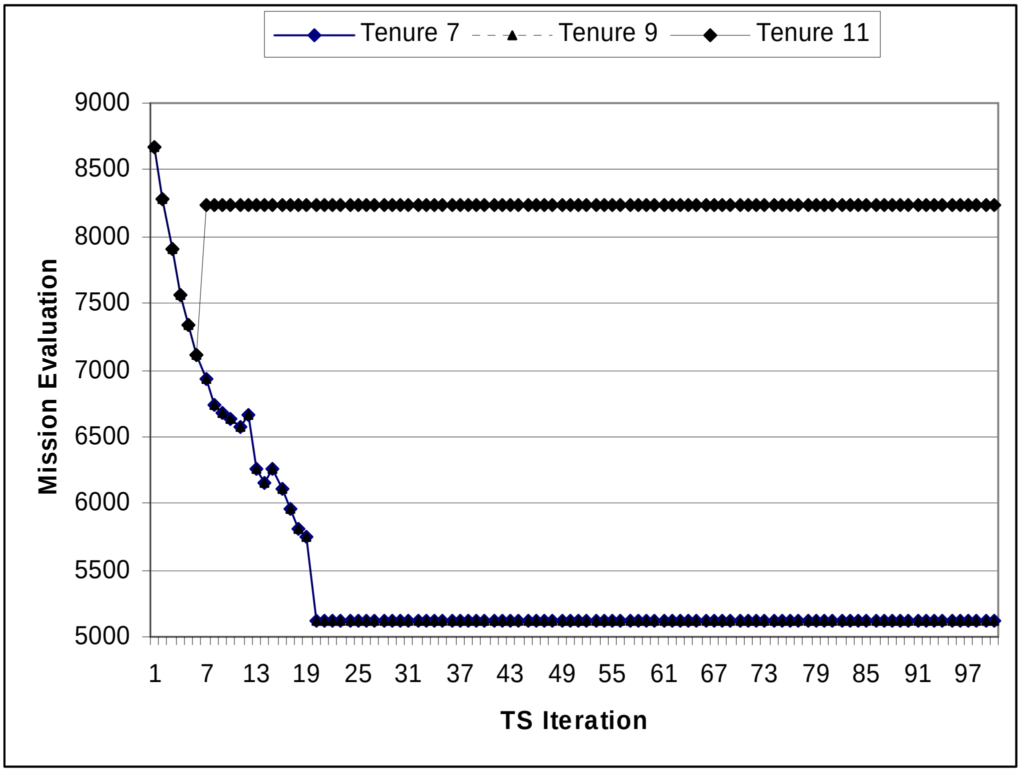

Figure 10: Tenure Comparison for Cuba Scenario

Tenure 7 and 9 reach many of the same solution values due to the structure of the scenario. There are multiple DVs that have the same costs and fuel requirements. Tenure 11 begins improving but then gets trapped in a local optimum region around iteration 12, and never finds a feasible solution.

4.2.4 TS Method Comparison for Caribbean Scenario

We again compare the results of different TS methods based on 100 iterations. We test the following methods using the same notation as before: {7, 1, Large}, {7, 2, Large}, {7, 1, Small}, {7, 2, Small}. Figure 11 shows the mission evaluations during these 4 runs.

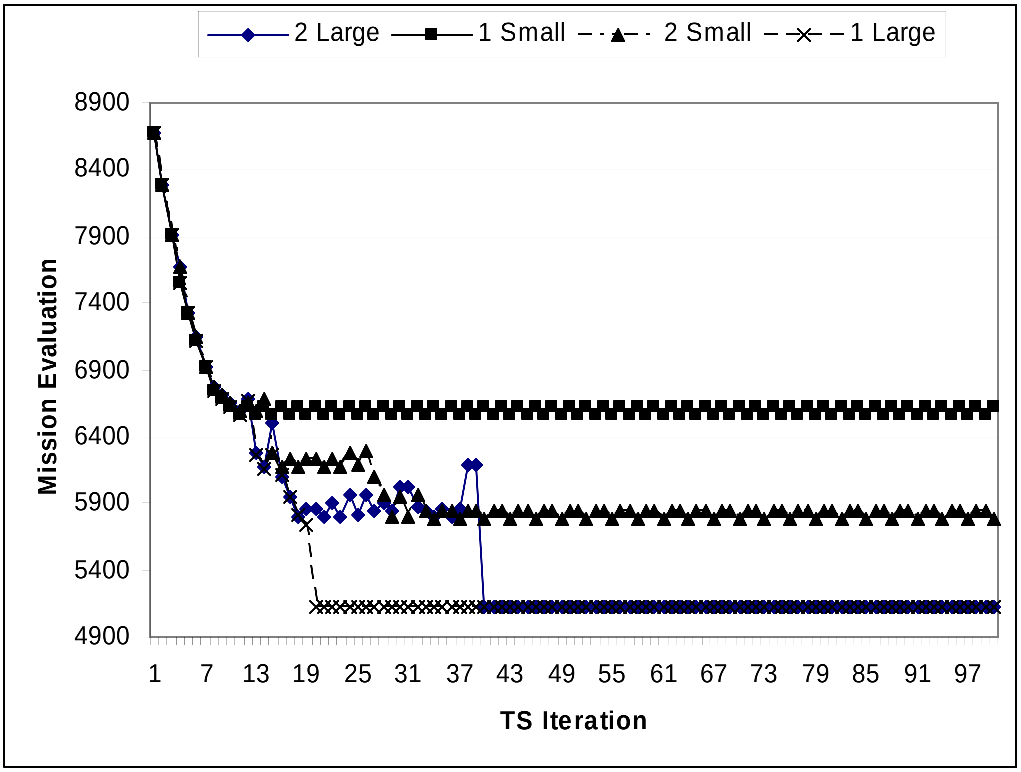

Figure 11: TS Method Comparison for Caribbean Scenario

The size of the tabu restriction has an effect on the search. When using the small restriction, the search falls into a chaotic attractor much sooner, and actually fails to find feasible solutions. The size of the candidate list speeds up the overall speed of the search and either has no effect or an improvement to the best solution found.

4.3 Middle East Employment Scenario

4.3.1 Description of Middle East Scenario

The second employment we test is modeled after Operation Iraqi Freedom. This is a notional scenario designed to demonstrate the ability of the TET tool to select good anchor areas, and partition the tankers and sorties among them. The scenario is designed to depict air operations in support of ground push, with priority being placed on combat air support. It consists of 244 aircraft departing from 7 different bases located in Turkey, Kuwait, Bahrain, Qatar, Jordan, and an aircraft carrier located in the Persian Gulf. The tankers available for this scenario are located in United Arab Emirates, Oman, Qatar, Saudi Arabia, and Turkey. Table 12 provides a subset of the 64 sorties (for full list see Appendix B) and Table 13 provides the available tanker list. Tables 14 - 16 provide aircraft totals and sortie breakdown for the scenario. The following figure depicts the Middle East employment scenario.

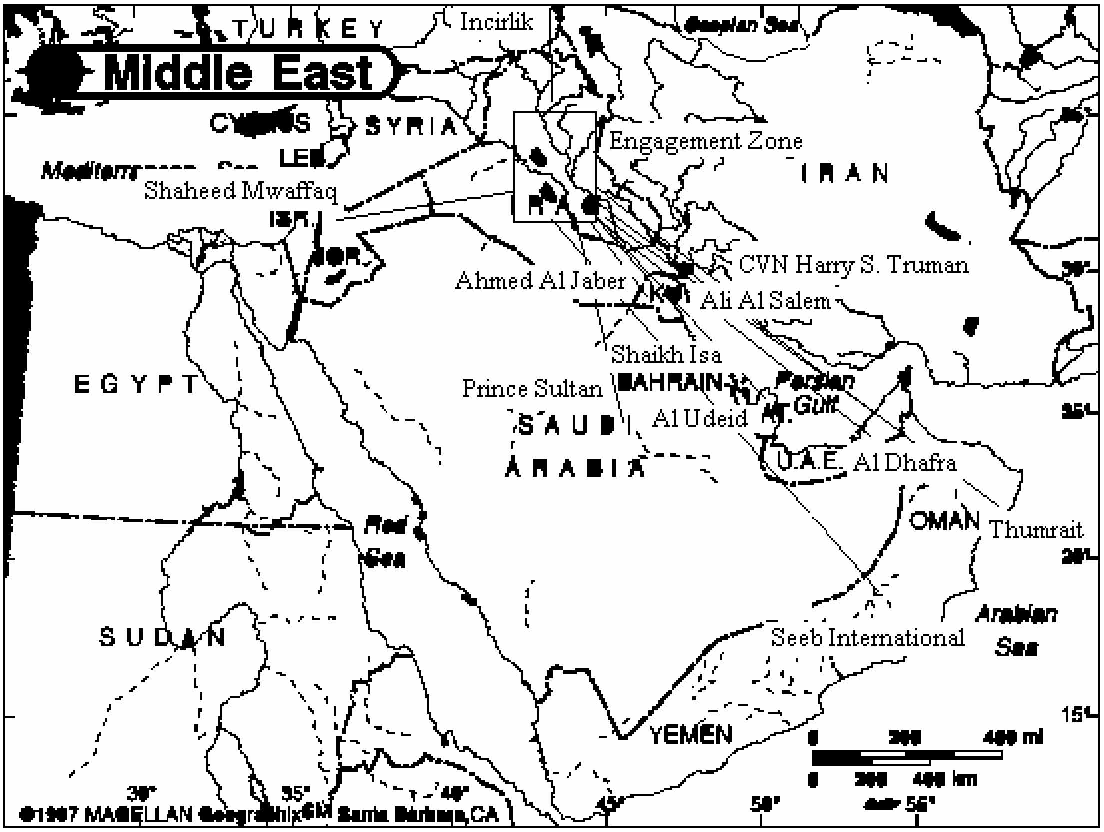

Figure 12: Middle East Employment Scenario

Table 13: Subsection of Sorties Middle East Scenario

Sortie ID	Receiver Type	# of Receivers	Origin Base	Mission Type
A10_1	A/OA10	4	Ahmed Al Jaber	CAS
F117_1	F117	2	Ali Al Salem	Strike
F14_3	F14	4	CVN Harry S. Truman	CAP
F15_3	F15	4	Ahmed Al Jaber	CAP
F15_4	F15	4	Shaikh Isa	CAP
F15_7	F15	4	Al Udeid	CAP
F15E_6	F15E	4	Al Udeid	Strike
F16_1	F16	4	Ahmed Al Jaber	CAS
F16_9	F16	4	Ali Al Salem	Strike
F16_19	F16	2	Shaikh Isa	CAS
F16_20	F16	4	Shaheed Mwaffaq	CAS
F16_31	F16	4	Incirlik	CAS
F18_1	F18	4	CVN Harry S. Truman	CAS

Table 14: Tankers Middle East Scenario

Tanker Base	Number Available	Type
Prince Sultan	6	KC-135E
Incirlik	12	KC-135R
Al Udeid	12	KC-135R
Thumrait	4	KC-10
Seeb International Airport	12	KC-135R
Al Dhafra	12	KC-135R
Al Dhafra	4	KC-10

Table 15: Fighter Aircraft totals Middle East Scenario

Type	Number
A-10	24
F-14	12
F-15	36
F-15E	24
F-16	120
F-18	24
F-117	4
Total	**244**

Table 16: Tanker Aircraft totals Middle East Scenario

Type	Number
KC-135E	6
KC-135R	48
KC-10	8
Total	**62**

Table 17: Mission Type Breakdown Middle East Scenario

A/C									
Missions	A-10	F-14	F-15	F-15E	F-16	F-18	F-117	**Total**	**Percent**
CAP		12	36					**48**	**20%**
CAS	24				96	24		**144**	**59%**
STRIKE				24	24		4	**52**	**21%**
Total	**24**	**12**	**36**	**24**	**120**	**24**	**4**	**244**	

4.3.2 Analysis of Middle East Scenario

On the initial run of this scenario, the TET tool failed to find a feasible solution. Upon further analysis it was determined that the defined engagement area was too small. Due to its small size, only 38 refueling tracts could be defined along its edges, which artificially limited the number of tankers that could be used. The engagement zone was then expanded to allow 63 refueling tracts to fit along the edge, which allows all available tankers to be used if required by demand. This could also have been achieved by defining two rings of available anchor areas at different altitudes. This modification, however, is left for future research.

Again we apply TS, this time with tenure of 7; skip number of 1; and large tabu restriction. Tables 17 and 18 show a subsection of the initial feasible mission plan reached during the search routine (for the complete solutions see Appendix B). The search was run on a Gateway dual processor Pentium 4 and both processor speeds are 2.8 GHz. Tables 19 and 20 show a subsection of the best mission plan along with the total computation time for the search. Again the complete solutions and computation time statistics are contained in Appendix B.

Table 18: Subsection of Initial Feasible Mission Plan Middle East Scenario

Sortie ID	Receiver Type	Num Aircraft	Sortie RP	RP Number	Origin Base	Anchor Area Assigned	Tanker Assigned
A10_1	A/OA10	4	1	1	AHMED AL JABER	31	1
F117_1	F117	2	1	19	ALI AL SALEM AB	30	2
F14_1	F14	4	1	25	Harry S. Truman	29	6
F15_1	F15	4	1	34	AHMED AL JABER	33	4
F15_4	F15	4	1	43	BAHRAIN INTL	28	19
F15_7	F15	4	1	52	Al Udeid	34	5
F15E_1	F15E	4	1	61	Al Udeid	36	23
F16_1	F16	4	1	79	AHMED AL JABER	33	4
F16_9	F16	4	1	103	ALI AL SALEM AB	39	29
F16_12	F16	4	1	112	BAHRAIN INTL	22	30
F16_20	F16	4	1	136	Shaheed Mwaffaq	44	44
F16_26	F16	4	1	154	INCIRLIK CDI	61	7
F18_1	F18	4	1	172	Harry S. Truman	21	35

Table 19: Initial Feasible Mission Evaluation Middle East Scenario

Total Tanker Distance (nm)	31900.309
Total Fuel Offload (1000 lbs)	4655.1612
Number of Tankers Used	60
Mean Computation Time (s)	125.5701

Table 20: Subsection of Best Mission Plan Middle East Scenario

Sortie ID	Receiver Type	Num Aircraft	Sortie RP	RP Number	Origin Base	Anchor Area Assigned	Tanker Assigned
A10_1	A/OA10	4	3	3	AHMED AL JABER	31	1
F117_1	F117	2	3	21	ALI AL SALEM AB	30	2
F14_1	F14	4	1	25	Harry S. Truman	29	6
F14_3	F14	4	3	33	Harry S. Truman	29	6
F15_1	F15	4	2	35	AHMED AL JABER	6	60
F15_4	F15	4	1	43	BAHRAIN INTL	28	19
F15_7	F15	4	1	52	Al Udeid	34	5
F15E_1	F15E	4	1	61	Al Udeid	36	23
F16_2	F16	4	2	83	AHMED AL JABER	10	54
F16_9	F16	4	1	103	ALI AL SALEM AB	39	29
F16_13	F16	4	2	116	BAHRAIN INTL	13	43
F16_25	F16	4	3	153	Shaheed Mwaffaq	46	46
F16_28	F16	4	1	160	INCIRLIK CDI	61	7
F16_28	F16	4	2	161	INCIRLIK CDI	17	39
F18_1	F18	4	1	172	Harry S. Truman	21	35

Table 21: Best Mission Evaluation Middle East Scenario

Total Tanker Distance (nm)	31900.309
Total Fuel Offload (1000 lbs)	4655.1612
Number of Tankers Used	60
Mean Computation Time (s)	1453.08

Both mission plans utilize 60 tankers. This is slightly better than the 4:1 rule of thumb ratio. For this instance the TS failed to improve on the solution generated by the construction heuristic. Although we would have liked to improve upon the initial solution, generating a feasible solution quickly to a large problem is desirable.

Table 22: Lower Bound Middle East Scenario

Total Tanker Distance (nm)	15523.028
Total Fuel Offload (1000 lbs)	4159.9994
Number of Tankers Used	33

Although tanker distance is over twice the value of the calculated lower bound, total fuel offload is within 12% of the calculated lower bound. The lower bounds were again achieved by relaxing the capacity constraints and solving the facility location subproblem. The disparity between the tanker distances is likely due to high number of sorties that are departing from each base. By removing the capacity constraint, the relaxation is able to meet the demand without utilizing as many tankers.

4.3.3 TS Performance and Method Comparison Middle East Scenario

We performed the same comparison of the results of different TS tenures based on 500 iterations, testing the same parameters as previously tested with the Caribbean Scenario. For this particular scenario, TS failed to improve on the initial feasible solution generated from the construction heuristic.

4.4 Conclusion

This thesis has demonstrated that the tanker employment problem can be modeled as a capacitated facility location problem with sole sourcing constraints. This tool considers tanker availability during a single snapshot of the employment. This tool provides a feasible partitioning of tankers and sorties to particular anchor areas in a short amount of time. The TET tool allows AMC to perform gross feasibility checks on how many missions a proposed number of tankers can support, or the number of tankers required to support a given number of missions. The TET tool also provides a good starting solution for the placement of refueling tracts, and should reduce overall planning time. The tool is also flexible by allowing the user to input new aircraft or airbases that are not currently contained in the database, as well as allowing an engagement zone to be defined by 4 latitude longitude pairs anywhere on the globe.

Almost every computer in the Air Force contains Microsoft Office with Excel; therefore, this tool is extremely portable. Also, the guided input menus and instructions built into the model increase the usability. Ease of use allows new personnel to use the tool with minimal training.

4.5 Problems

One drawback of the TET tool is that it does not allow a formation of receivers to be refueled by separate tankers. If a sortie consists of a four ship of F-16s, then each refueling point during this sortie requires a single tanker to refuel each of them. It is possible for a formation to split up to nearby tankers to refuel, which allows for more efficient use of tanker capacity. If for example, two nearby tankers have 10,000 lbs of fuel left each, and the four-ship requires 11,000 lbs, they cannot be assigned to either tanker. While if they could be split up, 3 of the planes could receive gas from one tanker, while the fourth gets refueled by the neighboring tanker and the mission remains feasible.

Another drawback of the TET tool is that it is deterministic rather than stochastic. All refueling points, including fuel required are calculated up front. This works well for planning purposes as long as the assumptions overestimate the amount of fuel required. The tool could be more realistic by taking into account the “fog of war” and generating fuel requirements based on some known probabilities, or probabilities derived from simulation results (Sun Tzu, 1988).

Also the tool does not account for unplanned refuelings. Unplanned refuelings include emergency refuelings, e.g., a plane has taken damage and as a result has lost more fuel than planned, and time sensitive intelligence reveals a new target that is beyond the strike range of current platform.

The final drawback of the TET tool is the lack of multiple search routines. It could be possible to implement multiple search techniques, which would promote a more diverse search. Diversifying the search methods could then yield better solutions, additionally upper and lower bounds could be iteratively improved which would allow the search to converge to the optimal solution. Currently the TET tool only provides upper bounds.

V. Contributions and Recommendations

5.1 Introduction

This chapter discusses the contributions produced by this research and suggestions for future research.

5.2 Research

Scheduling of aerial refueling operations is a complex task. This task is made more complex during the employment phase of operations due to smaller airspace and a high volume of demand with strict constraints. Because of this the time necessary to solve problems to optimality can be enormous even for small unrealistic examples. To combat this, heuristic methods can be used to provide very good solutions to the Tanker Employment Problem in a relatively short amount of time.

This research is the first in this particular area of refueling operations. The TET tool adapted some of the methodologies and assumptions from previous work in the deployment phase by Capehart (2000), Tekelioglu (2001), Wiley (2001), and Annaballi (2002).

For this research, a capacitated facility location problem with sole sourcing constraints formulation was used to model the employment phase of air refueling operations. A tabu search meta-heuristic was used in the spreadsheet model to generate tanker refueling results. It was found through testing that this formulation allowed the spreadsheet model to return good solutions in terms of minimizing the number of tankers used as well as maximizing offload available. It also did this in a short amount of time.

5.3 Contributions

This research has provided AMC with a tool capable of performing a quick look analysis on proposed employment operations. The TET tool can quickly answer questions such as:

1. Given system constraints/capacities, and information on receiver employment missions, where should anchor areas be located in order to maximize support for a particular phase of employment?
2. Given system constraints/capacities, and information on receiver employment missions, how many tankers will it take to meet receiver air refueling requirements?
3. Given system constraints/capacities, and a fixed number of tankers, how many receiver employment missions can be supported?

5.4 Future Research

This section provides a description of possible directions for future research that were encountered during this research.

Due to complexity of the problem, the affects of time were left out of this research. Every refueling point has a specific time window during which the refueling must take place. Additionally, tankers have a specific sortie duration, during which they are available. Also refueling points for an individual sortie have inherent precedence constraints, i.e. the initial refueling must occur a certain amount of time before the mid-mission refueling. There also may be artificially enforced precedence constraints, i.e. a high priority target needs to be struck first, therefore the refueling points associated with that sortie must come before those associated with the dependent sorties. Future research could implement a scheduling routine to the TET tool results to implement these constraints and check for feasibility. They could adjust the mission plan to reach a time feasible schedule.

The TET tool assumes instantaneous transfer of fuel. Actual fuel transfer is tanker configuration (boom/drogue) and receiver dependent. Future research could calculate the actual fuel transfer times based on a lookup table of the known transfer rates. Currently the TET tool calculates capacity for a tanker based on a static on station time. In the current model, if this time is lowered, flight time decreases which decreases tanker fuel burn, which in turn increases capacity. Implementing fuel transfer times and enforcing scheduling constraints would fix this current problem.

Additionally, the TET tool assumes that all input sorties will take place simultaneously. For this reason it only provides a snapshot of a single push during the employment. Introducing time to the model could allow the user to define specific dates for sorties, as well as initial tanker availability. The tanker availability could then be updated daily using calculations to capture maintenance downtime, crew constraints, and aircraft losses or increases. This broader model would allow for analysis of an entire employment rather than on a single day.

This research has proposed a new formulation for the Tanker Employment Problem. Viewing the problem as a capacitated facility location problem with sole sourcing constraints offers a unique perspective which allows for fast solutions to relaxed versions of the problem. For a given allocation of tankers to anchor areas, the problem can be viewed as a multiple knapsack problem, which has been previously researched, and algorithms are readily available to quickly solve these types of problems. Future research could include a Bender's partitioning style approach which utilizes multiple knapsack algorithms or heuristics for each given set of Y_{ij}*'s* which represent a particular set of tanker assignments. Another possible approach would be to use cross decomposition, which utilizes Lagrangean relaxation and Bender's partitioning to generate upper and lower bounds. One would have to be concerned about convergence due to the duality gap, since the problem is integer. Sweeney and Murphy decomposition is a form of Lagrangean relaxation that deals with this problem. Future research could combine these methods and apply them to the tanker employment problem.

The TET tool considers the fact that tankers fly at different altitudes and speeds when refueling and traveling to and from the anchor area. Although the TET tool does account for a change in speed and altitude while a tanker is refueling, it does not account for the fact that these altitudes and speeds are receiver dependent. Tankers match the speed of fighter aircraft during refueling and fly a particular buddy cruise speed. A calculation would be required to incorporate these speed changes, which would affect the amount of fuel burned, which would in turn affect the tanker's capacity.

Although the TET tool allows the user to input waypoints for each sortie, it does not allow for waypoints to be specified for egress routing. If a particular target was located in the SE corner of the engagement zone, and the strike aircraft's base was located to the NW, waypoints would allow the aircraft to exit the engagement zone on the nearest border to their final target. This would be more realistic than the current approach which forces the egress refueling to take place on one of the borders near the base of origin, possibly forcing the aircraft to cross the entire engagement zone to egress.

The current method for defining all possible anchor areas (refueling tracts) places them side by side bordering one another. Air Force doctrine dictates there is free space between the refueling tracks to prevent mid-air collisions or near misses. This could be accomplished by forcing a hard distance between the tracks, or allowing the tracks to also be defined at specific altitudes. By allowing tracks to be “stacked” it could be possible to utilize more tankers in a smaller airspace than the TET tool currently allows. Currently the TET tool does not prescribe a specific altitude for each refueling track, but uses a uniform altitude of 25,000 ft.

Our TET tool uses a penalized objective function to determine the mission evaluation during the search process. We consider the total distance traveled by all tankers, the total fuel offloaded, and the capacities of the tankers. Future research can allow the user to change the weights associated with these penalties to make them more representative of the actual goals of AMC.

Additionally, future research could improve the performance of the tabu search heuristic employed within the TET tool. Different TS methods were easily trapped into regions of local optimum or chaotic attractors. The use of better diversification strategies, possibly including a hashing function, may avoid this.

Other specific characteristics could further enhance the code. These include adding to the mission type definition to allow the user to specify a specific armament and flight characteristics including proposed airspeed and altitude throughout the flight. Wind characteristics at several locations around the globe could be added to allow for their effect on the fuel burn of both tankers and receivers. Including wind would also impact the time that refueling points would need to be serviced.

Another addition would be the ability to visualize the generated mission plan. A map function would be useful to see the actual anchor areas employed as well as tanker routes to service the anchor areas. This visualization would also be useful in quickly determining if any errors had been made during the modeling process.

Currently the TET tool only considers a single four sided static restricted operating zone that is always active when determining the tanker routes to anchor areas. This type of restriction is similar to those employed in UAV routing. Future work could include tying in the tanker cost calculations using the best route from *Route Builder* (Brown 2001) developed for the UAV Battlelab. Future work could also focus on applying time window restrictions to operating zones, defining more than one engagement zone, and allowing a broader range of shapes, to better model actual employment scenarios.

Although the TET tool has a reasonable computation time, it could be improved. If there are n refueling points, m anchor areas and k tankers, each iteration performs nm move evaluations, each of which must perform ($mk + nk + m + n + k$) calculations. That means the total number of calculations is $n \cdot m\{k(m+n+1)+m+n\}$. Java code could be written to import the input data, which the user inputs through an Excel interface. Java is platform independent and is object oriented. This object orientation could increase the manageability of the code and possibly decrease the computation time.

Finally, this model does not take into account tanker flight crew schedules. The number of flight crews would have a tremendous impact on operations. Future researchers could implement restrictions such as maximum flight hours per day, or crew rest restrictions. Scheduling flight crews is a difficult problem on its own. Previous AFIT research has combined tanker and aircrew scheduling for the deployment problem. It could be possible to adapt this research to build a complete tanker scheduling tool.

Appendix A. TET Tool VBA Coding

```
'Module:        Assign
'Author:        Lt. Jeffrey R. Miller, USAF AFIT/ENS
'Last Updated:  21 Mar 05
'Function:      This module contains the main portion of the Tanker Employment tool
'               The subroutines in this module are layed out in 3 sections.
'               Section I contains input data manipulation to prepare for the solutions
'               Section II builds an intitial solution and intitiates the search to find best
'               Tanker and sortie allocation
'               Section III outputs the solutions to Excel Spreadsheets for the User
Option Base 1
Dim TCost() As Double, RCost() As Double, Sik() As Double, Yik() As Integer, Xij() As Integer
Dim LB As Double, UB As Double, TabuList() As Integer, Value As Double, BestMove() As Variant
Dim LBValue As Double, BestYik() As Integer, BestXij() As Integer, NumFeas As Integer, Aspire As
Boolean
Dim NoImprove As Integer, UBold As Double, Delta As Double, CurrentVal As Double, SearchTime As
Single
Dim Done As Boolean
'***************************************************************************************
*******
'SECTION I
'***************************************************************************************
*******
'Subroutine: Tanker Costs
'this subroutine calculates the travel distance from tanker's beddown base to
'anchor area, taking into account flying around the engagement area

Sub TankerCosts()

Dim NWlat, NWlong, NElat, NElong, SWlat, SWlong, SElat, SElong, Alat, Along, Tlat, Tlong As Integer
Dim NWNE, NWSW, NESE, SWSE As Double
Worksheets("Engagement Area").Select
With Range("A2")
   NWlat = .Offset(1, 0)
   NWlong = .Offset(1, 1)
   NElat = .Offset(1, 2)
   NElong = .Offset(1, 3)
   SElat = .Offset(1, 6)
   SElong = .Offset(1, 7)
   SWlat = .Offset(1, 4)
   SWlong = .Offset(1, 5)
End With
With Range("H10")
   NWNE = .Offset(1, 0)
   NWSW = .Offset(2, 0)
   NESE = .Offset(3, 0)
   SWSE = .Offset(4, 0)
End With
Worksheets("Tankers").Select
With Range("L2")
   numtank = Range(.Offset(1, 0), .End(xlDown)).Rows.Count 'number tankers
End With
```

```
Worksheets("Engagement Area").Select
With Range("A7")
  numanch = Range(.Offset(1, 0), .End(xlDown)).Rows.Count 'number of anchor areas
End With
ReDim TCost(numanch, numtank) 'Tanker Cost Matrix
ReDim Yik(numanch, numtank) 'Tanker DV Matrix
'fill in the distance data
For i = 1 To numanch
  Pos = Worksheets("Engagement Area").Range("D7").Offset(i, 0).Value
  Alat = Worksheets("Engagement Area").Range("B7").Offset(i, 0).Value
  Along = Worksheets("Engagement Area").Range("C7").Offset(i, 0).Value
  For j = 1 To numtank
    Octnt = Worksheets("Tankers").Range("P2").Offset(j, 0)
    Tlat = Worksheets("Tankers").Range("N2").Offset(j, 0)
    Tlong = Worksheets("Tankers").Range("O2").Offset(j, 0)
    Select Case Octnt
    Case Is = 1
      Select Case Pos
      Case Is = "N" 'check for intersect
        If Intersect(Tlat, Tlong, Alat, Along, NWlat, NWlong, SWlat, SWlong) Then
          TCost(i, j) = GreatCircleDistance(Tlat, Tlong, NWlat, NWlong) + _
          GreatCircleDistance(NWlat, NWlong, Alat, Along)
          'if intersects then fly NW-straight
        Else 'else fly straight
          TCost(i, j) = GreatCircleDistance(Tlat, Tlong, Alat, Along)
        End If
      Case Is = "W" 'check for intersect
        If Intersect(Tlat, Tlong, Alat, Along, NWlat, NWlong, NElat, NElong) Then
          TCost(i, j) = GreatCircleDistance(Tlat, Tlong, NWlat, NWlong) + _
          GreatCircleDistance(NWlat, NWlong, Alat, Along)
          'if intersects then fly NW-straight
        Else 'else fly straight
          TCost(i, j) = GreatCircleDistance(Tlat, Tlong, Alat, Along)
        End If
      Case Is = "E" 'fly min (NW-NE-straight,NW-SW-SE-straight)
        TCost(i, j) = GreatCircleDistance(Tlat, Tlong, NWlat, NWlong) + _
        Min(NWNE + GreatCircleDistance(NElat, NElong, Alat, Along), NWSW + _
        SWSE + GreatCircleDistance(SElat, SElong, Alat, Along))
      Case Is = "S" 'fly min (NW-SW-straight,NW-NE-SE-straight)
        TCost(i, j) = GreatCircleDistance(Tlat, Tlong, NWlat, NWlong) + Min(NWSW + _
        GreatCircleDistance(SWlat, SWlong, Alat, Along), NWNE + NESE + _
        GreatCircleDistance(SElat, SElong, Alat, Along))
      End Select
    Case Is = 2
      Select Case Pos
      Case Is = "N" 'fly straight
        TCost(i, j) = GreatCircleDistance(Tlat, Tlong, Alat, Along)
      Case Is = "W" 'check for intersect
        If Intersect(Tlat, Tlong, Alat, Along, NWlat, NWlong, NElat, NElong) Then
          'if intersect fly min (NW-straight,NE-SE-SW-straight)
          TCost(i, j) = Min(GreatCircleDistance(Tlat, Tlong, NWlat, NWlong) + _
          GreatCircleDistance(NWlat, NWlong, Alat, Along), GreatCircleDistance(Tlat, Tlong, NElat,
NElong) + _
          NESE + SWSE + GreatCircleDistance(SWlat, SWlong, Alat, Along))
```

```
            Else 'else straight
                TCost(i, j) = GreatCircleDistance(Tlat, Tlong, Alat, Along)
            End If
        Case Is = "E" 'check for intersect
            If Intersect(Tlat, Tlong, Alat, Along, NWlat, NWlong, NElat, NElong) Then
            'if intersect fly min (NE-straight,NW-SW-SE-straight) else straight
                TCost(i, j) = Min(GreatCircleDistance(Tlat, Tlong, NElat, NElong) + _
                GreatCircleDistance(NElat, NElong, Alat, Along), GreatCircleDistance(Tlat, Tlong, NWlat,
NWlong) + _
                NWSW + SWSE + GreatCircleDistance(SElat, SElong, Alat, Along))
            Else
                TCost(i, j) = GreatCircleDistance(Tlat, Tlong, Alat, Along)
            End If
        Case Is = "S" 'Fly min route (NW-SW-straight) or (NE-SE-Straight)
            TCost(i, j) = Min(GreatCircleDistance(Tlat, Tlong, NWlat, NWlong) + _
            NWSW + GreatCircleDistance(SWlat, SWlong, Alat, Along), GreatCircleDistance(Tlat, Tlong,
NElat, NElong) + _
            NESE + GreatCircleDistance(SElat, SElong, Alat, Along))
        End Select
    Case Is = 3
        Select Case Pos
        Case Is = "N" 'check for intersect
            If Intersect(Tlat, Tlong, Alat, Along, SElat, SElong, NElat, NElong) Then 'if intersect then fly
NE-straight else straight
                TCost(i, j) = GreatCircleDistance(Tlat, Tlong, NElat, NElong) + GreatCircleDistance(NElat,
NElong, Alat, Along)
            Else
                TCost(i, j) = GreatCircleDistance(Tlat, Tlong, Alat, Along)
            End If
        Case Is = "E" 'check for intersect
            If Intersect(Tlat, Tlong, Alat, Along, NWlat, NWlong, NElat, NElong) Then 'if intersect then fly
NE-straight else straight
                TCost(i, j) = GreatCircleDistance(Tlat, Tlong, NElat, NElong) + GreatCircleDistance(NElat,
NElong, Alat, Along)
            Else
                TCost(i, j) = GreatCircleDistance(Tlat, Tlong, Alat, Along)
            End If
        Case Is = "S" 'fly min (NE-SE-straight,NE-NW-SW-straight)
            TCost(i, j) = GreatCircleDistance(Tlat, Tlong, NElat, NElong) + Min(NESE + _
            GreatCircleDistance(SElat, SElong, Alat, Along), NWNE + NWSW +
GreatCircleDistance(SWlat, SWlong, Alat, Along))
        Case Is = "W" 'fly min (NE-NW-straight,NE-SE-SW-Straight)
            TCost(i, j) = GreatCircleDistance(Tlat, Tlong, NElat, NElong) + Min(NWNE + _
            GreatCircleDistance(NWlat, NWlong, Alat, Along), NESE + SWSE +
GreatCircleDistance(SWlat, SWlong, Alat, Along))
        End Select
    Case Is = 4
        Select Case Pos
        Case Is = "N" 'check for intersect
            'if intersect fly NE-straight else straight
            If Intersect(Tlat, Tlong, Alat, Along, SElat, SElong, NElat, NElong) Then
                TCost(i, j) = GreatCircleDistance(Tlat, Tlong, NElat, NElong) + _
                GreatCircleDistance(NElat, NElong, Alat, Along)
            Else
```

```
            TCost(i, j) = GreatCircleDistance(Tlat, Tlong, Alat, Along)
         End If
      Case Is = "E" 'fly straight
         TCost(i, j) = GreatCircleDistance(Tlat, Tlong, Alat, Along)
      Case Is = "S" 'check for intersect
         'if intersect fly SE-straight else straight
         If Intersect(Tlat, Tlong, Alat, Along, SElat, SElong, NElat, NElong) Then
            TCost(i, j) = GreatCircleDistance(Tlat, Tlong, SElat, SElong) + _
            GreatCircleDistance(SElat, SElong, Alat, Along)
         Else
            TCost(i, j) = GreatCircleDistance(Tlat, Tlong, Alat, Along)
         End If
      Case Is = "W" 'fly min route (SE-SW-Straight) (NE-NW-Straight)
         TCost(i, j) = Min(GreatCircleDistance(Tlat, Tlong, SElat, SElong) + _
         SWSE + GreatCircleDistance(SWlat, SWlong, Alat, Along), GreatCircleDistance(Tlat, Tlong,
NElat, NElong) + _
         NWNE + GreatCircleDistance(NWlat, NWlong, Alat, Along))
      End Select
   Case Is = 5
      Select Case Pos
      Case Is = "N" 'fly min(SE-NE-straight,SE-SW-NW-straight)
         TCost(i, j) = GreatCircleDistance(Tlat, Tlong, SElat, SElong) + Min(NESE + _
         GreatCircleDistance(NElat, NElong, Alat, Along), SWSE + NWSW +
GreatCircleDistance(NWlat, NWlong, Alat, Along))
      Case Is = "E"  'check for intersect
         'if intersect then fly SE-straight else straight
         If Intersect(Tlat, Tlong, Alat, Along, SWlat, SWlong, SElat, SElong) Then
            TCost(i, j) = GreatCircleDistance(Tlat, Tlong, SElat, SElong) + _
            GreatCircleDistance(SElat, SElong, Alat, Along)
         Else
            TCost(i, j) = GreatCircleDistance(Tlat, Tlong, Alat, Along)
         End If
      Case Is = "S" 'check for intersect
         'if intersect then fly SE-straight else straight
         If Intersect(Tlat, Tlong, Alat, Along, NElat, NElong, SElat, SElong) Then
            TCost(i, j) = GreatCircleDistance(Tlat, Tlong, SElat, SElong) + _
            GreatCircleDistance(SElat, SElong, Alat, Along)
         Else
            TCost(i, j) = GreatCircleDistance(Tlat, Tlong, Alat, Along)
         End If
      Case Is = "W" 'fly min (SE-SW-straight, SE-NE-NW-straight)
         TCost(i, j) = GreatCircleDistance(Tlat, Tlong, SElat, SElong) + _
         Min(SWSE + GreatCircleDistance(SWlat, SWlong, Alat, Along), NESE + NWNE + _
         GreatCircleDistance(NWlat, NWlong, Alat, Along))
      End Select
   Case Is = 6
      Select Case Pos
      Case Is = "N" 'fly min (SW-NW-straight,SE-NE-straight)
         TCost(i, j) = Min(GreatCircleDistance(Tlat, Tlong, SWlat, SWlong) + NWSW + _
         GreatCircleDistance(NWlat, NWlong, Alat, Along), GreatCircleDistance(Tlat, Tlong, SElat,
SElong) + _
         NESE + GreatCircleDistance(NElat, NElong, Alat, Along))
      Case Is = "E" 'check for intersect
         'if intersect fly SE-straight else straight
```

```
        If Intersect(Tlat, Tlong, Alat, Along, SWlat, SWlong, SElat, SElong) Then
            TCost(i, j) = GreatCircleDistance(Tlat, Tlong, SElat, SElong) + _
            GreatCircleDistance(SElat, SElong, Alat, Along)
        Else
            TCost(i, j) = GreatCircleDistance(Tlat, Tlong, Alat, Along)
        End If
    Case Is = "S" 'fly straight
        TCost(i, j) = GreatCircleDistance(Tlat, Tlong, Alat, Along)
    Case Is = "W" 'check for intersect
        'if intersect fly SW-straight else straight
        If Intersect(Tlat, Tlong, Alat, Along, SWlat, SWlong, SElat, SElong) Then
            TCost(i, j) = GreatCircleDistance(Tlat, Tlong, SWlat, SWlong) + _
            GreatCircleDistance(SWlat, SWlong, Alat, Along)
        Else
            TCost(i, j) = GreatCircleDistance(Tlat, Tlong, Alat, Along)
        End If
    End Select
Case Is = 7
    Select Case Pos
    Case Is = "N" 'fly min (SW-NW-straight,SW-SE-NE-straight)
        TCost(i, j) = GreatCircleDistance(Tlat, Tlong, SWlat, SWlong) + _
        Min(NWSW + GreatCircleDistance(NWlat, NWlong, Alat, Along), SWSE + NESE + _
        GreatCircleDistance(NElat, NElong, Alat, Along))
    Case Is = "E" 'fly min (SW-SE-straight, SW-NW-NE-straight)
        TCost(i, j) = GreatCircleDistance(Tlat, Tlong, SWlat, SWlong) + _
        Min(SWSE + GreatCircleDistance(SElat, SElong, Alat, Along), NWSW + NWNE + _
        GreatCircleDistance(NElat, NElong, Alat, Along))
    Case Is = "S" 'check for intersect
        'if intersect then fly to SW-straight else straight
        If Intersect(Tlat, Tlong, Alat, Along, NWlat, NWlong, SWlat, SWlong) Then
            TCost(i, j) = GreatCircleDistance(Tlat, Tlong, SWlat, SWlong) + _
            GreatCircleDistance(SWlat, SWlong, Alat, Along)
        Else
            TCost(i, j) = GreatCircleDistance(Tlat, Tlong, Alat, Along)
        End If
    Case Is = "W" 'check for intersect
        'if intersect then fly to SW-straight else straight
        If Intersect(Tlat, Tlong, Alat, Along, SElat, SElong, SWlat, SWlong) Then
            TCost(i, j) = GreatCircleDistance(Tlat, Tlong, SWlat, SWlong) + _
            GreatCircleDistance(SWlat, SWlong, Alat, Along)
        Else
            TCost(i, j) = GreatCircleDistance(Tlat, Tlong, Alat, Along)
        End If
    End Select
Case Is = 8
    Select Case Pos
    Case Is = "N" 'check for intersect
        'if intersect fly NW-straight else straight
        If Intersect(Tlat, Tlong, Alat, Along, NWlat, NWlong, SWlat, SWlong) Then
            TCost(i, j) = GreatCircleDistance(Tlat, Tlong, NWlat, NWlong) + _
            GreatCircleDistance(NWlat, NWlong, Alat, Along)
        Else
            TCost(i, j) = GreatCircleDistance(Tlat, Tlong, Alat, Along)
        End If
```

```
        Case Is = "E" 'fly min (NW-NE-straight, SW-SE-Straight)
          TCost(i, j) = Min(GreatCircleDistance(Tlat, Tlong, NWlat, NWlong) + _
          NWNE + GreatCircleDistance(NElat, NElong, Alat, Along), GreatCircleDistance(Tlat, Tlong,
SWlat, SWlong) + _
          SWSE + GreatCircleDistance(SElat, SElong, Alat, Along))
        Case Is = "S" 'check for intersect
          'if intersect fly SW-straight else fly straight
          If Intersect(Tlat, Tlong, Alat, Along, NWlat, NWlong, SWlat, SWlong) Then
            TCost(i, j) = GreatCircleDistance(Tlat, Tlong, SWlat, SWlong) + _
            GreatCircleDistance(SWlat, SWlong, Alat, Along)
          Else
            TCost(i, j) = GreatCircleDistance(Tlat, Tlong, Alat, Along)
          End If
        Case Is = "W" 'fly straight
          TCost(i, j) = GreatCircleDistance(Tlat, Tlong, Alat, Along)
        End Select
      End Select
    Next j
Next i
End Sub
'Subroutine: Tanker List
'This subroutine creates the list of available tankers from the inputs
'This list is used to build tanker costs as well as assign particular tankers
'to specific anchor areas
'provides numbering system for reference overall as well
'as index number for particular base
Sub CTList()

Worksheets("Tankers").Select
With Range("A2")
    'loop through each base
    ofst = 1 'offset for tanker resource
    offst = 1 'offset for tanker numbered list
    Do
      If .Offset(ofst, 1) > 0 Then 'tankers available
        Index = 1
        Name = .Offset(ofst, 0) 'Base Name
        TType = .Offset(ofst, 2) 'Tanker type
        For i = 1 To .Offset(ofst, 1)
          .Offset(offst, 9) = Name
          .Offset(offst, 10) = TType
          .Offset(offst, 11) = offst
          .Offset(offst, 12) = Index
          .Offset(offst, 13).Formula = "=VLOOKUP(RC[-4],Bases,4,False)" 'Latitude
          .Offset(offst, 14) = "=VLOOKUP(RC[-5],Bases,5,False)" 'Longitude
          Call TOctant(.Offset(offst, 13), .Offset(offst, 14), offst) 'Orientation to Engagement Area
          .Offset(offst, 16).Formula = "=VLOOKUP(RC[-6],tnktypes,12)"
          offst = offst + 1
          Index = Index + 1
        Next i
      Else: Exit Do
      End If
      ofst = ofst + 1
    Loop
```

```
End With
End Sub
'Subroutine: TOctant
'This subroutine defines a tanker airbases position relative to the engagement area
' The surrounding area is divided into 8 sections. Beginning with the NW corner
' proceeding in a clockwise direction as follows
' 1 - NW corner, 2 - N, 3 - NE corner, 4 - E, 5 - SE corner, 6 - S, 7 - SW corner, 8 - W
Sub TOctant(BLat, BLong, offst)
    Worksheets("Engagement Area").Select
    With Range("A2")
        NWlat = .Offset(1, 0)
        NWlong = .Offset(1, 1)
        NElat = .Offset(1, 2)
        NElong = .Offset(1, 3)
        SElat = .Offset(1, 6)
        SElong = .Offset(1, 7)
        SWlat = .Offset(1, 4)
        SWlong = .Offset(1, 5)
    End With
    Worksheets("Tankers").Select
    'following line of logic partitions the orientation into 8 areas
    'counting clockwise from NW to W
    With Range("P2")
        If BLat >= NWlat And BLong >= NWlong Then
            .Offset(offst, 0) = 1
        End If
        If BLat >= Min(NWlat, NElat) And BLong < NWlong And BLong >= NElong Then
            .Offset(offst, 0) = 2
        End If
        If BLat >= NElat And BLong < NElong Then
            .Offset(offst, 0) = 3
        End If
        If BLat < NElat And BLat >= SElat And BLong < Min(NElong, SElong) Then
            .Offset(offst, 0) = 4
        End If
        If BLat < SElat And BLong < SElong Then
            .Offset(offst, 0) = 5
        End If
        If BLat < Min(SElat, SWlat) And BLong < SWlong And BLong >= SElong Then
            .Offset(offst, 0) = 6
        End If
        If BLat < SWlat And BLong >= SWlong Then
            .Offset(offst, 0) = 7
        End If
        If BLat < NWlat And BLat >= SWlat And BLong >= Min(NWlong, SWlong) Then
            .Offset(offst, 0) = 8
        End If
    End With
    End Sub
    'Subroutine: ROctant
    'This subroutine defines a tanker airbases position relative to the engagement area
    ' The surrounding area is divided into 8 sections. Beginning with the NW corner
    ' proceeding in a clockwise direction as follows
    ' 1 - NW corner, 2 - N, 3 - NE corner, 4 - E, 5 - SE corner, 6 - S, 7 - SW corner, 8 - W
```

```
    Sub ROctant(BLat, BLong, offst)
    Worksheets("Engagement Area").Select
    With Range("A2")
        NWlat = .Offset(1, 0)
        NWlong = .Offset(1, 1)
        NElat = .Offset(1, 2)
        NElong = .Offset(1, 3)
        SElat = .Offset(1, 6)
        SElong = .Offset(1, 7)
        SWlat = .Offset(1, 4)
        SWlong = .Offset(1, 5)
    End With
    Worksheets("RPPoints").Select
    'following line of logic partitions the orientation into 8 areas
    'counting clockwise from NW to W
    With Range("J1")
        If BLat >= NWlat And BLong >= NWlong Then
            .Offset(offst, 0) = 1
            Exit Sub
        End If
        If BLat >= Min(NWlat, NElat) And BLong < NWlong And BLong >= NElong Then
            .Offset(offst, 0) = 2
            Exit Sub
        End If
        If BLat >= NElat And BLong < NElong Then
            .Offset(offst, 0) = 3
            Exit Sub
        End If
        If BLat < NElat And BLat >= SElat And BLong < Min(NElong, SElong) Then
            .Offset(offst, 0) = 4
            Exit Sub
        End If
        If BLat < SElat And BLong < SElong Then
            .Offset(offst, 0) = 5
            Exit Sub
        End If
        If BLat < Min(SElat, SWlat) And BLong < SWlong And BLong >= SElong Then
            .Offset(offst, 0) = 6
            Exit Sub
        End If
        If BLat < SWlat And BLong >= SWlong Then
            .Offset(offst, 0) = 7
            Exit Sub
        End If
        If BLat < NWlat And BLat >= SWlat And BLong >= Min(NWlong, SWlong) Then
            .Offset(offst, 0) = 8
            Exit Sub
        End If
    End With
End Sub
Sub RPlist()
'This subroutine creates a list of all necessary refueling points in order
'to support the input sorties Assumption is that every sortie requires 3 refuelings
'an initial ingress refueling made at the border of the anchor area, a mid mission
```

```
'refueling, and an egress refueling.
'If scenario is such that an additional refueling is required because aircraft
'are stationed out of takeoff fuel range, this subroutine would need to be modified
Worksheets("Inputs").Select
With Range("A1")
   Numsorties = Range(.Offset(1, 0), .End(xlDown)).Rows.Count
End With
NumRP = Numsorties * 3
   For i = 1 To Numsorties
   Mtype = Worksheets("inputs").Range("E1").Offset(i, 0)

      For j = 1 To 3
         'Sortie ID
         Worksheets("RPPoints").Range("A1").Offset(i * 3 - 3 + j, 0).Value = _
         Worksheets("Inputs").Range("A1").Offset(i, 0).Value
         'Recevier Type
         Worksheets("RPPoints").Range("A1").Offset(i * 3 - 3 + j, 1).Value = _
         Worksheets("Inputs").Range("A1").Offset(i, 1).Value
         'Number in formation
         NumForm = Worksheets("Inputs").Range("A1").Offset(i, 2).Value
         Worksheets("RPPoints").Range("A1").Offset(i * 3 - 3 + j, 2).Value = NumForm
         'Base ICAO
         Worksheets("RPPoints").Range("A1").Offset(i * 3 - 3 + j, 6).Value = _
         Worksheets("Inputs").Range("A1").Offset(i, 3).Value
         'Base Lat
         Worksheets("RPPoints").Range("A1").Offset(i * 3 - 3 + j, 7).Formula = "=VLOOKUP(RC[-
1],Bases,4,False)" 'Latitude
         'Base Long
         Worksheets("RPPoints").Range("A1").Offset(i * 3 - 3 + j, 8) = "=VLOOKUP(RC[-
2],Bases,5,False)" 'Longitude
         'RP Number
         Worksheets("RPPoints").Range("A1").Offset(i * 3 - 3 + j, 4).Value = i * 3 - 3 + j
         'Sortie RP
         Worksheets("RPPoints").Range("A1").Offset(i * 3 - 3 + j, 3).Value = j
         'Octant
         Call ROctant(Worksheets("RPPoints").Range("A1").Offset(i * 3 - 3 + j, 7), _
         Worksheets("RPPoints").Range("A1").Offset(i * 3 - 3 + j, 8), i * 3 - 3 + j)
         'Max Offload Required
         FullLoad = FindFuel(Worksheets("Inputs").Range("A1").Offset(i, 1).Value)
         Worksheets("RPPoints").Range("F1").Offset(i * 3 - 3 + j, 0) = FullLoad * NumForm 'Max needed
for sortie
         'Aircraft specific Data
         'TAS
         Worksheets("RPPoints").Range("K1").Offset(i * 3 - 3 + j, 0).Formula = "=VLOOKUP(RC[-
9],performance,2,False)"
         'MaxFuel
         Worksheets("RPPoints").Range("K1").Offset(i * 3 - 3 + j, 1).Formula = "=VLOOKUP(RC[-
10],performance,4,False)"
         'Reserve Fuel
         Worksheets("RPPoints").Range("K1").Offset(i * 3 - 3 + j, 2).Formula = "=VLOOKUP(RC[-
11],performance,5,False)"
         'Min Weight
         Worksheets("RPPoints").Range("K1").Offset(i * 3 - 3 + j, 3).Formula = "=VLOOKUP(RC[-
12],performance,8,False)"
```

```
        'Armament
        Worksheets("RPPoints").Range("K1").Offset(i * 3 - 3 + j, 4).Formula =
Worksheets("Inputs").Range("F1").Offset(i, 0)
        'Climb Fuel
        Worksheets("RPPoints").Range("K1").Offset(i * 3 - 3 + j, 5).Formula = "=VLOOKUP(RC[-
14],performance,6,False)"
        'C1
        Worksheets("RPPoints").Range("K1").Offset(i * 3 - 3 + j, 6).Formula = "=VLOOKUP(RC[-
15],performance,10,False)"
        'c2
        Worksheets("RPPoints").Range("K1").Offset(i * 3 - 3 + j, 7).Formula = "=VLOOKUP(RC[-
16],performance,11,False)"
        'c3
        Worksheets("RPPoints").Range("K1").Offset(i * 3 - 3 + j, 8).Formula = "=VLOOKUP(RC[-
17],performance,12,False)"
        'c4
        Worksheets("RPPoints").Range("K1").Offset(i * 3 - 3 + j, 9).Formula = "=VLOOKUP(RC[-
18],performance,13,False)"
      Next j
   Next i
End Sub
Sub RCosts()
'This subroutine defines the costs for a particular receiver group refueling at
'a particular anchor area. These costs are stored in array
'Cij -cost for anchor area i refueling receiver j
Worksheets("RPPoints").Select
With Range("A1")
   NumRPS = Range(.Offset(1, 0), .End(xlDown)).Rows.Count
End With
Worksheets("Engagement Area").Select
With Range("A7")
   NumAnchs = Range(.Offset(1, 0), .End(xlDown)).Rows.Count
End With
ReDim RCost(NumAnchs, NumRPS) 'Refueling point cost matrix
ReDim Xij(NumAnchs, NumRPS) 'RP DV matrix
BigM = 10000 'Cost for refueling at an unreachable anchor area
For i = 1 To NumRPS
   'Get data about refueling point
   BLat = Worksheets("RPPoints").Range("H1").Offset(i, 0) 'origin base latitude
   BLong = Worksheets("RPPoints").Range("H1").Offset(i, 1) 'origin base longitude
   NumArcft = Worksheets("RPPoints").Range("H1").Offset(i, 2) 'number aircraft in formation
   TAS = Worksheets("RPPoints").Range("H1").Offset(i, 3) 'True Air Speed
   FC = Worksheets("RPPoints").Range("H1").Offset(i, 4) 'Fuel Capacity
   RF = Worksheets("RPPoints").Range("H1").Offset(i, 5) 'Reserve Fuel
   minwt = Worksheets("RPPoints").Range("H1").Offset(i, 6) 'min takeoff weight
   arm = Worksheets("RPPoints").Range("H1").Offset(i, 7) 'Armament weight
   climbf = Worksheets("RPPoints").Range("H1").Offset(i, 8) 'Climb Fuel
   c1 = Worksheets("RPPoints").Range("H1").Offset(i, 9)
   c2 = Worksheets("RPPoints").Range("H1").Offset(i, 10)
   c3 = Worksheets("RPPoints").Range("H1").Offset(i, 11)
   c4 = Worksheets("RPPoints").Range("H1").Offset(i, 12)
   RPNum = Worksheets("RPPoints").Range("D1").Offset(i, 0) 'initial mid or egress
   For j = 1 To NumAnchs
   Alat = Worksheets("Engagement Area").Range("B7").Offset(j, 0)
```

```
    Along = Worksheets("Engagement Area").Range("B7").Offset(j, 1)
        Select Case RPNum
            Case Is = 1  'initial refuel
                If InRange(i, j) Then  'find out if sortie can reach refueling point on initial gas
                    RCost(j, i) = recflburn(GreatCircleDistance(BLat, BLong, Alat, Along), _
                    TAS, FC, RF, minwt, arm, climbf, c1, c2, c3, c4) * NumArcft
                Else
                    RCost(j, i) = BigM
                End If
            Case Is = 3 'egress refuel
                If InRange(i, j) Then 'make sure return trip is feasible
                   RCost(j, i) = recflburn(GreatCircleDistance(BLat, BLong, Alat, Along), _
                    TAS, FC, RF, minwt, 0, 0, c1, c2, c3, c4) * NumArcft
                Else
                    RCost(j, i) = BigM
                End If
            Case Else 'mid mission
                RCost(j, i) = Worksheets("RPPoints").Range("F1").Offset(i, 0) / 2 'half refuel from reserve to
max load
        End Select
    Next j
Next i
End Sub
Sub Capacities()
'This sub routine defines capacities for anchor areas if serviced by a particular tanker
Worksheets("Tankers").Select
With Range("L2")
    numtank = Range(.Offset(1, 0), .End(xlDown)).Rows.Count 'number tankers
End With
Worksheets("Engagement Area").Select
With Range("A7")
    numanch = Range(.Offset(1, 0), .End(xlDown)).Rows.Count 'number of anchor areas
End With
ReDim Sik(numanch, numtank) 'Tanker Cost Matrix
tos = Worksheets("Miscellaneous").Range("E9").Value 'time on station
bcrate = Worksheets("Miscellaneous").Range("E10").Value 'buddy cruise speed
ierate = Worksheets("Aircraft Info").Range("B21").Value 'rate they get to the anchor area
FuelCap = Worksheets("Aircraft Info").Range("D21").Value 'fuel capacity
Res = Worksheets("Aircraft Info").Range("E21").Value 'reserve
minwt = Worksheets("Aircraft Info").Range("G21").Value 'min wt empty
cargo = Worksheets("Aircraft Info").Range("H21").Value 'cargo
climbf = Worksheets("Aircraft Info").Range("F21").Value 'climb fuel
c1 = Worksheets("Miscellaneous").Range("C17").Value  'fuel flow constants for KC-135R
c2 = Worksheets("Miscellaneous").Range("C17").Offset(0, 1)
c3 = Worksheets("Miscellaneous").Range("C17").Offset(0, 2)
c4 = Worksheets("Miscellaneous").Range("C17").Offset(0, 3)
c5 = Worksheets("Miscellaneous").Range("C17").Offset(0, 4)
c6 = Worksheets("Miscellaneous").Range("C17").Offset(0, 5)
c7 = Worksheets("Miscellaneous").Range("C17").Offset(0, 6)
For i = 1 To numanch
    For j = 1 To numtank
        sclfactor = Worksheets("Tankers").Range("Q3").Offset(j, 0)
        fuelburned = tnkburn(TCost(i, j), 4, bcrate, ierate, FuelCap, Res, minwt, cargo, _
        climbf, c1, c2, c3, c4, c5, c6, c7) 'fuel burned during flight
```

```
        Sik(i, j) = (FuelCap - fuelburned) * sclfactor 'available to offload
    Next j
Next i
'Loop through and determine available offload based on tanker distances
'if distance is too far, then assign offload available of 0.

End Sub



'*************************************************************************************
*******
'Section II.
'*************************************************************************************
*******
'This is the solution generation and search phase



Sub Initial()
Dim manned() As Boolean, Caps() As Double
'This is the initial solution generator
'Relax the capacity constraints to build an initial "super optimal" solution
'*********************************************************
'Manipulate input data to matrices and define costs

tmr = Timer
Call DefTracks
Call CTList
Call TankerCosts
Call RPlist
Call RCosts
Call Capacities
elapsed = Timer - tmr
Debug.Print elapsed
'**********************************************************
ReDim manned(UBound(Yik))
ReDim Caps(UBound(Sik))
'assign all tankers to best anchor area not already manned
For k = 1 To UBound(Yik, 2)
    Maximum = 0
    For i = 1 To UBound(Yik)
        If Sik(i, k) > Maximum And manned(i) = False Then
            Maximum = Sik(i, k)
            Index = i
        End If
    Next i
    Yik(Index, k) = 1
    manned(Index) = True
    Caps(Index) = Sik(Index, k)
Next k
For i = 1 To UBound(Caps)
    Workbooks("overnight.xls").Worksheets("Sheet1").Range("A1").Offset(i, 0) = Caps(i)
    For k = 1 To UBound(Yik, 2)
```

```
        If Yik(i, k) = 1 Then
            Workbooks("overnight.xls").Worksheets("Sheet1").Range("B1").Offset(i, 0) = k
        End If
    Next k
Next i
For j = 1 To UBound(Xij, 2)
    minimum = 10000
    For i = 1 To UBound(Xij)
        If RCost(i, j) < minimum And manned(i) = True And RCost(i, j) < Caps(i) Then
            minimum = RCost(i, j)
            Index = i
        End If
    Next i
    Xij(Index, j) = 1
    Caps(Index) = Caps(Index) - RCost(Index, j)
    If Caps(Index) < 0 Then
        Debug.Print Index; Caps(Index)
    End If
Next j
'remove any unused tankers
For i = 1 To UBound(Yik)
    If rowsum(i, Xij) < 1 Then
        For j = 1 To UBound(Yik, 2)
            Yik(i, j) = 0
            manned(i) = False
        Next j
    End If
Next i
For i = 1 To UBound(Caps)
    Workbooks("overnight.xls").Worksheets("Sheet1").Range("C1").Offset(i, 0) = Caps(i)
Next i
For j = 1 To UBound(Xij, 2)
    mini = 100000
    For i = 1 To UBound(Xij)
        If RCost(i, j) < mini Then
            mini = RCost(i, j)
        End If
    Next i
    Workbooks("overnight.xls").Worksheets("Sheet1").Range("D1").Offset(j, 0) = mini
Next j

End Sub
Sub TabuSearch()
Dim Tenure As Integer, Skip As Integer, Reactive As Boolean, LargeRestriction As Boolean
ReDim BestMove(5) 'oldi newi j value
ReDim BestYik(UBound(Yik), UBound(Yik, 2))
ReDim BestXij(UBound(Xij), UBound(Xij, 2))
'This is the search routine
'only evaluate neighborhoods that include variables that make sense.
'no need to evaluate a solution that assigns a customer to an unreachable facility
' or a solution that assigns a tanker that is unreachable.
Application.ScreenUpdating = False
Tenure = Worksheets("Miscellaneous").Range("C22").Value
Skip = Worksheets("Miscellaneous").Range("D22").Value
```

```
If Worksheets("Miscellaneous").Range("E22").Value = "LARGE" Then
    LargeRestriction = True
Else: LargeRestriction = False
End If
If Worksheets("Miscellaneous").Range("F22").Value = "YES" Then
    Reactive = True
Else: Reactive = False
End If
maxiter = 500
iter = 1
ReDim TabuList(Tenure)
Call SolutionEvaluate(False)
LB = LBValue
UB = 1000 * Value
UBold = 10000 * Value
SearchTime = Timer
Do Until iter > maxiter
    Call MoveNeighborhood(Skip, iter, LargeRestriction)
    'update tabu list
    For i = 1 To Tenure - 1
        TabuList(Tenure - i + 1) = TabuList(Tenure - i)
    Next i
    TabuList(1) = BestMove(3)
    Call SolutionEvaluate(False)
    iter = iter + 1
    'reactive section
    'if 10 consecutive iterations have failed to find best then increase tabu tenure
    If Reactive Then
        If NoImprove > 0.2 * UBound(Xij, 2) Then
            Tenure = Tenure + 1
            ReDim Preserve TabuList(Tenure)
            Debug.Print "Increase"; Tenure
        End If
       'if tabu tenure reaches 90% of customers then reset to initial
       If NoImprove > 0.9 * UBound(Xij, 2) Then
            NoImprove = 0
            Tenure = Worksheets("Miscellaneous").Range("C22").Value
            ReDim Preserve TabuList(Tenure)
            Debug.Print "Reset"; Tenure
        End If
    End If
'    If (UBold - UB) < 0.1 Then
'        Debug.Print "<.1"
'        Exit Do
'    ElseIf Delta = 0 Then
'        Debug.Print "D=0"
'        Exit Do
'    End If
    If Done Then
        Exit Do
    End If
Loop
SearchTime = Timer - SearchTime
Debug.Print SearchTime; iter
```

```
Application.ScreenUpdating = True
End Sub
Sub MoveNeighborhood(SkpNum As Integer, IterNum, Large As Boolean)
'This subroutine develops the candidate list
'remembers best move
Dim TempXij() As Integer, TempYik() As Integer
BestMove(1) = 0
BestMove(2) = 0
BestMove(3) = 0
BestMove(5) = 0
BestMove(4) = 1000000000
'Skip Number Routine
For test = 1 To 10
    check = test
    While check < 1.5 * IterNum
        If check = IterNum Then
            j = test
            Exit For
        End If
        check = check + SkpNum
    Wend
Next test
'beginning of move neighborhood
While j < UBound(Xij, 2)
    Tabu = False
    For n = 1 To UBound(TabuList)
        If TabuList(n) = j Then
            Tabu = True
            Exit For
        End If
    Next n
    If Not Tabu Then 'customer is restricted with size large
        For i = 1 To UBound(Xij)
            If Xij(i, j) = 1 Then
                Index = i
                Exit For
            End If
        Next i
        'step through all other possible anchor areas where demand is less than M
        For i = 1 To UBound(Xij)
            TempXij = Xij
            TempYik = Yik
            TempXij(Index, j) = 0
            If RCost(i, j) < 10000 And i <> Index Then
                TempXij(i, j) = 1
                'if anchor area doesn't have tanker then assign nearest
                tankerinsert = False
                If rowsum(i, Yik) = 0 Then
                    tankerinsert = True
                    cheapest = 10000
                    For k = 1 To UBound(Yik, 2)
                        If TCost(i, k) < cheapest Then
                            If colsum(k, Yik) < 1 Then
                                cheapest = TCost(i, k)
```

```
                    tanker = k
                End If
            End If
        Next k
        TempYik(i, tanker) = 1
    End If
    NVal = EvalF(TempYik, TempXij)
    If NVal < BestMove(4) Then
        BestMove(1) = Index
        BestMove(2) = i
        BestMove(3) = j
        BestMove(4) = NVal
        If tankerinsert Then
            BestMove(5) = tanker
        Else: BestMove(5) = 0
        End If
    End If
  End If
Next i

ElseIf Tabu And Not Large Then 'customer is tabu but with size restriction small
  For i = 1 To UBound(Xij)
    If Xij(i, j) = 1 Then
      Index = i
      Exit For
    End If
  Next i
  'step through all other possible anchor areas where demand is less than M
  For i = 1 To UBound(Xij)
    TempXij = Xij
    TempYik = Yik
    TempXij(Index, j) = 0
    If RCost(i, j) < 10000 And i <> Index Then
      TempXij(i, j) = 1
      'if anchor area doesn't have tanker then move is tabu
      If rowsum(i, Yik) <> 0 Then
        NVal = EvalF(TempYik, TempXij)
        If NVal < BestMove(4) Then
          BestMove(1) = Index
          BestMove(2) = i
          BestMove(3) = j
          BestMove(4) = NVal
          BestMove(5) = 0
        End If
      End If
    End If
  Next i
ElseIf Aspire = True Then 'move is tabu check for aspiration
  For i = 1 To UBound(Xij)
    If Xij(i, j) = 1 Then
      Index = i
      Exit For
    End If
  Next i
```

```
        'step through all other possible anchor areas where demand is less than M
        For i = 1 To UBound(Xij)
            TempXij = Xij
            TempYik = Yik
            TempXij(Index, j) = 0
            If RCost(i, j) < 10000 And i <> Index Then
                TempXij(i, j) = 1
                'if anchor area doesn't have tanker then assign nearest
                tankerinsert = False
                If rowsum(i, Yik) = 0 Then
                    tankerinsert = True
                    cheapest = 10000
                    For k = 1 To UBound(Yik, 2)
                        If TCost(i, k) < cheapest Then
                            If colsum(k, Yik) < 1 Then
                                cheapest = TCost(i, k)
                                tanker = k
                            End If
                        End If
                    Next k
                    TempYik(i, tanker) = 1
                End If
                NVal = EvalF(TempYik, TempXij)
                If NVal < UB Then
                    BestMove(1) = Index
                    BestMove(2) = i
                    BestMove(3) = j
                    BestMove(4) = NVal
                    If tankerinsert Then
                        BestMove(5) = tanker
                    Else: BestMove(5) = 0
                    End If
                End If
            End If
        Next i
    End If
    If BestMove(4) < CurrentVal Then
        j = UBound(Xij, 2) + 1
    Else: j = j + SkpNum
    End If
Wend
If BestMove(1) <> 0 Then
    'perform the move
    Xij(BestMove(1), BestMove(3)) = 0 'unassign receiver
    Xij(BestMove(2), BestMove(3)) = 1 'assign to new group
    Delta = CurrentVal - BestMove(4)
    CurrentVal = BestMove(4)
    If BestMove(5) <> 0 Then
        Yik(BestMove(2), tanker) = 1
    End If
Else: Debug.Print "no move"
End If
'remove any unused tankers
For i = 1 To UBound(Yik)
```

```
    If rowsum(i, Xij) < 1 Then
        For j = 1 To UBound(Yik, 2)
            Yik(i, j) = 0
        Next j
    End If
Next i
End Sub
Sub SolutionEvaluate(Last As Boolean)
'This subroutine evaluates the current or proposes solution
'Calculates objective function value, adds in penalties for infeasibilities.
Dim ObjTCost As Double, ObjRCost As Double
penwt = 1000
'Tanker Costs - costs to operate specific anchor areas as defined by Yik
ObjTCost = 0
For i = 1 To UBound(Yik)
        ObjTCost = ObjTCost + rowmult(i, Yik, TCost)
Next i
'Refueling costs -Costs to refuel according to Xij
ObjRCost = 0
For i = 1 To UBound(Xij)
    ObjRCost = ObjRCost + rowmult(i, Xij, RCost)
Next i
'Penalties -capacity violation penalties
'multiply across yik to sik to find capacity for area i
'then multiply across xij to RCost(i,j) to find demand for area i
cviolations = 0 'number of capacity constraint violations
overcapacity = 0
For i = 1 To UBound(Yik)
    Cap = rowmult(i, Yik, Sik)
    dem = rowmult(i, Xij, RCost)
    If dem > Cap Then
        cviolations = cviolations + 1
        overcapacity = overcapacity + dem - Cap
    End If
Next i
aviolations = 0 'number of unassigned refueling points
For j = 1 To UBound(Xij, 2)
    assigned = colsum(j, Xij)
    If assigned <> 1 Then
        aviolations = aviolations + 1
    End If
Next j
anviolations = 0 'number of anchor areas with more than 1 tanker assigned
For i = 1 To UBound(Yik)
    assigned = rowsum(i, Yik)
    If assigned > 1 Then
        anviolations = anviolations + 1
    End If
Next i
tviolations = 0 'number of times a tanker is assigned more than 1 anchor area
For j = 1 To UBound(Yik, 2)
    assigned = colsum(j, Yik)
    If assigned > 1 Then
        tviolations = tviolations + 1
```

```
    End If
Next j
Infeasibility = cviolations '+ aviolations + anviolations + tviolations
Value = ObjRCost + ObjTCost + penwt * Infeasibility + penwt * overcapacity / 100
CurrentVal = Value
LBValue = ObjRCost + ObjTCost
Debug.Print Value, ObjRCost, ObjTCost, overcapacity, cviolations ', aviolations, anviolations, tviolations
If Infeasibility = 0 And Value < UB Then
    Aspire = True
    UBold = UB
    UB = Value
    BestYik = Yik
    BestXij = Xij
    NumFeas = NumFeas + 1
ElseIf Value = UB Then
    NumFeas = NumFeas + 1
    NoImprove = NoImprove + 1
Else: NoImprove = NoImprove + 1
End If
If NumFeas = 1 Then
    Call SolutionOutput(NumFeas, ObjTCost, ObjRCost)
ElseIf NumFeas > 10 And NoImprove > 10 Then
    Done = True
    Call SolutionOutput(NumFeas, ObjTCost, ObjRCost)
End If
'If Last Then
'    Call SolutionOutput(2, ObjTCost, ObjRCost)
'End If

End Sub
Function EvalF(YMat, XMat) As Double
Dim fObjTCost, fObjRCost
fpenwt = 1000
'Tanker Costs - costs to operate specific anchor areas as defined by Yik
fObjTCost = 0
For i = 1 To UBound(YMat)
        fObjTCost = fObjTCost + rowmult(i, YMat, TCost)
Next i
'Refueling costs -Costs to refuel according to Xij
fObjRCost = 0
For i = 1 To UBound(XMat)
    fObjRCost = fObjRCost + rowmult(i, XMat, RCost)
Next i
'Penalties -capacity violation penalties
'multiply across yik to sik to find capacity for area i
'then multiply across xij to RCost(i,j) to find demand for area i
fcviolations = 0 'number of capacity constraint violations
fovercapacity = 0
For i = 1 To UBound(YMat)
    fcap = rowmult(i, YMat, Sik)
    fdem = rowmult(i, XMat, RCost)
    If fdem > fcap Then
        fcviolations = fcviolations + 1
        fovercapacity = fovercapacity + fdem - fcap
```

```
    End If
Next i
'faviolations = 0 'number of unassigned refueling points
'For j = 1 To UBound(XMat, 2)
'    fassigned = colsum(j, XMat)
'    If fassigned <> 1 Then
'        faviolations = faviolations + 1
'    End If
'Next j
'fanviolations = 0 'number of anchor areas with more than 1 tanker assigned
'For i = 1 To UBound(YMat)
'    fassigned = rowsum(i, YMat)
'    If fassigned > 1 Then
'        fanviolations = fanviolations + 1
'    End If
'Next i
'ftviolations = 0 'number of times a tanker is assigned more than 1 anchor point
'For j = 1 To UBound(YMat, 2)
'    fassigned = colsum(j, YMat)
'    If fassigned > 1 Then
'        ftviolations = ftviolations + 1
'    End If
'Next j
fInfeasibility = fcviolations '+ faviolations + fanviolations + ftviolations
EvalF = fObjRCost + fObjTCost + fpenwt * fInfeasibility + fpenwt * fovercapacity / 100

End Function
'*****************************************************************************************
*******
'Section III.
'*****************************************************************************************
*******
'This is the output generation phase
Sub Solve()
'This subroutine is the main driving subroutine that makes call to other subroutines
'calls subroutines in both Phase I and II and then call subroutines to output.
Application.ScreenUpdating = False
Call Initial
Application.ScreenUpdating = True
TabuSearchForm.Show
Application.ScreenUpdating = False
Call TabuSearch
Call SolutionEvaluate(True)
MsgBox "The search procedure is complete", , "Tanker Employment Tool"
End Sub
Sub SolutionOutput(Which As Integer, TankDist As Double, Fuel As Double)
'This subroutine displays the best solution for user output
Select Case Which
    Case Is = 1
    'Set up refueling point info
        Worksheets("RPPoints").Select
        Columns("A:E").Select
        Selection.Copy
        Sheets("InitialSolution").Select
```

```
    Range("A1").Select
    ActiveSheet.Paste
    Worksheets("RPPoints").Select
    Columns("G:I").Select
    Selection.Copy
    Sheets("InitialSolution").Select
    Range("F1").Select
    ActiveSheet.Paste
'transfer solution info
    With Range("I1")
        For j = 1 To UBound(BestXij, 2)
            For i = 1 To UBound(BestXij)
                If BestXij(i, j) = 1 Then
                    .Offset(j, 0) = i
                    For k = 1 To UBound(BestYik, 2)
                        If BestYik(i, k) = 1 Then
                            .Offset(j, 1) = k
                        End If
                    Next k
                End If
            Next i
        Next j
    End With
    Worksheets("InitialSolution").Range("M3") = TankDist 'tanker distance
    Worksheets("InitialSolution").Range("M4") = Fuel ' offload
    Worksheets("InitialSOlution").Range("N3") = Timer - SearchTime
Case Else
    Worksheets("RPPoints").Select
    Columns("A:E").Select
    Selection.Copy
    Sheets("BestSolution").Select
    Range("A1").Select
    ActiveSheet.Paste
    Worksheets("RPPoints").Select
    Columns("G:I").Select
    Selection.Copy
    Sheets("BestSolution").Select
    Range("F1").Select
    ActiveSheet.Paste
    'output solution info
    With Range("I1")
        For j = 1 To UBound(BestXij, 2)
            For i = 1 To UBound(BestXij)
                If BestXij(i, j) = 1 Then
                    .Offset(j, 0) = i
                    For k = 1 To UBound(BestYik, 2)
                        If BestYik(i, k) = 1 Then
                            .Offset(j, 1) = k
                        End If
                    Next k
                End If
            Next i
        Next j
    End With
```

```
        Worksheets("BestSolution").Range("M3") = TankDist 'tanker distance
        Worksheets("BestSolution").Range("M4") = Fuel   'offload
        Worksheets("BestSolution").Range("M6") = SearchTime 'computation time
End Select
End Sub


'Module:    DistCalc
'Function:  This module contains the functions for calculating the great circle
'           distance from the origin base to the Area of Engagement
' Adapted from Capehart(2000)


'Function Name: DecDeg
'Functionality: To decode the DDDMM.M format (where D=degrees, M=Minutes) for Latitude
'               and Longitude to degrees.
'Arguments:     Number - Value passed to function in DDDMM.M format
'Return Value:  Temp - the Latitude or Longitude in degrees
Private Function DecDeg(Number)

    Num = Abs(Number)   ' Get absolute value of Number to use in Int()

    temp = Int(Num / 100) + (Num / 100 - Int(Num / 100)) / 0.6
                                ' Convert by separating integer degrees from
                                '  minutes portion.  Then divide minutes by 60
                                '  to get fractional degrees and add to integer
                                '  degrees.

    If Num > Number Then            ' Check that Temp has same sign (+/-) as Number
        temp = -temp                '  before assigning to return value
    End If
    DecDeg = temp                   ' Assign Temp to function's return value
End Function
Function DegDec(Number)

    Num = Abs(Number)

    temp = (Floor(Num) * 100) + (60 * (Num - Floor(Num)))

    If Number < 0 Then temp = -temp

    DegDec = temp
End Function

'Function Name: GreatCircleDistance
'Functionality: To compute great circle distance between two points on Earth.  Points
'               are (Latitude1, Longitude1) and (Latitude2, Longitude2).  This function
'               accepts latitude and longitude in real degrees or in DDDMM.M format.
'Arguments:     latitude1 - origin latitude
'               longitude1 - origin longitude
'               latitude2 - destination latitude
'               longitude2 - destination longitude
'Return Value:  GreatCircleDistance - the great circle distance
```

```
Function GreatCircleDistance(latitude1, longitude1, latitude2, longitude2)

    Deg2Rad = 3.14159265358979 / 180    'Define constants
    Rad2Deg = 180 / 3.14159265358979    'for angle conversions
    NMperDeg = 60

    lat1 = latitude1
    lat2 = latitude2
    long1 = longitude1
    long2 = longitude2

    If (Abs(lat1) > 90) Or (Abs(lat2) > 90) Or (Abs(long1) > 180) Or (Abs(long2) > 180) Then
        lat1 = DecDeg(lat1)   ' Assumes all coordinates are in same
        lat2 = DecDeg(lat2)   '  format. If any are found in DDDMM.M
        long1 = DecDeg(long1) '  format then convert all to degrees.
        long2 = DecDeg(long2)
    End If

    lat1 = lat1 * Deg2Rad     ' Convert all degrees to radians
    lat2 = lat2 * Deg2Rad
    long1 = long1 * Deg2Rad
    long2 = long2 * Deg2Rad

    temp = Cos(lat1) * Cos(lat2) * Cos(long2 - long1)

    temp = Application.Acos(temp + Sin(lat1) * Sin(lat2)) * Rad2Deg
                                    ' Calculated the angle of the great circle
                                    '  arc between the two points.  Formula
                                    '  came from original AMCSAF Distcalc
                                    '  spreadsheet.  Uses Excel's ACOS().

  GreatCircleDistance = NMperDeg * temp  ' Convert arc degrees to NM and return
End Function
Function getAz(latitude1, longitude1, latitude2, longitude2)

    Deg2Rad = 3.14159265358979 / 180    'Define constants
    Rad2Deg = 180 / 3.14159265358979    'for angle conversions
    NMperDeg = 60

    lat1 = latitude1
    long1 = longitude1
    lat2 = latitude2
    long2 = longitude2
    dist = GreatCircleDistance(lat1, long1, lat2, long2)

    If (Abs(lat2) > 90) Or (Abs(long1) > 180) Or (Abs(long2) > 180) Then
        lat1 = DecDeg(lat1)   ' Assumes all coordinates are in same
        lat2 = DecDeg(lat2)   '  format. If any are found in DDDMM.M
        long1 = DecDeg(long1) '  format then convert all to degrees.
        long2 = DecDeg(long2)
    End If

    lat1 = lat1 * Deg2Rad
    lat2 = lat2 * Deg2Rad
```

```
    long1 = long1 * Deg2Rad
    long2 = long2 * Deg2Rad
    dist = dist / NMperDeg
    dist = dist * Deg2Rad

    sinAz = (Cos(lat2) * Sin(long2 - long1) / Sin(dist))
    cosAz = ((Sin(lat2) - (Cos(dist) * Sin(lat1))) / (Sin(dist) * Cos(lat1)))
    If sinAz >= 0 And cosAz >= 0 Then
        temp = Application.Asin(sinAz)
    ElseIf sinAz >= 0 And cosAz < 0 Then
        temp = 3.14159265358979 - Application.Asin(sinAz)
    ElseIf cosAz >= 0 Then
        temp = -Application.Acos(cosAz)
    Else
        temp = -(3.14159265358979 + Application.Asin(sinAz))
    End If
    temp = temp * Rad2Deg
    getAz = temp
End Function
Function getLat(latitude1, distance, azymuth)

    Deg2Rad = 3.14159265358979 / 180    'Define constants
    Rad2Deg = 180 / 3.14159265358979    'for angle conversions
    NMperDeg = 60

    lat1 = latitude1
    dist = distance
    az = azymuth

    If (Abs(lat1) > 90) Then lat1 = DecDeg(lat1)

    lat1 = lat1 * Deg2Rad
    dist = dist / NMperDeg
    dist = dist * Deg2Rad
    az = az * Deg2Rad

    temp = Application.Acos(Sin(lat1) * Cos(dist) + Cos(lat1) * Sin(dist) * Cos(az))
    temp = temp * Rad2Deg
    temp = 90 - temp
    getLat = DegDec(temp)
End Function
Function getLong(longitude1, distance, azymuth, latitudeRP)

    Deg2Rad = 3.14159265358979 / 180    'Define constants
    Rad2Deg = 180 / 3.14159265358979    'for angle conversions
    NMperDeg = 60

    long1 = longitude1
    dist = distance
    az = azymuth
    latRP = latitudeRP

    If (Abs(long1) > 90) Or (Abs(latRP) > 90) Then
        long1 = DecDeg(long1)
```

```
        latRP = DecDeg(latRP)
    End If

    dist = dist / NMperDeg
    dist = dist * Deg2Rad
    az = az * Deg2Rad
    latRP = latRP * Deg2Rad
    long1 = long1 * Deg2Rad

    temp = Application.Asin(Sin(dist) * Sin(az) / Cos(latRP))
    temp = temp + long1
    temp = temp * Rad2Deg
    If temp > 180 Then temp = temp - 360
    getLong = DegDec(temp)
End Function

Function TrueCourse(dist, latitude1, longitude1, latitude2, longitude2)

    Deg2Rad = 3.14159265358979 / 180    'Define constants
    Rad2Deg = 180 / 3.14159265358979    'for angle conversions
    p = 3.1415926535897

    la1 = latitude1
    lg1 = longitude1
    la2 = latitude2
    lg2 = longitude2

    If (Abs(la1) > 90) Or (Abs(la2) > 90) Or (Abs(lg1) > 180) Or (Abs(lg2) > 180) Then
        la1 = DecDeg(la1)   ' Assumes all coordinates are in same
        la2 = DecDeg(la2)   '  format. If any are found in DDDMM.M
        lg1 = DecDeg(lg1) '  format then convert all to degrees.
        lg2 = DecDeg(lg2)
    End If

    la1 = la1 * Deg2Rad     ' Convert all degrees to radians
    la2 = la2 * Deg2Rad
    lg1 = lg1 * Deg2Rad
    lg2 = lg2 * Deg2Rad
    D = (dist / 60) * Deg2Rad

    H1 = Application.Acos((Sin(la2) - Sin(la1) * Cos(D)) / (Sin(D) * Cos(la1)))
    H2 = Application.Acos((Sin(la1) - Sin(la2) * Cos(D)) / (Sin(D) * Cos(la2)))

    If Sin(lg2 - lg1) < 0 Then
        Hi1 = H1
    Else
        Hi1 = 2 * p - H1
    End If

    If Sin(lg1 - lg2) < 0 Then
        Hi2 = H2
    Else
        Hi2 = 2 * p - H2
    End If
```

```
   If Hi2 >= p Then
      Hi2 = Hi2 - p
   Else
      Hi2 = Hi2 + p
   End If
   TrueCourse = (Hi1 + Hi2) / 2 * Rad2Deg

End Function

Function GroundSpeed(TAS, TC, Wd, Wv)

   Deg2Rad = 3.14159265358979 / 180    'Define constants
   Rad2Deg = 180 / 3.14159265358979    'for angle conversions

   TCr = TC * Deg2Rad

   Wdr = Wd * Deg2Rad
   DCA = Application.Asin((Wv / TAS) * Sin(Wdr - TCr))
   GroundSpeed = TAS * Cos(DCA) - Wv * Cos(Wdr - TCr)

End Function

‘Module Examples
‘This Module loads the two example scenarios

Sub LoadCarribean()
'Clear Current Info
Call ResetSheet
'Load Carribean example
'Sorties
For i = 1 To 10
   For j = 1 To 6
      Worksheets("Inputs").Range("A1").Offset(i, j - 1) = _
      Worksheets("ExampleData").Range("B3").Offset(i, j - 1)
   Next j
Next i
'Tankers
For i = 1 To 3
   Worksheets("Tankers").Range("A2").Offset(1, i - 1) = _
   Worksheets("ExampleData").Range("B15").Offset(1, i - 1)
Next i
'Engagement Area
For i = 1 To 8
   Worksheets("Engagement Area").Range("A2").Offset(1, i - 1) = _
   Worksheets("ExampleData").Range("B19").Offset(1, i - 1)
Next i

End Sub
Sub LoadSWAsia()
'Clear the sheet
Call ResetSheet
'Load Middle East Example
'Sorties
```

```
For i = 1 To 63
    For j = 1 To 6
        Worksheets("Inputs").Range("A1").Offset(i, j - 1) = _
        Worksheets("ExampleData").Range("B25").Offset(i, j - 1)
    Next j
Next i
'Tankers
For i = 1 To 3
    For j = 1 To 7
        Worksheets("Tankers").Range("A2").Offset(j, i - 1) = _
        Worksheets("ExampleData").Range("B90").Offset(j, i - 1)
    Next j
Next i
'Engagement Area
For i = 1 To 8
    Worksheets("Engagement Area").Range("A2").Offset(1, i - 1) = _
    Worksheets("ExampleData").Range("B100").Offset(1, i - 1)
Next i
End Sub
Sub ResetSheet()
'Clear Sorties
Worksheets("Inputs").Select
Range("A2:J1000").ClearContents
'Clear Tankers
Worksheets("Tankers").Select
Range("A3:C1000").ClearContents
Range("J3:Q9999").ClearContents
'Clear Engagement Area
Worksheets("Engagement Area").Select
Range("A3:H3").ClearContents
Range("A8:D9999").ClearContents
Range("H11:H14").ClearContents
'Clear RPPoints
Worksheets("RPPoints").Select
Range("A2:T9999").ClearContents
'Clear Initial Solution
Worksheets("InitialSolution").Select
Range("A2:J9999").ClearContents
Range("M3:M4").ClearContents
'Clear Best Solution
Worksheets("BestSolution").Select
Range("A2:J9999").ClearContents
Range("M3:M4").ClearContents
Range("M6").ClearContents
End Sub

'Module: Functions
'This Module contains functions used in calculations

Function Ceiling(x)
    temp = CInt(x)
    temp1 = temp - x
    If temp1 > 0 Then
        Ceiling = temp
```

```
    Else
        Ceiling = temp + 1
    End If
End Function
Function Max(a, b)
    If a > b Then
        Max = a
    Else
        Max = b
    End If
End Function
Function Min(a, b)
    If a < b Then
        Min = a
    Else
        Min = b
    End If
End Function
Function Floor(x)
    temp = Abs(x)
    temp = Ceiling(temp)
    If x > 0 Then
        Floor = temp - 1
    Else
        Floor = -temp
    End If
End Function
Function Intersect(lat1, long1, lat2, long2, plat1, plong1, plat2, plong2) As Boolean
'This function determines if a flight path defined by lat/long 1 and lat/long 2
'intersects the line segment formed by plat/plong 1 & 2
x1 = ToMin(lat1)
y1 = ToMin(long1)
x2 = ToMin(lat2)
y2 = ToMin(long2)
px1 = ToMin(plat1)
py1 = ToMin(plong1)
px2 = ToMin(plat2)
py2 = ToMin(plong2)
'x1,y1 ----- x2,y2 single line segment defining flight path in minutes
'px1,py1------px2,py2 single line segment defining possible border in minutes
'm is slope of line 1, b1 is intercept of line 1
'n is slope of line 2, b2 is intercept of line 2
'ix is x of intersection, iy is y of intersection
If x2 - x1 <> 0 Then
    m = (y2 - y1) / (x2 - x1)
Else: m = 1000000
End If
b1 = y1 - m * x1
If px2 - px1 <> 0 Then
    n = (py2 - py1) / (px2 - px1)
Else: n = 1000000
End If
If m = n Then 'parallel then intersect equals false
    Intersect = False
```

```
    Exit Function
End If
b2 = py1 - n * px1
ix = (b1 - b2) / (m - n)
If px1 < ix And ix < px2 Then
    Intersect = True
ElseIf px1 > ix And ix > px2 Then
    Intersect = True
    Debug.Print Intersect
Else
    Intersect = False

End If
End Function
Function ToMin(DegMin)
'this function converts Latitude or longitude to only minutes
If DegMin > 0 Then
    temp = Floor(DegMin / 100)
    x = temp
    temp = (DegMin / 100 - x) * 100
Else
    temp = Ceiling(DegMin / 100)
    x = temp
    temp = (Abs(x) - Abs(DegMin / 100)) * 100
End If
ToMin = 60 * x + temp
End Function
Function FindFuel(AType)
'This function finds the full onload required to fill from min reserve to Max fuel load

Worksheets("Aircraft Info").Select
With Range("A4")
rcount = Range(.Offset(1, 0), .End(xlDown)).Rows.Count
For i = 1 To rcount
    If StrComp(AType, .Offset(i, 0)) = 0 Then
        j = i
        Exit For
    End If
Next i
FindFuel = .Offset(j, 3) - .Offset(j, 4)
End With
End Function


Function recflburn(dist, rate, FuelCap, reserve, minwt, cargo, climbf, c1, c2, c3, c4)
'Function Name: fuelburn
'Functionality: This function is used to determine the fuel
'               burned by a given fighter for a period of
'               flight given by: Flight time = Distance/True Air Speed
'               The algorithm assumes a nominal flight altitude and
'               true air speed. The fuel flow is calculated with a third
'               order polynomial model of the fuel flow depending on gross weight.
'               It is assumed that the fighter's fuel is burned down to
'               the fuel reserve level and then completely refueled.
```

```
'Arguments:     dist - the distance the fighter will travel
'               tas - the true airspeed the fighter will travel at
'               r - the fighter performance matrix
'               j - the position of the desired fighter in the performance matrix
'Return Value:  totfb - total fuel burned over the flight
'Adapted from Capehart (2000)

   fb = 0
   totfb = climbf
   ff = 0
   mult = 0

   gw = 0
   gwi = FuelCap + minwt + cargo
   maxburn = FuelCap - reserve

   Flighttime = dist / rate
   dt = 0.01

   For t = 1 To Flighttime * 100
     Nar = Ceiling(totfb / maxburn) - 1
     gw = gwi - totfb + Nar * maxburn
     ff = c1 + c2 * gw + c3 * gw * gw + c4 * gw * gw * gw
     fb = ff * dt
     totfb = totfb + fb
   Next t

   recflburn = totfb

End Function
Function tnkburn(dist, tos, bcrate, ierate, FuelCap, reserve, minwt, cargo, climbf, c1, c2, c3, c4, c5, c6, c7)
   'this function calculates the tanker fuel burn for an anchor area that is a given distance away
   'with a known tos-time on station, bcrate-buddy cruise speed (avg), ierate (inbound outbound TAS)
   'this does not include the amount of fuel offloaded. It only calculates the amount of fuel the
   'tanker actually burns.
   fb = 0
   totfb = climbf
   ff = 0
   mult = 0

   gw = 0
   gwi = FuelCap + minwt + cargo
   maxburn = FuelCap - reserve

   ietime = (dist / ierate) * 2 'ingress egress time
   Flighttime = ietime + tos 'trip there and back plus hover time at anchor area
   dt = 0.01
   alt1 = 430 'ingress and egress
   alt2 = 250 'refueling
   t = 0
   While t < Flighttime * 100
      If t < ietime * 100 Then
         alt = alt1
         TAS = ierate
```

```
        Else
            alt = alt2
            TAS = bcrate
        End If
        gw = gwi - totfb
        fflow = c1 + c2 * alt + c3 * alt * alt + c4 * TAS + c5 * TAS * TAS + c6 * gw + c7 * gw * gw
        fb = fflow * dt
        totfb = totfb + fb
        t = t + 1
    Wend

    tnkburn = totfb
End Function
Function InRange(i, j) As Boolean
'This function determines whether an ingress refueling point at anchor area j is
'reachable
'*********************************************************
'parameters
Octnt = Worksheets("RPPoints").Range("H1").Offset(i, 2)
pstn = Worksheets("Engagement Area").Range("B7").Offset(j, 2)
'*********************************************************
'the following logic ensures that receivers have fairly direct ingress and egress routes.
'They are not allowed to fly all the way around an engagement zone to reach an anchor area.
Select Case Octnt
    Case Is = 1
        If pstn = "N" Or pstn = "W" Then
            InRange = True
        Else: InRange = False
        End If
    Case Is = 2
        If pstn = "S" Then
            InRange = False
        Else: InRange = True
        End If
    Case Is = 3
        If pstn = "N" Or pstn = "E" Then
            InRange = True
        Else: InRange = False
        End If
    Case Is = 4
        If pstn = "W" Then
            InRange = False
        Else: InRange = True
        End If
    Case Is = 5
        If pstn = "E" Or pstn = "S" Then
            InRange = True
        Else: InRange = False
        End If
    Case Is = 6
        If pstn = "N" Then
            InRange = False
        Else: InRange = True
        End If
```

```
    Case Is = 7
        If pstn = "S" Or pstn = "W" Then
            InRange = True
        Else: InRange = False
        End If
    Case Is = 8
        If pstn = "E" Then
            InRange = False
        Else: InRange = True
        End If
End Select
End Function
Function rowmult(row, a, b)
'this function multiplies two rows of two matrices
'summation over j xij * cij
temp = 0
acol = UBound(a, 2)
For j = 1 To acol
    temp = a(row, j) * b(row, j) + temp
Next j
rowmult = temp
End Function
Function colmult(col, a, b)
'this function multiplies two rows of two matrices
'summation over j xij * cij
temp = 0
arow = UBound(a)
For i = 1 To arow
    temp = a(i, col) * b(i, col) + temp
Next i
colmult = temp
End Function
Function colsum(col, a)
temp = 0
arow = UBound(a)
For i = 1 To arow
    temp = a(i, col) + temp
Next i
colsum = temp
End Function
Function rowsum(row, a)
temp = 0
acol = UBound(a, 2)
For j = 1 To acol
    temp = a(row, j) + temp
Next j
rowsum = temp
End Function


'Module Refueling Tracks
Option Base 1
'PSeudocode
'get distance, get azimuth, divide distance by short side of track.
```

```
'take the ceiling
'for i to ceiling loop, get lat, get long for short side distance, store in tracks

Sub DefTracks()
Application.ScreenUpdating = False
Dim EAarray(4, 2) As Single
Worksheets("Engagement Area").Select
Range("A8:D1000").Select
Selection.ClearContents
With Range("A2")
   EAarray(1, 1) = .Offset(1, 0)
   EAarray(1, 2) = .Offset(1, 1)
   EAarray(2, 1) = .Offset(1, 2)
   EAarray(2, 2) = .Offset(1, 3)
   EAarray(3, 1) = .Offset(1, 4)
   EAarray(3, 2) = .Offset(1, 5)
   EAarray(4, 1) = .Offset(1, 6)
   EAarray(4, 2) = .Offset(1, 7)
End With
'Define tracks in clockwise order

'NW to NE
Call DefSide("N", EAarray(1, 1), EAarray(1, 2), EAarray(2, 1), EAarray(2, 2))
'NE to SE
Call DefSide("E", EAarray(2, 1), EAarray(2, 2), EAarray(4, 1), EAarray(4, 2))
'SE to SW
Call DefSide("S", EAarray(4, 1), EAarray(4, 2), EAarray(3, 1), EAarray(3, 2))
'SW to NW
Call DefSide("W", EAarray(3, 1), EAarray(3, 2), EAarray(1, 1), EAarray(1, 2))
'Number the tracks
With Range("A8")
   ofst = 0
   Do
      If .Offset(ofst, 1).Value = "" Then
         Exit Do
      Else
         .Offset(ofst, 0).Value = ofst + 1
         ofst = ofst + 1
      End If
   Loop
End With
Application.ScreenUpdating = True
End Sub
Sub DefSide(Pos, lat1, long1, lat2, long2)
'this will be called 4 times for each side of the rombus.
'find the short side of the defined track size
Worksheets("Engagement Area").Select
With Range("H7")
   a = .Offset(0, 0)
   b = .Offset(1, 0)
End With
short = Min(a, b) ' short side of track
'find the Distance of the side
dist = GreatCircleDistance(lat1, long1, lat2, long2)
```

```
With Range("H10")
   Select Case Pos
      Case Is = "N"
         .Offset(1, 0) = dist
      Case Is = "W"
         .Offset(2, 0) = dist
      Case Is = "E"
         .Offset(3, 0) = dist
      Case Is = "S"
         .Offset(4, 0) = dist
   End Select
End With
'find the angle
az = getAz(lat1, long1, lat2, long2)
'determine the number of refueling tracks on the side
numtrks = Ceiling(dist / short)
'generate Lat/Longs for each refueling track
With Range("A7")
ofst = 0
Do
   If .Offset(ofst, 1).Value = "" Then
      Exit Do
   Else: ofst = ofst + 1
   End If
Loop
   For i = 1 To numtrks
      nlat = getLat(lat1, short * i, az)
      .Offset(ofst, 1) = nlat
      .Offset(ofst, 2) = getLong(long1, short * i, az, nlat)
      .Offset(ofst, 3) = Pos
      ofst = ofst + 1
   Next i
End With
End Sub

‘Module: Pop up menus

Sub Menu_Click()
'User clicks on the opening start screen
'Initalizes Entry menus
ExampleForm.Show
End Sub
Sub OpenHelp_Click()
'User clicks the help button and the help menus appear
   OpenHelpForm.Show
End Sub
'Pseudocode
'Generate DV for each refueling point
'fuel burn calculation to determine 1 hr flight radius.
   'For each sortie, create a possible ingress RP's, and the midpoint and egress points.
   'create a data array that gives the costs for getting serviced at the tract locations
   ' Array would have the offload required, and variable cost for choosing that location, make the cost
   ' infinity for those tracts that are not allowed. (BIG M).
'Generate DV for each tanker serving a tract.
```

```
    ' have to know how much offload is available if serving the particular tract.
    ' data array giving the travel distance for serving the tract, and offload available, 3 dimensional, tankers,
tracts, distance and offload
'Solution representation
    'array list, first number is tanker, second is track, then customer list..
'Solution evaluation
    ' be able to read in the arrays and calculate objective measures.
'Move neighborhoods
    'Customer swap
        'Two sorties swap positions
    ' Customer insert move
        'Sortie moves from current location and gets inserted into other location
    ' Facility insert move
        ' New tanker serving new tract, to service customer that is moving.
        'latitude 0 to 90
        'DDDMM.M Format use this number for the distances
Sub MainMenu()
    Menu.Show
End Sub
```

Code for each menu

ADD AVAILABLE TANKER

```
Private Sub AddAnother_Click()
'if select unrecognized ICAO, prompt for adding airbase
With TankerBase
    If .ListIndex = -1 And .Value = "" Then
        MsgBox "Please select an ICAO from the list", , "Tanker Employment Tool"
        .SetFocus
        Exit Sub
    ElseIf .ListIndex = -1 Then 'if not in list then prompt to add base
        MsgBox "Please add base or select ICAO from list", , "Tanker Employment Tool"
        Unload Me
        AddBaseForm.Show
    Else: TICAO = .Value
    End If
End With

'if select unrecognized tanker type prompt for adding tanker
With TankerType
    If .ListIndex = -1 And .Value = "" Then
        MsgBox "Please select a Tanker from the list", , "Tanker Employment Tool"
        .SetFocus
        Exit Sub
    ElseIf .ListIndex = -1 Then 'if not in list then prompt to add aircraft
        MsgBox "Please add new tanker type or select tanker from list", , "Tanker Employment Tool"
        Unload Me
        AddTankerForm.Show
    Else: TType = .Value
    End If
End With
'make sure number available is number
With TankNum
    If .Value = "" Or Not IsNumeric(.Value) Then
```

```
        MsgBox "Please enter a number > 0", , "Tanker Employment Tool"
        .SetFocus
        Exit Sub
    Else: TNum = .Value
    End If
End With
'dump the info to excel input area
Unload Me
Worksheets("Tankers").Select
Range("A3:C100").Select
Selection.Cut Destination:=Range("A4:C1001")
Range("A3:C3").Select
    With Selection
        .HorizontalAlignment = xlCenter
        .VerticalAlignment = xlBottom
        .WrapText = False
        .Orientation = 0
        .AddIndent = False
        .IndentLevel = 0
        .ShrinkToFit = False
        .ReadingOrder = xlContext
        .MergeCells = False
    End With
With Range("A2")
    .Offset(1, 0) = TICAO
    .Offset(1, 1) = TNum
    .Offset(1, 2) = TType
End With
    'display message
    MsgBox "Tankers added", , "Tanker Employment Tool"
    Menu.Show
End Sub

Private Sub Cancel_Click()
    Unload Me
    Menu.Show
End Sub

ADD BASE
Private Sub AddBaseButton_Click()
'Check to See if they are adding a new Unique ICAO
With BaseICAO
    If .ListIndex <> -1 Then
        MsgBox "Please enter a unique ICAO", , "Tanker Employment Tool"
        .SetFocus
        Exit Sub
    Else: BICAO = .Value
    End If
End With
'Make Sure they have added a Name and Region
With BaseName
    If .Value = "" Then
        MsgBox "Please Enter an Airbase Name", , "Tanker Employment Tool"
        .SetFocus
```

```
    Else: BName = .Value
    End If
End With
With BaseRegion
    If .Value = "" Then
        MsgBox "Please Enter a Region", , "Tanker Employment Tool"
        .SetFocus
    Else: BReg = .Value
    End If
End With

'Check lat longs to make sure they are valid
With LatDeg
    If .Value = "" Or Not IsNumeric(.Value) Or .Value < 0 Or .Value > 180 Then
        MsgBox "Please enter a number from 1 to 180", , "Tanker Employment Tool"
        .SetFocus
        Exit Sub
    Else: LatDeg = .Value
    End If
End With
With LongDeg
    If .Value = "" Or Not IsNumeric(.Value) Or .Value < 0 Or .Value > 180 Then
        MsgBox "Please enter a number from 1 to 180", , "Tanker Employment Tool"
        .SetFocus
        Exit Sub
    Else: LongDeg = .Value
    End If
End With
With LatMin
    If .Value = "" Or Not IsNumeric(.Value) Or .Value < 0 Or .Value > 60 Then
        MsgBox "Please enter a number from 1 to 60", , "Tanker Employment Tool"
        .SetFocus
        Exit Sub
    Else: LatMin = .Value
    End If
End With
With LongMin
    If .Value = "" Or Not IsNumeric(.Value) Or .Value < 0 Or .Value > 60 Then
        MsgBox "Please enter a number from 1 to 60", , "Tanker Employment Tool"
        .SetFocus
        Exit Sub
    Else: LongMin = .Value
    End If
End With
If LatNorth = False And LatSouth = False Then
    MsgBox "Please select North or South for Latitude", , "Tanker Employment Tool"
ElseIf LatNorth = True Then
    North = True
Else: North = False
End If
If LongEast = False And LongWest = False Then
    MsgBox "Please select East or West for Longitude", , "Tanker Employment Tool"
ElseIf LongEast = True Then
    East = True
```

```
Else: East = False
End If
'collect info dump into excel
Application.ScreenUpdating = False
Worksheets("Bases Info").Select
Rows(30).Select
Selection.Insert
With Range("A1")
   Index = 29
   .Offset(Index, 0) = BICAO
   .Offset(Index, 1) = BName
   .Offset(Index, 2) = BReg
   If LatMin > 10 And LongMin > 10 Then
      If North = False Then
         .Offset(Index, 3) = "-" & LatDeg & LatMin
      Else
         .Offset(Index, 3) = LatDeg & LatMin
      End If
      If East = True Then
         .Offset(Index, 4) = "-" & LongDeg & LongMin
      Else
         .Offset(Index, 4) = LongDeg & LongMin
      End If
   ElseIf LatMin < 10 And LongMin < 10 Then 'add "0" for spacing
      If North = False Then
         .Offset(Index, 3) = "-" & LatDeg & "0" & LatMin
      Else
         .Offset(Index, 3) = LatDeg & "0" & LatMin
      End If
      If East = True Then
        .Offset(Index, 4) = "-" & LongDeg & "0" & LongMin
      Else
         .Offset(Index, 4) = LongDeg & "0" & LongMin
      End If
   ElseIf LatMin < 10 Then
      If North = False Then
         .Offset(Index, 3) = "-" & LatDeg & "0" & LatMin
      Else
         .Offset(Index, 3) = LatDeg & "0" & LatMin
      End If
      If East = True Then
         .Offset(Index, 4) = "-" & LongDeg & LongMin
      Else
         .Offset(Index, 4) = LongDeg & LongMin
      End If
   Else
      If North = False Then
         .Offset(Index, 3) = "-" & LatDeg & LatMin
      Else
        .Offset(Index, 3) = LatDeg & LatMin
      End If
      If East = True Then
        .Offset(Index, 4) = "-" & LongDeg & "0" & LongMin
      Else
```

```
            .Offset(Index, 4) = LongDeg & "0" & LongMin
        End If
    End If
End With
Range("A2:E13").Select
    Range(Selection, Selection.End(xlDown)).Select
    Selection.Sort Key1:=Range("A2"), Order1:=xlAscending, Header:=xlGuess, _
        OrderCustom:=1, MatchCase:=False, Orientation:=xlTopToBottom, _
        DataOption1:=xlSortNormal
Application.ScreenUpdating = True
Unload Me
End Sub

Private Sub BaseCancel_Click()
    Unload Me
    Menu.Show
End Sub

ADD SORTIE
Private Sub AddSortieButton_Click()
Dim SID As String, Numsorties As Integer
'check to see that sortie ID is unique
With SortieID
    If .Value = "" Then
        MsgBox "Please enter a unique Sortie ID", , "Tanker Employment Tool"
        .SetFocus
        Exit Sub
    Else: SID = .Value
    End If
End With
With Worksheets("Inputs").Range("A1")
    If .Offset(1, 0) <> "" Then
        Numsorties = Range(.Offset(1, 0), .End(xlDown)).Rows.Count
        For i = 1 To Numsorties
            If .Offset(i, 0).Value = SID Then
                MsgBox "Please enter a unique Sortie ID", , "Tanker Employment Tool"
                SortieID.SetFocus
                Exit Sub
            End If
        Next i
    End If
End With

'Make sure ICAO is not blank
With SortieICAO
    If .ListIndex = -1 And .Value = "" Then
        MsgBox "Please select an ICAO from the list", , "Tanker Employment Tool"
        .SetFocus
        Exit Sub
    ElseIf .ListIndex = -1 Then 'if not in list then prompt to add base
        MsgBox "Please add base or select ICAO from list", , "Tanker Employment Tool"
        Unload Me
        AddBaseForm.Show
    Else: SICAO = .Value
```

```
    End If

End With

'check to make sure number receivers is numeric
With SortieNumRec
    If .Value = "" Or Not IsNumeric(.Value) Then
        MsgBox "Please enter a number from 1 to 10", , "Tanker Employment Tool"
        .SetFocus
        Exit Sub
    Else: SNumR = .Value
    End If
End With

'if receiver type is not in list prompt them to add receiver aircraft
With SortieReceiver
    If .ListIndex = -1 And .Value = "" Then
        MsgBox "Please select a Receiver from the list", , "Tanker Employment Tool"
        .SetFocus
        Exit Sub
    ElseIf .ListIndex = -1 Then 'if not in list then prompt to add aircraft
        MsgBox "Please add aircraft or select receiver from list", , "Tanker Employment Tool"
        Unload Me
        InsertAircraftForm.Show
    Else: SRec = .Value
    End If
End With

'if mission type is not in list display warning
With SortieType
    If .ListIndex = -1 Then
        MsgBox "Please enter a valid mission type", , "Tanker Employment Tool"
        .SetFocus
        Exit Sub
    Else: SType = .Value
    End If
End With

'if number of waypoints is greater than 2 display warning
With SortieWaypoint
    If .ListIndex = -1 Then
        MsgBox "Sorry model is limited to 2 waypoints, please select a number between 0-2", , "Tanker
Employment Tool"
        .SetFocus
        Exit Sub
    Else: SWPT = .Value
    End If
End With
' dump data into input spreadsheet
Unload Me
Worksheets("Inputs").Select
Range("A2:J1000").Select
Selection.Cut Destination:=Range("A3:J1001")
Range("A2:J2").Select
```

```
    With Selection
        .HorizontalAlignment = xlCenter
        .VerticalAlignment = xlBottom
        .WrapText = False
        .Orientation = 0
        .AddIndent = False
        .IndentLevel = 0
        .ShrinkToFit = False
        .ReadingOrder = xlContext
        .MergeCells = False
        .Font.ColorIndex = 5
    End With
With Range("A1")
    .Offset(1, 0) = SID
    .Offset(1, 1) = SRec
    .Offset(1, 2) = SNumR
    .Offset(1, 3) = SICAO
    .Offset(1, 4) = SType
End With
Worksheets("Inputs").Range("F1").Offset(1, 0).Formula = "= VLOOKUP(RC[-
1],Armament,VLOOKUP(RC[-4],Atype,2)+2)"
'if waypoints selected prompt for waypoint info
If SWPT > 0 Then
    Application.ScreenUpdating = False
    Worksheets("Inputs").Select
    Range("G3:J3").Select
    Selection.AutoFill Destination:=Range("G2:J3"), Type:=xlFillDefault
    Range("G2:J3").Select
    Range("G2:J2").Select
    Selection.ClearContents
    For i = 1 To SWPT
        AddWPT.Show
    Next i
    Application.ScreenUpdating = True
End If

End Sub

Private Sub SortieCancel_Click()
    Unload Me
    Menu.Show
End Sub

ADD TANKER
Private Sub AddTanker_Click()
'Check to See if they are adding a tanker
With TankerName
    If .Value = "" Then
        MsgBox "Please enter a Tanker name", , "Tanker Employment Tool"
        .SetFocus
        Exit Sub
    Else: TName = .Value
    End If
End With
```

```
'Make Sure they have added a legitimate TAS
With TAS
   If .Value = "" Or Not IsNumeric(.Value) Or .Value < 0 Then
      MsgBox "Please Enter a Number for TAS", , "Tanker Employment Tool"
      .SetFocus
   Else: TTAS = .Value
   End If
End With
'make sure they added a true fuel flow
With FuelFlow
   If .Value = "" Or Not IsNumeric(.Value) Or .Value < 0 Then
      MsgBox "Please Enter a Number for Fuel Flow", , "Tanker Employment Tool"
      .SetFocus
   Else: TFlow = .Value
   End If
End With
'make sure they added a true fuel capacity
With FuelCap
   If .Value = "" Or Not IsNumeric(.Value) Or .Value < 0 Then
      MsgBox "Please Enter a Number for Fuel Cap", , "Tanker Employment Tool"
      .SetFocus
   Else: TCap = .Value
   End If
End With

'make sure they added a true fuel reserve
With Res
   If .Value = "" Or Not IsNumeric(.Value) Or .Value < 0 Then
      MsgBox "Please Enter a Number for Reserve", , "Tanker Employment Tool"
      .SetFocus
   Else: TRes = .Value
   End If
End With
'make sure they added a true offload
With AvgOffload
   If .Value = "" Or Not IsNumeric(.Value) Or .Value < 0 Then
      MsgBox "Please Enter a Number for Offload", , "Tanker Employment Tool"
      .SetFocus
   Else: TLoad = .Value
   End If
End With
'collect info dump into excel
Application.ScreenUpdating = False
Worksheets("Aircraft Info").Select
With Range("A2")
guess = Range(.Offset(0, 0), .End(xlDown)).Rows.Count + 6
End With
Rows(guess).Select
Selection.Insert
With Range("A2")
   Index = Range(.Offset(0, 0), .End(xlDown)).Rows.Count + 4
   .Offset(Index, 0) = TName
   .Offset(Index, 1) = TAS
   .Offset(Index, 2) = TFlow
```

```
    .Offset(Index, 3) = TCap
    .Offset(Index, 4) = TRes
    .Offset(Index, 10) = TLoad
    .Offset(Index, 11).Formula = "=RC[-1]/122.94"
    .Offset(Index, 11).Font.ColorIndex = 3
End With
Range("A19:L100").Select
    Selection.Sort Key1:=Range("A2"), Order1:=xlAscending, Header:=xlGuess, _
        OrderCustom:=1, MatchCase:=False, Orientation:=xlTopToBottom, _
        DataOption1:=xlSortNormal
Application.ScreenUpdating = True
Unload Me
End Sub

Private Sub TankerCancel_Click()
    Unload Me
    Menu.Show
End Sub
```

ADD WAYPOINT

```
Private Sub AddWPT_Click()
'check to see if the first waypoint is blank
If Range("F2").Value = "" Then
    ofst = 5
Else: ofst = 7
End If
'error check data entry and save
With NWLatDeg
    If .Value = "" Or Not IsNumeric(.Value) Or .Value < 0 Or .Value > 180 Then
        MsgBox "Please enter a number from 1 to 180", , "Tanker Employment Tool"
        .SetFocus
        Exit Sub
    Else: LatDeg = .Value
    End If
End With
With NWLongDeg
    If .Value = "" Or Not IsNumeric(.Value) Or .Value < 0 Or .Value > 180 Then
        MsgBox "Please enter a number from 1 to 180", , "Tanker Employment Tool"
        .SetFocus
        Exit Sub
    Else: LongDeg = .Value
    End If
End With
With NWLatMin
    If .Value = "" Or Not IsNumeric(.Value) Or .Value < 0 Or .Value > 60 Then
        MsgBox "Please enter a number from 1 to 60", , "Tanker Employment Tool"
        .SetFocus
        Exit Sub
    Else: LatMin = .Value
    End If
End With
With NWLongMin
    If .Value = "" Or Not IsNumeric(.Value) Or .Value < 0 Or .Value > 60 Then
        MsgBox "Please enter a number from 1 to 60", , "Tanker Employment Tool"
```

```
        .SetFocus
        Exit Sub
    Else: LongMin = .Value
    End If
End With
If NWLatNorth = False And NWLatSouth = False Then
    MsgBox "Please select North or South for Latitude", , "Tanker Employment Tool"
    Exit Sub
ElseIf NWLatNorth = True Then
    North = True
Else: North = False
End If
If NWLongEast = False And NWLongWest = False Then
    MsgBox "Please select East or West for Longitude", , "Tanker Employment Tool"
    Exit Sub
ElseIf NWLongEast = True Then
    East = True
Else: East = False
End If
With Range("A2")
    If LatMin > 10 And LongMin > 10 Then
        If North = False Then
            .Offset(0, ofst) = "-" & LatDeg & LatMin
        Else
            .Offset(0, ofst) = LatDeg & LatMin
        End If
        If East = True Then
            .Offset(0, ofst + 1) = "-" & LongDeg & LongMin
        Else
            .Offset(0, ofst + 1) = LongDeg & LongMin
        End If
    ElseIf LatMin < 10 And LongMin < 10 Then 'add "0" for spacing
        If North = False Then
            .Offset(0, ofst) = "-" & LatDeg & "0" & LatMin
        Else
            .Offset(0, ofst) = LatDeg & "0" & LatMin
        End If
        If East = True Then
           .Offset(0, ofst + 1) = "-" & LongDeg & "0" & LongMin
        Else
            .Offset(0, ofst + 1) = LongDeg & "0" & LongMin
        End If
    ElseIf LatMin < 10 Then
        If North = False Then
            .Offset(0, ofst) = "-" & LatDeg & "0" & LatMin
        Else
            .Offset(0, ofst) = LatDeg & "0" & LatMin
        End If
        If East = True Then
           .Offset(0, ofst + 1) = "-" & LongDeg & LongMin
        Else
            .Offset(0, ofst + 1) = LongDeg & LongMin
        End If
    Else
```

```
        If North = False Then
            .Offset(0, ofst) = "-" & LatDeg & LatMin
        Else
            .Offset(0, ofst) = LatDeg & LatMin
        End If
        If East = True Then
            .Offset(0, ofst + 1) = "-" & LongDeg & "0" & LongMin
        Else
           .Offset(0, ofst + 1) = LongDeg & "0" & LongMin
        End If
    End If
End With
Unload Me
End Sub

Private Sub WPTCancel_Click()
Unload Me
End Sub

DEFINE ENGAGEMENT AREA
Private Sub DefineArea_Click()
'Define Array that holds lat longs for dump into excel
Dim EAarray(4, 2) As Integer
'Ensure that each corner has proper lat longs and they have selected North/South and East/West on each
option
'Check lat longs to make sure they are valid
With NWLatDeg
    If .Value = "" Or Not IsNumeric(.Value) Or .Value < 0 Or .Value > 180 Then
        MsgBox "Please enter a number from 1 to 180", , "Tanker Employment Tool"
        .SetFocus
        Exit Sub
    Else: LatDeg = .Value
    End If
End With
With NWLongDeg
    If .Value = "" Or Not IsNumeric(.Value) Or .Value < 0 Or .Value > 180 Then
        MsgBox "Please enter a number from 1 to 180", , "Tanker Employment Tool"
        .SetFocus
        Exit Sub
    Else: LongDeg = .Value
    End If
End With
With NWLatMin
    If .Value = "" Or Not IsNumeric(.Value) Or .Value < 0 Or .Value > 60 Then
        MsgBox "Please enter a number from 1 to 60", , "Tanker Employment Tool"
        .SetFocus
        Exit Sub
    Else: LatMin = .Value
    End If
End With
With NWLongMin
    If .Value = "" Or Not IsNumeric(.Value) Or .Value < 0 Or .Value > 60 Then
        MsgBox "Please enter a number from 1 to 60", , "Tanker Employment Tool"
        .SetFocus
```

```
        Exit Sub
    Else: LongMin = .Value
    End If
End With
If NWLatNorth = False And NWLatSouth = False Then
    MsgBox "Please select North or South for Latitude", , "Tanker Employment Tool"
    Exit Sub
ElseIf NWLatNorth = True Then
    North = True
Else: North = False
End If
If NWLongEast = False And NWLongWest = False Then
    MsgBox "Please select East or West for Longitude", , "Tanker Employment Tool"
    Exit Sub
ElseIf NWLongEast = True Then
    East = True
Else: East = False
End If
If LatMin > 10 And LongMin > 10 Then

    If North = False Then
        EAarray(1, 1) = "-" & LatDeg & LatMin
    Else
        EAarray(1, 1) = LatDeg & LatMin
    End If
    If East = True Then
        EAarray(1, 2) = "-" & LongDeg & LongMin
    Else
        EAarray(1, 2) = LongDeg & LongMin
    End If
ElseIf LatMin < 10 And LongMin < 10 Then 'add "0" for spacing
    If North = False Then
        EAarray(1, 1) = "-" & LatDeg & "0" & LatMin
    Else
        EAarray(1, 1) = LatDeg & "0" & LatMin
    End If
    If East = True Then
        EAarray(1, 2) = "-" & LongDeg & "0" & LongMin
    Else
        EAarray(1, 2) = LongDeg & "0" & LongMin
    End If
ElseIf LatMin < 10 Then
    If North = False Then
        EAarray(1, 1) = "-" & LatDeg & "0" & LatMin
    Else
        EAarray(1, 1) = LatDeg & "0" & LatMin
    End If
    If East = True Then
        EAarray(1, 2) = "-" & LongDeg & LongMin
    Else
        EAarray(1, 2) = LongDeg & LongMin
    End If
Else
    If North = False Then
```

```
        EAarray(1, 1) = "-" & LatDeg & LatMin
    Else
        EAarray(1, 1) = LatDeg & LatMin
    End If
    If East = True Then
        EAarray(1, 2) = "-" & LongDeg & "0" & LongMin
    Else
        EAarray(1, 2) = LongDeg & "0" & LongMin
    End If
End If
'Repeat this procedure for each corner
'NorthEast Corner
With NELatDeg
    If .Value = "" Or Not IsNumeric(.Value) Or .Value < 0 Or .Value > 180 Then
        MsgBox "Please enter a number from 1 to 180", , "Tanker Employment Tool"
        .SetFocus
        Exit Sub
    Else: LatDeg = .Value
    End If
End With
With NELongDeg
    If .Value = "" Or Not IsNumeric(.Value) Or .Value < 0 Or .Value > 180 Then
        MsgBox "Please enter a number from 1 to 180", , "Tanker Employment Tool"
        .SetFocus
        Exit Sub
    Else: LongDeg = .Value
    End If
End With
With NELatMin
    If .Value = "" Or Not IsNumeric(.Value) Or .Value < 0 Or .Value > 60 Then
        MsgBox "Please enter a number from 1 to 60", , "Tanker Employment Tool"
        .SetFocus
        Exit Sub
    Else: LatMin = .Value
    End If
End With
With NELongMin
    If .Value = "" Or Not IsNumeric(.Value) Or .Value < 0 Or .Value > 60 Then
        MsgBox "Please enter a number from 1 to 60", , "Tanker Employment Tool"
        .SetFocus
        Exit Sub
    Else: LongMin = .Value
    End If
End With
If NELatNorth = False And NELatSouth = False Then
    MsgBox "Please select North or South for Latitude", , "Tanker Employment Tool"
    Exit Sub
ElseIf NELatNorth = True Then
    North = True
Else: North = False
End If
If NELongEast = False And NELongWest = False Then
    MsgBox "Please select East or West for Longitude", , "Tanker Employment Tool"
    Exit Sub
```

```
ElseIf NELongEast = True Then
   East = True
Else: East = False
End If
If LatMin > 10 And LongMin > 10 Then

   If North = False Then
      EAarray(2, 1) = "-" & LatDeg & LatMin
   Else
      EAarray(2, 1) = LatDeg & LatMin
   End If
   If East = True Then
      EAarray(2, 2) = "-" & LongDeg & LongMin
   Else
      EAarray(2, 2) = LongDeg & LongMin
   End If
ElseIf LatMin < 10 And LongMin < 10 Then 'add "0" for spacing
   If North = False Then
      EAarray(2, 1) = "-" & LatDeg & "0" & LatMin
   Else
      EAarray(2, 1) = LatDeg & "0" & LatMin
   End If
   If East = True Then
      EAarray(2, 2) = "-" & LongDeg & "0" & LongMin
   Else
      EAarray(2, 2) = LongDeg & "0" & LongMin
   End If
ElseIf LatMin < 10 Then
   If North = False Then
      EAarray(2, 1) = "-" & LatDeg & "0" & LatMin
   Else
      EAarray(2, 1) = LatDeg & "0" & LatMin
   End If
   If East = True Then
      EAarray(2, 2) = "-" & LongDeg & LongMin
   Else
      EAarray(2, 2) = LongDeg & LongMin
   End If
Else
   If North = False Then
      EAarray(2, 1) = "-" & LatDeg & LatMin
   Else
      EAarray(2, 1) = LatDeg & LatMin
   End If
   If East = True Then
      EAarray(2, 2) = "-" & LongDeg & "0" & LongMin
   Else
      EAarray(2, 2) = LongDeg & "0" & LongMin
   End If
End If

'SouthWest Corner
With SWLatDeg
   If .Value = "" Or Not IsNumeric(.Value) Or .Value < 0 Or .Value > 180 Then
```

```
        MsgBox "Please enter a number from 1 to 180", , "Tanker Employment Tool"
        .SetFocus
        Exit Sub
    Else: LatDeg = .Value
    End If
End With
With SWLongDeg
    If .Value = "" Or Not IsNumeric(.Value) Or .Value < 0 Or .Value > 180 Then
        MsgBox "Please enter a number from 1 to 180", , "Tanker Employment Tool"
        .SetFocus
        Exit Sub
    Else: LongDeg = .Value
    End If
End With
With SWLatMin
    If .Value = "" Or Not IsNumeric(.Value) Or .Value < 0 Or .Value > 60 Then
        MsgBox "Please enter a number from 1 to 60", , "Tanker Employment Tool"
        .SetFocus
        Exit Sub
    Else: LatMin = .Value
    End If
End With
With SWLongMin
    If .Value = "" Or Not IsNumeric(.Value) Or .Value < 0 Or .Value > 60 Then
        MsgBox "Please enter a number from 1 to 60", , "Tanker Employment Tool"
        .SetFocus
        Exit Sub
    Else: LongMin = .Value
    End If
End With
If SWLatNorth = False And SWLatSouth = False Then
    MsgBox "Please select North or South for Latitude", , "Tanker Employment Tool"
    Exit Sub
ElseIf SWLatNorth = True Then
    North = True
Else: North = False
End If
If SWLongEast = False And SWLongWest = False Then
    MsgBox "Please select East or West for Longitude", , "Tanker Employment Tool"
    Exit Sub
ElseIf SWLongEast = True Then
    East = True
Else: East = False
End If
If LatMin > 10 And LongMin > 10 Then

    If North = False Then
        EAarray(3, 1) = "-" & LatDeg & LatMin
    Else
        EAarray(3, 1) = LatDeg & LatMin
    End If
    If East = True Then
        EAarray(3, 2) = "-" & LongDeg & LongMin
    Else
```

```
        EAarray(3, 2) = LongDeg & LongMin
    End If
ElseIf LatMin < 10 And LongMin < 10 Then 'add "0" for spacing
    If North = False Then
        EAarray(3, 1) = "-" & LatDeg & "0" & LatMin
    Else
        EAarray(3, 1) = LatDeg & "0" & LatMin
    End If
    If East = True Then
        EAarray(3, 2) = "-" & LongDeg & "0" & LongMin
    Else
        EAarray(3, 2) = LongDeg & "0" & LongMin
    End If
ElseIf LatMin < 10 Then
    If North = False Then
        EAarray(3, 1) = "-" & LatDeg & "0" & LatMin
    Else
        EAarray(3, 1) = LatDeg & "0" & LatMin
    End If
    If East = True Then
        EAarray(3, 2) = "-" & LongDeg & LongMin
    Else
        EAarray(3, 2) = LongDeg & LongMin
    End If
Else
    If North = False Then
        EAarray(3, 1) = "-" & LatDeg & LatMin
    Else
        EAarray(3, 1) = LatDeg & LatMin
    End If
    If East = True Then
        EAarray(3, 2) = "-" & LongDeg & "0" & LongMin
    Else
        EAarray(3, 2) = LongDeg & "0" & LongMin
    End If
End If
'SouthEast Corner
With SELatDeg
    If .Value = "" Or Not IsNumeric(.Value) Or .Value < 0 Or .Value > 180 Then
        MsgBox "Please enter a number from 1 to 180", , "Tanker Employment Tool"
        .SetFocus
        Exit Sub
    Else: LatDeg = .Value
    End If
End With
With SELongDeg
    If .Value = "" Or Not IsNumeric(.Value) Or .Value < 0 Or .Value > 180 Then
        MsgBox "Please enter a number from 1 to 180", , "Tanker Employment Tool"
        .SetFocus
        Exit Sub
    Else: LongDeg = .Value
    End If
End With
With SELatMin
```

```
    If .Value = "" Or Not IsNumeric(.Value) Or .Value < 0 Or .Value > 60 Then
        MsgBox "Please enter a number from 1 to 60", , "Tanker Employment Tool"
        .SetFocus
        Exit Sub
    Else: LatMin = .Value
    End If
End With
With SELongMin
    If .Value = "" Or Not IsNumeric(.Value) Or .Value < 0 Or .Value > 60 Then
        MsgBox "Please enter a number from 1 to 60", , "Tanker Employment Tool"
        .SetFocus
        Exit Sub
    Else: LongMin = .Value
    End If
End With
If SELatNorth = False And SELatSouth = False Then
    MsgBox "Please select North or South for Latitude", , "Tanker Employment Tool"
    Exit Sub
ElseIf SELatNorth = True Then
    North = True
Else: North = False
End If
If SELongEast = False And SELongWest = False Then
    MsgBox "Please select East or West for Longitude", , "Tanker Employment Tool"
    Exit Sub
ElseIf SELongEast = True Then
    East = True
Else: East = False
End If
If LatMin > 10 And LongMin > 10 Then

    If North = False Then
        EAarray(4, 1) = "-" & LatDeg & LatMin
    Else
        EAarray(4, 1) = LatDeg & LatMin
    End If
    If East = True Then
        EAarray(4, 2) = "-" & LongDeg & LongMin
    Else
        EAarray(4, 2) = LongDeg & LongMin
    End If
ElseIf LatMin < 10 And LongMin < 10 Then 'add "0" for spacing
    If North = False Then
        EAarray(4, 1) = "-" & LatDeg & "0" & LatMin
    Else
        EAarray(4, 1) = LatDeg & "0" & LatMin
    End If
    If East = True Then
        EAarray(4, 2) = "-" & LongDeg & "0" & LongMin
    Else
        EAarray(4, 2) = LongDeg & "0" & LongMin
    End If
ElseIf LatMin < 10 Then
    If North = False Then
```

```
        EAarray(4, 1) = "-" & LatDeg & "0" & LatMin
    Else
        EAarray(4, 1) = LatDeg & "0" & LatMin
    End If
    If East = True Then
        EAarray(4, 2) = "-" & LongDeg & LongMin
    Else
        EAarray(4, 2) = LongDeg & LongMin
    End If
Else
    If North = False Then
        EAarray(4, 1) = "-" & LatDeg & LatMin
    Else
        EAarray(4, 1) = LatDeg & LatMin
    End If
    If East = True Then
        EAarray(4, 2) = "-" & LongDeg & "0" & LongMin
    Else
        EAarray(4, 2) = LongDeg & "0" & LongMin
    End If
End If
Unload Me
Application.ScreenUpdating = False
Worksheets("Engagement Area").Select
With Range("A2")
    .Offset(1, 0) = EAarray(1, 1)
    .Offset(1, 1) = EAarray(1, 2)
    .Offset(1, 2) = EAarray(2, 1)
    .Offset(1, 3) = EAarray(2, 2)
    .Offset(1, 4) = EAarray(3, 1)
    .Offset(1, 5) = EAarray(3, 2)
    .Offset(1, 6) = EAarray(4, 1)
    .Offset(1, 7) = EAarray(4, 2)
End With
Application.ScreenUpdating = True

End Sub

Private Sub EngagementCancel_Click()
    Unload Me
    Menu.Show
End Sub
```

LOAD EXAMPLE

```
Private Sub ExampleCancel_Click()
'Display a message and end program
    Dim Response As Variant
    Unload Me
    Response = MsgBox("Please input all data before running program. Help is available by clicking the
help button.", _
    vbOKOnly, "Tanker Employment Tool")
    End
End Sub
```

```
Private Sub ExampleOK_Click()
Dim Example As Integer, Response As Variant
    'Capture the user's example choice
    If Example1 = True Then
        Example = 1
    ElseIf Example2 = True Then
        Example = 2
    Else
        Example = 4 'No example selected
    End If
    Unload Me
    Select Case Example
    Case Is = 1
        Call LoadCarribean
    Case Is = 2
        Call LoadSWAsia
    Case Else
        ExpertForm.Show
    End Select
End Sub
```

INSERT AIRCRAFT

```
Private Sub AddAircraftButton_Click()
'Check to See if they are typed a name
With AircraftName
    If .Value = "" Then
        MsgBox "Please enter an aircraft name", , "Tanker Employment Tool"
        .SetFocus
        Exit Sub
    Else: AName = .Value
    End If
End With
'Make Sure they have added a legitimate TAS
With AircraftTAS
    If .Value = "" Or Not IsNumeric(.Value) Or .Value < 0 Then
        MsgBox "Please Enter a Number for TAS", , "Tanker Employment Tool"
        .SetFocus
    Else: TAS = .Value
    End If
End With
'make sure they added a true fuel flow
With AircraftFlow
    If .Value = "" Or Not IsNumeric(.Value) Or .Value < 0 Then
        MsgBox "Please Enter a Number for Fuel Flow", , "Tanker Employment Tool"
        .SetFocus
    Else: Flow = .Value
    End If
End With
'make sure they added a true fuel capacity
With AircraftFCap
    If .Value = "" Or Not IsNumeric(.Value) Or .Value < 0 Then
        MsgBox "Please Enter a Number for Fuel Cap", , "Tanker Employment Tool"
        .SetFocus
    Else: Cap = .Value
```

```
    End If
End With

'make sure they added a true fuel reserve
With AircraftFRes
    If .Value = "" Or Not IsNumeric(.Value) Or .Value < 0 Then
        MsgBox "Please Enter a Number for Fuel Flow", , "Tanker Employment Tool"
        .SetFocus
    Else: Res = .Value
    End If
End With
'make sure they added a true offload
With AircraftFClimb
    If .Value = "" Or Not IsNumeric(.Value) Or .Value < 0 Then
        MsgBox "Please Enter a Number for Climb Fuel", , "Tanker Employment Tool"
        .SetFocus
    Else: Climb = .Value
    End If
End With
'Correct weight
With AircraftEmptyWT
    If .Value = "" Or Not IsNumeric(.Value) Or .Value < 0 Then
        MsgBox "Please Enter a Number for Aircraft Empty Weight", , "Tanker Employment Tool"
        .SetFocus
    Else: Weight = .Value
    End If
End With
With AircraftPPM
    If .Value = "" Or Not IsNumeric(.Value) Or .Value < 0 Then
        MsgBox "Please Enter a Number for Onload Rate", , "Tanker Employment Tool"
        .SetFocus
    Else: PPM = .Value
    End If
End With

'collect info dump into excel
Application.ScreenUpdating = False
Worksheets("Aircraft Info").Select

Rows(9).Select
Selection.Insert
With Range("A2")
    Index = 7
    .Offset(Index, 0) = AName
    .Offset(Index, 1) = TAS
    .Offset(Index, 2) = Flow
    .Offset(Index, 3) = Cap
    .Offset(Index, 4) = Res
    .Offset(Index, 5) = Climb
    .Offset(Index, 7) = Weight
End With
Range("A5:M5").Select
    Range(Selection, Selection.End(xlDown)).Select
    Selection.Sort Key1:=Range("A2"), Order1:=xlAscending, Header:=xlGuess, _
```

```
        OrderCustom:=1, MatchCase:=False, Orientation:=xlTopToBottom, _
        DataOption1:=xlSortNormal
Application.ScreenUpdating = True
Unload Me
End Sub

Private Sub AircraftCancel_Click()
    Unload Me
    Menu.Show
End Sub

MAIN MENU
Private Sub MenuCancel_Click()
    Dim Response As Variant
    Unload Me
    Response = MsgBox("Come again soon", vbOKOnly, "Tanker Employment Tool")
    End
End Sub

Private Sub TaskGo_Click()
Dim Task As Integer
With TaskList
    If .ListIndex <> -1 Then
        Task = .ListIndex
    Else
        MsgBox "Select a task from the list.", , "Tanker Employment Tool"
        .SetFocus
        Exit Sub
    End If
    Unload Me
    Select Case Task
        Case Is = 0
            AddAvailTankerForm.Show
        Case Is = 1
            AddBaseForm.Show
        Case Is = 2
            AddSortieForm.Show
        Case Is = 3
            EngagementForm.Show
        Case Is = 4
            OpenHelpForm.Show
        Case Is = 5
            InsertAircraftForm.Show
        Case Is = 6
            AddTankerForm.Show
        Case Is = 7
            Worksheets("Instructions").Select
            Unload Me
            End
        Case Is = 8
            Call LoadCarribean
        Case Is = 9
            Call LoadSWAsia
        Case Is = 10
```

```
        Call ResetSheet
      Case Is = 11
        Call Solve
  End Select
End With
End Sub
```

TABU SEARCH FORM

```
Private Sub CancelButton_Click()
  Unload Me
  End
End Sub

Private Sub SearchButton_Click()

With TabuTenure
  If .Value = "" Or Not IsNumeric(.Value) Then
    MsgBox "Please enter a number > 0", , "Tanker Employment Tool"
    .SetFocus
    Exit Sub
  Else: Worksheets("Miscellaneous").Range("C22") = .Value
  End If
End With
With Reactive
  If .Value = True Then
    Worksheets("Miscellaneous").Range("F22") = "YES"
  Else: Worksheets("Miscellaneous").Range("F22") = "NO"
  End If
End With
With SkipNumber
  If .Value = "" Or Not IsNumeric(.Value) Then
    MsgBox "Please enter a number > 0", , "Tanker Employment Tool"
    .SetFocus
    Exit Sub
  Else: Worksheets("Miscellaneous").Range("D22") = .Value
  End If
End With
With Large
  If .Value = True Then
    Worksheets("Miscellaneous").Range("E22") = "LARGE"
  Else: Worksheets("Miscellaneous").Range("E22") = "SMALL"
  End If
End With
Unload Me
End Sub
```

Appendix B. Complete Solutions

Table 23: Complete Sortie List Middle East Scenario

Sortie ID	Receiver Type	# of Receivers	Origin ICAO	Mission Type	Armament
A10_1	A/OA10	4	OKAJ	CAS (Close Air Support)	4
A10_2	A/OA10	4	OKAJ	CAS (Close Air Support)	4
A10_3	A/OA10	4	OKAJ	CAS (Close Air Support)	4
A10_4	A/OA10	4	OKAJ	CAS (Close Air Support)	4
A10_5	A/OA10	4	OKAJ	CAS (Close Air Support)	4
A10_6	A/OA10	4	OKAJ	CAS (Close Air Support)	4
F117_1	F117	2	OKAS	Strike	12
F117_2	F117	2	OKAS	Strike	12
F14_1	F14	4	XXX2	CAP (Combat Air Patrol)	3
F14_2	F14	4	XXX2	CAP (Combat Air Patrol)	3
F14_3	F14	4	XXX2	CAP (Combat Air Patrol)	3
F15_1	F15	4	OKAJ	CAP (Combat Air Patrol)	3
F15_2	F15	4	OKAJ	CAP (Combat Air Patrol)	3
F15_3	F15	4	OKAJ	CAP (Combat Air Patrol)	3
F15_4	F15	4	OBBS	CAP (Combat Air Patrol)	3
F15_5	F15	4	OBBS	CAP (Combat Air Patrol)	3
F15_6	F15	4	OBBS	CAP (Combat Air Patrol)	3
F15_7	F15	4	OTBH	CAP (Combat Air Patrol)	3
F15_8	F15	4	OTBH	CAP (Combat Air Patrol)	3
F15_9	F15	4	OTBH	CAP (Combat Air Patrol)	3
F15E_1	F15E	4	OTBH	Strike	6
F15E_2	F15E	4	OTBH	Strike	6
F15E_3	F15E	4	OTBH	Strike	6
F15E_4	F15E	4	OTBH	Strike	6
F15E_5	F15E	4	OTBH	Strike	6
F15E_6	F15E	4	OTBH	Strike	6
F16_1	F16	4	OKAJ	CAS (Close Air Support)	3.5
F16_2	F16	4	OKAJ	CAS (Close Air Support)	3.5
F16_3	F16	4	OKAJ	CAS (Close Air Support)	3.5
F16_4	F16	4	OKAJ	CAS (Close Air Support)	3.5
F16_5	F16	4	OKAJ	CAS (Close Air Support)	3.5
F16_6	F16	4	OKAJ	CAS (Close Air Support)	3.5
F16_7	F16	4	OKAJ	CAS (Close Air Support)	3.5
F16_8	F16	2	OKAJ	CAS (Close Air Support)	3.5
F16_9	F16	4	OKAS	Strike	4
F16_10	F16	4	OKAS	Strike	4
F16_11	F16	4	OKAS	Strike	4
F16_12	F16	4	OBBS	CAS (Close Air Support)	3.5
F16_13	F16	4	OBBS	CAS (Close Air Support)	3.5
F16_14	F16	4	OBBS	CAS (Close Air Support)	3.5

F16_15	F16	4	OBBS	CAS (Close Air Support)	3.5
F16_16	F16	4	OBBS	CAS (Close Air Support)	3.5
F16_17	F16	4	OBBS	CAS (Close Air Support)	3.5
F16_18	F16	4	OBBS	CAS (Close Air Support)	3.5
F16_19	F16	2	OBBS	CAS (Close Air Support)	3.5
F16_20	F16	4	XXX1	CAS (Close Air Support)	3.5
F16_21	F16	4	XXX1	CAS (Close Air Support)	3.5
F16_22	F16	4	XXX1	CAS (Close Air Support)	3.5
F16_23	F16	4	XXX1	CAS (Close Air Support)	3.5
F16_24	F16	4	XXX1	CAS (Close Air Support)	3.5
F16_25	F16	4	XXX1	CAS (Close Air Support)	3.5
F16_26	F16	4	LTAG	Strike	4
F16_27	F16	4	LTAG	Strike	4
F16_28	F16	4	LTAG	Strike	4
F16_29	F16	4	LTAG	CAS (Close Air Support)	3.5
F16_30	F16	4	LTAG	CAS (Close Air Support)	3.5
F16_31	F16	4	LTAG	CAS (Close Air Support)	3.5
F18_1	F18	4	XXX2	CAS (Close Air Support)	5
F18_2	F18	4	XXX2	CAS (Close Air Support)	5
F18_3	F18	4	XXX2	CAS (Close Air Support)	5
F18_4	F18	4	XXX2	CAS (Close Air Support)	5
F18_5	F18	4	XXX2	CAS (Close Air Support)	5
F18_6	F18	4	XXX2	CAS (Close Air Support)	5

Table 24: Complete Initial Feasible Solution Middle East Scenario

Sortie ID	Receiver Type	Num Aircraft	Sortie RP	RP Number	Origin Base	Anchor Area Assigned	Tanker Assigned
A10_1	A/OA10	4	1	1	AHMED AL JABER	31	1
A10_1	A/OA10	4	2	2	AHMED AL JABER	42	33
A10_1	A/OA10	4	3	3	AHMED AL JABER	31	1
A10_2	A/OA10	4	1	4	AHMED AL JABER	31	1
A10_2	A/OA10	4	2	5	AHMED AL JABER	1	13
A10_2	A/OA10	4	3	6	AHMED AL JABER	31	1
A10_3	A/OA10	4	1	7	AHMED AL JABER	31	1
A10_3	A/OA10	4	2	8	AHMED AL JABER	1	13
A10_3	A/OA10	4	3	9	AHMED AL JABER	31	1
A10_4	A/OA10	4	1	10	AHMED AL JABER	31	1
A10_4	A/OA10	4	2	11	AHMED AL JABER	2	17
A10_4	A/OA10	4	3	12	AHMED AL JABER	31	1
A10_5	A/OA10	4	1	13	AHMED AL JABER	30	2
A10_5	A/OA10	4	2	14	AHMED AL JABER	2	17
A10_5	A/OA10	4	3	15	AHMED AL JABER	30	2
A10_6	A/OA10	4	1	16	AHMED AL JABER	30	2
A10_6	A/OA10	4	2	17	AHMED AL JABER	2	17
A10_6	A/OA10	4	3	18	AHMED AL JABER	30	2
F117_1	F117	2	1	19	ALI AL SALEM AB	30	2
F117_1	F117	2	2	20	ALI AL SALEM AB	1	13
F117_1	F117	2	3	21	ALI AL SALEM AB	30	2
F117_2	F117	2	1	22	ALI AL SALEM AB	32	3
F117_2	F117	2	2	23	ALI AL SALEM AB	2	17
F117_2	F117	2	3	24	ALI AL SALEM AB	32	3
F14_1	F14	4	1	25	Harry S. Truman	29	6
F14_1	F14	4	2	26	Harry S. Truman	6	60
F14_1	F14	4	3	27	Harry S. Truman	29	6
F14_2	F14	4	1	28	Harry S. Truman	29	6
F14_2	F14	4	2	29	Harry S. Truman	6	60
F14_2	F14	4	3	30	Harry S. Truman	29	6
F14_3	F14	4	1	31	Harry S. Truman	32	3
F14_3	F14	4	2	32	Harry S. Truman	6	60
F14_3	F14	4	3	33	Harry S. Truman	29	6
F15_1	F15	4	1	34	AHMED AL JABER	33	4
F15_1	F15	4	2	35	AHMED AL JABER	6	60
F15_1	F15	4	3	36	AHMED AL JABER	32	3
F15_2	F15	4	1	37	AHMED AL JABER	33	4
F15_2	F15	4	2	38	AHMED AL JABER	7	59
F15_2	F15	4	3	39	AHMED AL JABER	29	6
F15_3	F15	4	1	40	AHMED AL JABER	28	19
F15_3	F15	4	2	41	AHMED AL JABER	7	59
F15_3	F15	4	3	42	AHMED AL JABER	33	4
F15_4	F15	4	1	43	BAHRAIN INTL	28	19
F15_4	F15	4	2	44	BAHRAIN INTL	7	59
F15_4	F15	4	3	45	BAHRAIN INTL	28	19
F15_5	F15	4	1	46	BAHRAIN INTL	27	20
F15_5	F15	4	2	47	BAHRAIN INTL	7	59

F15_5	F15	4	3	48	BAHRAIN INTL	27	20
F15_6	F15	4	1	49	BAHRAIN INTL	27	20
F15_6	F15	4	2	50	BAHRAIN INTL	8	58
F15_6	F15	4	3	51	BAHRAIN INTL	34	5
F15_7	F15	4	1	52	Al Udeid	34	5
F15_7	F15	4	2	53	Al Udeid	8	58
F15_7	F15	4	3	54	Al Udeid	35	21
F15_8	F15	4	1	55	Al Udeid	35	21
F15_8	F15	4	2	56	Al Udeid	8	58
F15_8	F15	4	3	57	Al Udeid	35	21
F15_9	F15	4	1	58	Al Udeid	26	22
F15_9	F15	4	2	59	Al Udeid	8	58
F15_9	F15	4	3	60	Al Udeid	26	22
F15E_1	F15E	4	1	61	Al Udeid	36	23
F15E_1	F15E	4	2	62	Al Udeid	6	60
F15E_1	F15E	4	3	63	Al Udeid	36	23
F15E_2	F15E	4	1	64	Al Udeid	25	24
F15E_2	F15E	4	2	65	Al Udeid	9	56
F15E_2	F15E	4	3	66	Al Udeid	25	24
F15E_3	F15E	4	1	67	Al Udeid	37	25
F15E_3	F15E	4	2	68	Al Udeid	9	56
F15E_3	F15E	4	3	69	Al Udeid	37	25
F15E_4	F15E	4	1	70	Al Udeid	24	26
F15E_4	F15E	4	2	71	Al Udeid	10	54
F15E_4	F15E	4	3	72	Al Udeid	24	26
F15E_5	F15E	4	1	73	Al Udeid	38	27
F15E_5	F15E	4	2	74	Al Udeid	10	54
F15E_5	F15E	4	3	75	Al Udeid	38	27
F15E_6	F15E	4	1	76	Al Udeid	23	28
F15E_6	F15E	4	2	77	Al Udeid	11	52
F15E_6	F15E	4	3	78	Al Udeid	23	28
F16_1	F16	4	1	79	AHMED AL JABER	33	4
F16_1	F16	4	2	80	AHMED AL JABER	9	56
F16_1	F16	4	3	81	AHMED AL JABER	28	19
F16_2	F16	4	1	82	AHMED AL JABER	26	22
F16_2	F16	4	2	83	AHMED AL JABER	10	54
F16_2	F16	4	3	84	AHMED AL JABER	28	19
F16_3	F16	4	1	85	AHMED AL JABER	36	23
F16_3	F16	4	2	86	AHMED AL JABER	11	52
F16_3	F16	4	3	87	AHMED AL JABER	34	5
F16_4	F16	4	1	88	AHMED AL JABER	25	24
F16_4	F16	4	2	89	AHMED AL JABER	11	52
F16_4	F16	4	3	90	AHMED AL JABER	26	22
F16_5	F16	4	1	91	AHMED AL JABER	37	25
F16_5	F16	4	2	92	AHMED AL JABER	11	52
F16_5	F16	4	3	93	AHMED AL JABER	24	26
F16_6	F16	4	1	94	AHMED AL JABER	38	27
F16_6	F16	4	2	95	AHMED AL JABER	12	50
F16_6	F16	4	3	96	AHMED AL JABER	24	26
F16_7	F16	4	1	97	AHMED AL JABER	39	29
F16_7	F16	4	2	98	AHMED AL JABER	12	50
F16_7	F16	4	3	99	AHMED AL JABER	23	28

F16_8	F16	2	1	100	AHMED AL JABER	39	29
F16_8	F16	2	2	101	AHMED AL JABER	8	58
F16_8	F16	2	3	102	AHMED AL JABER	39	29
F16_9	F16	4	1	103	ALI AL SALEM AB	39	29
F16_9	F16	4	2	104	ALI AL SALEM AB	12	50
F16_9	F16	4	3	105	ALI AL SALEM AB	31	1
F16_10	F16	4	1	106	ALI AL SALEM AB	39	29
F16_10	F16	4	2	107	ALI AL SALEM AB	12	50
F16_10	F16	4	3	108	ALI AL SALEM AB	34	5
F16_11	F16	4	1	109	ALI AL SALEM AB	22	30
F16_11	F16	4	2	110	ALI AL SALEM AB	13	43
F16_11	F16	4	3	111	ALI AL SALEM AB	39	29
F16_12	F16	4	1	112	BAHRAIN INTL	22	30
F16_12	F16	4	2	113	BAHRAIN INTL	13	43
F16_12	F16	4	3	114	BAHRAIN INTL	22	30
F16_13	F16	4	1	115	BAHRAIN INTL	22	30
F16_13	F16	4	2	116	BAHRAIN INTL	13	43
F16_13	F16	4	3	117	BAHRAIN INTL	22	30
F16_14	F16	4	1	118	BAHRAIN INTL	22	30
F16_14	F16	4	2	119	BAHRAIN INTL	13	43
F16_14	F16	4	3	120	BAHRAIN INTL	22	30
F16_15	F16	4	1	121	BAHRAIN INTL	22	30
F16_15	F16	4	2	122	BAHRAIN INTL	14	42
F16_15	F16	4	3	123	BAHRAIN INTL	40	31
F16_16	F16	4	1	124	BAHRAIN INTL	40	31
F16_16	F16	4	2	125	BAHRAIN INTL	14	42
F16_16	F16	4	3	126	BAHRAIN INTL	40	31
F16_17	F16	4	1	127	BAHRAIN INTL	40	31
F16_17	F16	4	2	128	BAHRAIN INTL	14	42
F16_17	F16	4	3	129	BAHRAIN INTL	40	31
F16_18	F16	4	1	130	BAHRAIN INTL	40	31
F16_18	F16	4	2	131	BAHRAIN INTL	14	42
F16_18	F16	4	3	132	BAHRAIN INTL	40	31
F16_19	F16	2	1	133	BAHRAIN INTL	21	35
F16_19	F16	2	2	134	BAHRAIN INTL	12	50
F16_19	F16	2	3	135	BAHRAIN INTL	21	35
F16_20	F16	4	1	136	Shaheed Mwaffaq	44	44
F16_20	F16	4	2	137	Shaheed Mwaffaq	15	41
F16_20	F16	4	3	138	Shaheed Mwaffaq	44	44
F16_21	F16	4	1	139	Shaheed Mwaffaq	44	44
F16_21	F16	4	2	140	Shaheed Mwaffaq	15	41
F16_21	F16	4	3	141	Shaheed Mwaffaq	44	44
F16_22	F16	4	1	142	Shaheed Mwaffaq	45	45
F16_22	F16	4	2	143	Shaheed Mwaffaq	15	41
F16_22	F16	4	3	144	Shaheed Mwaffaq	45	45
F16_23	F16	4	1	145	Shaheed Mwaffaq	45	45
F16_23	F16	4	2	146	Shaheed Mwaffaq	15	41
F16_23	F16	4	3	147	Shaheed Mwaffaq	45	45
F16_24	F16	4	1	148	Shaheed Mwaffaq	46	46
F16_24	F16	4	2	149	Shaheed Mwaffaq	16	40
F16_24	F16	4	3	150	Shaheed Mwaffaq	46	46
F16_25	F16	4	1	151	Shaheed Mwaffaq	46	46

F16_25	F16	4	2	152	Shaheed Mwaffaq	16	40
F16_25	F16	4	3	153	Shaheed Mwaffaq	46	46
F16_26	F16	4	1	154	INCIRLIK CDI	61	7
F16_26	F16	4	2	155	INCIRLIK CDI	16	40
F16_26	F16	4	3	156	INCIRLIK CDI	61	7
F16_27	F16	4	1	157	INCIRLIK CDI	61	7
F16_27	F16	4	2	158	INCIRLIK CDI	16	40
F16_27	F16	4	3	159	INCIRLIK CDI	61	7
F16_28	F16	4	1	160	INCIRLIK CDI	61	7
F16_28	F16	4	2	161	INCIRLIK CDI	17	39
F16_28	F16	4	3	162	INCIRLIK CDI	61	7
F16_29	F16	4	1	163	INCIRLIK CDI	61	7
F16_29	F16	4	2	164	INCIRLIK CDI	17	39
F16_29	F16	4	3	165	INCIRLIK CDI	61	7
F16_30	F16	4	1	166	INCIRLIK CDI	61	7
F16_30	F16	4	2	167	INCIRLIK CDI	17	39
F16_30	F16	4	3	168	INCIRLIK CDI	61	7
F16_31	F16	4	1	169	INCIRLIK CDI	61	7
F16_31	F16	4	2	170	INCIRLIK CDI	17	39
F16_31	F16	4	3	171	INCIRLIK CDI	61	7
F18_1	F18	4	1	172	Harry S. Truman	21	35
F18_1	F18	4	2	173	Harry S. Truman	18	38
F18_1	F18	4	3	174	Harry S. Truman	20	36
F18_2	F18	4	1	175	Harry S. Truman	20	36
F18_2	F18	4	2	176	Harry S. Truman	18	38
F18_2	F18	4	3	177	Harry S. Truman	20	36
F18_3	F18	4	1	178	Harry S. Truman	19	37
F18_3	F18	4	2	179	Harry S. Truman	18	38
F18_3	F18	4	3	180	Harry S. Truman	19	37
F18_4	F18	4	1	181	Harry S. Truman	41	32
F18_4	F18	4	2	182	Harry S. Truman	19	37
F18_4	F18	4	3	183	Harry S. Truman	41	32
F18_5	F18	4	1	184	Harry S. Truman	41	32
F18_5	F18	4	2	185	Harry S. Truman	41	32
F18_5	F18	4	3	186	Harry S. Truman	41	32
F18_6	F18	4	1	187	Harry S. Truman	42	33
F18_6	F18	4	2	188	Harry S. Truman	41	32
F18_6	F18	4	3	189	Harry S. Truman	42	33

Table 25: Complete Best Solution Middle East Scenario

Sortie ID	Receiver Type	Num Aircraft	Sortie RP	RP Number	Origin Base	Anchor Area Assigned	Tanker Assigned
A10_1	A/OA10	4	1	1	AHMED AL JABER	31	1
A10_1	A/OA10	4	2	2	AHMED AL JABER	42	33
A10_1	A/OA10	4	3	3	AHMED AL JABER	31	1
A10_2	A/OA10	4	1	4	AHMED AL JABER	31	1
A10_2	A/OA10	4	2	5	AHMED AL JABER	1	13
A10_2	A/OA10	4	3	6	AHMED AL JABER	31	1
A10_3	A/OA10	4	1	7	AHMED AL JABER	31	1
A10_3	A/OA10	4	2	8	AHMED AL JABER	1	13
A10_3	A/OA10	4	3	9	AHMED AL JABER	31	1
A10_4	A/OA10	4	1	10	AHMED AL JABER	31	1
A10_4	A/OA10	4	2	11	AHMED AL JABER	2	17
A10_4	A/OA10	4	3	12	AHMED AL JABER	31	1
A10_5	A/OA10	4	1	13	AHMED AL JABER	30	2
A10_5	A/OA10	4	2	14	AHMED AL JABER	2	17
A10_5	A/OA10	4	3	15	AHMED AL JABER	30	2
A10_6	A/OA10	4	1	16	AHMED AL JABER	30	2
A10_6	A/OA10	4	2	17	AHMED AL JABER	2	17
A10_6	A/OA10	4	3	18	AHMED AL JABER	30	2
F117_1	F117	2	1	19	ALI AL SALEM AB	30	2
F117_1	F117	2	2	20	ALI AL SALEM AB	1	13
F117_1	F117	2	3	21	ALI AL SALEM AB	30	2
F117_2	F117	2	1	22	ALI AL SALEM AB	32	3
F117_2	F117	2	2	23	ALI AL SALEM AB	2	17
F117_2	F117	2	3	24	ALI AL SALEM AB	32	3
F14_1	F14	4	1	25	Harry S. Truman	29	6
F14_1	F14	4	2	26	Harry S. Truman	6	60
F14_1	F14	4	3	27	Harry S. Truman	29	6
F14_2	F14	4	1	28	Harry S. Truman	29	6
F14_2	F14	4	2	29	Harry S. Truman	6	60
F14_2	F14	4	3	30	Harry S. Truman	29	6
F14_3	F14	4	1	31	Harry S. Truman	32	3
F14_3	F14	4	2	32	Harry S. Truman	6	60
F14_3	F14	4	3	33	Harry S. Truman	29	6
F15_1	F15	4	1	34	AHMED AL JABER	33	4
F15_1	F15	4	2	35	AHMED AL JABER	6	60
F15_1	F15	4	3	36	AHMED AL JABER	32	3
F15_2	F15	4	1	37	AHMED AL JABER	33	4
F15_2	F15	4	2	38	AHMED AL JABER	7	59
F15_2	F15	4	3	39	AHMED AL JABER	29	6
F15_3	F15	4	1	40	AHMED AL JABER	28	19
F15_3	F15	4	2	41	AHMED AL JABER	7	59
F15_3	F15	4	3	42	AHMED AL JABER	33	4
F15_4	F15	4	1	43	BAHRAIN INTL	28	19
F15_4	F15	4	2	44	BAHRAIN INTL	7	59
F15_4	F15	4	3	45	BAHRAIN INTL	28	19
F15_5	F15	4	1	46	BAHRAIN INTL	27	20
F15_5	F15	4	2	47	BAHRAIN INTL	7	59

F15_5	F15	4	3	48	BAHRAIN INTL	27	20
F15_6	F15	4	1	49	BAHRAIN INTL	27	20
F15_6	F15	4	2	50	BAHRAIN INTL	8	58
F15_6	F15	4	3	51	BAHRAIN INTL	34	5
F15_7	F15	4	1	52	Al Udeid	34	5
F15_7	F15	4	2	53	Al Udeid	8	58
F15_7	F15	4	3	54	Al Udeid	35	21
F15_8	F15	4	1	55	Al Udeid	35	21
F15_8	F15	4	2	56	Al Udeid	8	58
F15_8	F15	4	3	57	Al Udeid	35	21
F15_9	F15	4	1	58	Al Udeid	26	22
F15_9	F15	4	2	59	Al Udeid	8	58
F15_9	F15	4	3	60	Al Udeid	26	22
F15E_1	F15E	4	1	61	Al Udeid	36	23
F15E_1	F15E	4	2	62	Al Udeid	6	60
F15E_1	F15E	4	3	63	Al Udeid	36	23
F15E_2	F15E	4	1	64	Al Udeid	25	24
F15E_2	F15E	4	2	65	Al Udeid	9	56
F15E_2	F15E	4	3	66	Al Udeid	25	24
F15E_3	F15E	4	1	67	Al Udeid	37	25
F15E_3	F15E	4	2	68	Al Udeid	9	56
F15E_3	F15E	4	3	69	Al Udeid	37	25
F15E_4	F15E	4	1	70	Al Udeid	24	26
F15E_4	F15E	4	2	71	Al Udeid	10	54
F15E_4	F15E	4	3	72	Al Udeid	24	26
F15E_5	F15E	4	1	73	Al Udeid	38	27
F15E_5	F15E	4	2	74	Al Udeid	10	54
F15E_5	F15E	4	3	75	Al Udeid	38	27
F15E_6	F15E	4	1	76	Al Udeid	23	28
F15E_6	F15E	4	2	77	Al Udeid	11	52
F15E_6	F15E	4	3	78	Al Udeid	23	28
F16_1	F16	4	1	79	AHMED AL JABER	33	4
F16_1	F16	4	2	80	AHMED AL JABER	9	56
F16_1	F16	4	3	81	AHMED AL JABER	28	19
F16_2	F16	4	1	82	AHMED AL JABER	26	22
F16_2	F16	4	2	83	AHMED AL JABER	10	54
F16_2	F16	4	3	84	AHMED AL JABER	28	19
F16_3	F16	4	1	85	AHMED AL JABER	36	23
F16_3	F16	4	2	86	AHMED AL JABER	11	52
F16_3	F16	4	3	87	AHMED AL JABER	34	5
F16_4	F16	4	1	88	AHMED AL JABER	25	24
F16_4	F16	4	2	89	AHMED AL JABER	11	52
F16_4	F16	4	3	90	AHMED AL JABER	26	22
F16_5	F16	4	1	91	AHMED AL JABER	37	25
F16_5	F16	4	2	92	AHMED AL JABER	11	52
F16_5	F16	4	3	93	AHMED AL JABER	24	26
F16_6	F16	4	1	94	AHMED AL JABER	38	27
F16_6	F16	4	2	95	AHMED AL JABER	12	50
F16_6	F16	4	3	96	AHMED AL JABER	24	26
F16_7	F16	4	1	97	AHMED AL JABER	39	29
F16_7	F16	4	2	98	AHMED AL JABER	12	50
F16_7	F16	4	3	99	AHMED AL JABER	23	28

F16_8	F16	2	1	100	AHMED AL JABER	39	29
F16_8	F16	2	2	101	AHMED AL JABER	8	58
F16_8	F16	2	3	102	AHMED AL JABER	39	29
F16_9	F16	4	1	103	ALI AL SALEM AB	39	29
F16_9	F16	4	2	104	ALI AL SALEM AB	12	50
F16_9	F16	4	3	105	ALI AL SALEM AB	31	1
F16_10	F16	4	1	106	ALI AL SALEM AB	39	29
F16_10	F16	4	2	107	ALI AL SALEM AB	12	50
F16_10	F16	4	3	108	ALI AL SALEM AB	34	5
F16_11	F16	4	1	109	ALI AL SALEM AB	22	30
F16_11	F16	4	2	110	ALI AL SALEM AB	13	43
F16_11	F16	4	3	111	ALI AL SALEM AB	39	29
F16_12	F16	4	1	112	BAHRAIN INTL	22	30
F16_12	F16	4	2	113	BAHRAIN INTL	13	43
F16_12	F16	4	3	114	BAHRAIN INTL	22	30
F16_13	F16	4	1	115	BAHRAIN INTL	22	30
F16_13	F16	4	2	116	BAHRAIN INTL	13	43
F16_13	F16	4	3	117	BAHRAIN INTL	22	30
F16_14	F16	4	1	118	BAHRAIN INTL	22	30
F16_14	F16	4	2	119	BAHRAIN INTL	13	43
F16_14	F16	4	3	120	BAHRAIN INTL	22	30
F16_15	F16	4	1	121	BAHRAIN INTL	22	30
F16_15	F16	4	2	122	BAHRAIN INTL	14	42
F16_15	F16	4	3	123	BAHRAIN INTL	40	31
F16_16	F16	4	1	124	BAHRAIN INTL	40	31
F16_16	F16	4	2	125	BAHRAIN INTL	14	42
F16_16	F16	4	3	126	BAHRAIN INTL	40	31
F16_17	F16	4	1	127	BAHRAIN INTL	40	31
F16_17	F16	4	2	128	BAHRAIN INTL	14	42
F16_17	F16	4	3	129	BAHRAIN INTL	40	31
F16_18	F16	4	1	130	BAHRAIN INTL	40	31
F16_18	F16	4	2	131	BAHRAIN INTL	14	42
F16_18	F16	4	3	132	BAHRAIN INTL	40	31
F16_19	F16	2	1	133	BAHRAIN INTL	21	35
F16_19	F16	2	2	134	BAHRAIN INTL	12	50
F16_19	F16	2	3	135	BAHRAIN INTL	21	35
F16_20	F16	4	1	136	Shaheed Mwaffaq	44	44
F16_20	F16	4	2	137	Shaheed Mwaffaq	15	41
F16_20	F16	4	3	138	Shaheed Mwaffaq	44	44
F16_21	F16	4	1	139	Shaheed Mwaffaq	44	44
F16_21	F16	4	2	140	Shaheed Mwaffaq	15	41
F16_21	F16	4	3	141	Shaheed Mwaffaq	44	44
F16_22	F16	4	1	142	Shaheed Mwaffaq	45	45
F16_22	F16	4	2	143	Shaheed Mwaffaq	15	41
F16_22	F16	4	3	144	Shaheed Mwaffaq	45	45
F16_23	F16	4	1	145	Shaheed Mwaffaq	45	45
F16_23	F16	4	2	146	Shaheed Mwaffaq	15	41
F16_23	F16	4	3	147	Shaheed Mwaffaq	45	45
F16_24	F16	4	1	148	Shaheed Mwaffaq	46	46
F16_24	F16	4	2	149	Shaheed Mwaffaq	16	40
F16_24	F16	4	3	150	Shaheed Mwaffaq	46	46
F16_25	F16	4	1	151	Shaheed Mwaffaq	46	46

F16_25	F16	4	2	152	Shaheed Mwaffaq	16	40
F16_25	F16	4	3	153	Shaheed Mwaffaq	46	46
F16_26	F16	4	1	154	INCIRLIK CDI	61	7
F16_26	F16	4	2	155	INCIRLIK CDI	16	40
F16_26	F16	4	3	156	INCIRLIK CDI	61	7
F16_27	F16	4	1	157	INCIRLIK CDI	61	7
F16_27	F16	4	2	158	INCIRLIK CDI	16	40
F16_27	F16	4	3	159	INCIRLIK CDI	61	7
F16_28	F16	4	1	160	INCIRLIK CDI	61	7
F16_28	F16	4	2	161	INCIRLIK CDI	17	39
F16_28	F16	4	3	162	INCIRLIK CDI	61	7
F16_29	F16	4	1	163	INCIRLIK CDI	61	7
F16_29	F16	4	2	164	INCIRLIK CDI	17	39
F16_29	F16	4	3	165	INCIRLIK CDI	61	7
F16_30	F16	4	1	166	INCIRLIK CDI	61	7
F16_30	F16	4	2	167	INCIRLIK CDI	17	39
F16_30	F16	4	3	168	INCIRLIK CDI	61	7
F16_31	F16	4	1	169	INCIRLIK CDI	61	7
F16_31	F16	4	2	170	INCIRLIK CDI	17	39
F16_31	F16	4	3	171	INCIRLIK CDI	61	7
F18_1	F18	4	1	172	Harry S. Truman	21	35
F18_1	F18	4	2	173	Harry S. Truman	18	38
F18_1	F18	4	3	174	Harry S. Truman	20	36
F18_2	F18	4	1	175	Harry S. Truman	20	36
F18_2	F18	4	2	176	Harry S. Truman	18	38
F18_2	F18	4	3	177	Harry S. Truman	20	36
F18_3	F18	4	1	178	Harry S. Truman	19	37
F18_3	F18	4	2	179	Harry S. Truman	18	38
F18_3	F18	4	3	180	Harry S. Truman	19	37
F18_4	F18	4	1	181	Harry S. Truman	41	32
F18_4	F18	4	2	182	Harry S. Truman	19	37
F18_4	F18	4	3	183	Harry S. Truman	41	32
F18_5	F18	4	1	184	Harry S. Truman	41	32
F18_5	F18	4	2	185	Harry S. Truman	41	32
F18_5	F18	4	3	186	Harry S. Truman	41	32
F18_6	F18	4	1	187	Harry S. Truman	42	33
F18_6	F18	4	2	188	Harry S. Truman	41	32
F18_6	F18	4	3	189	Harry S. Truman	42	33

Table 26: Computation Time Statistics Full Candidate List

Computation Time Statistics (Full Candidate List)				
Caribbean Scenario	Min	Mean	Max	Std Dev
Initial	5.6772087	5.7212	5.8552755	0.11
Best	67.730598	67.8972	68.071498	0.13
Middle East Scenario				
Initial	123.92655	125.5701	127.14458	1.34
Best	1450.3583	1453.08	1456.4191	2.03

Table 27: Computation Time Statistics Half Candidate List

Computation Time Statistics (Half Candidate List)				
Caribbean Scenario	Min	Mean	Max	Std Dev
Initial	2.3655036	2.8606	3.2529308	0.12
Best	32.252666	33.779701	33.967407	0.14
Middle East Scenario				
Initial	58.732963	63.419242	74.790931	1.32
Best	483.45276	719.34653	770.59209	2.01

Bibliography

AFDD 2-6.2. *Air Refueling: Air Force Doctrine Document 2-6.2*, Part of Joint Publication 3-17, Joint Tactics, Techniques, and Procedures for Air Mobility Operations. Washington: HQ USAF, 1999.

Air Force Link. *KC-10 Extender Fact Sheet*. Online Publication. http://www.af.mil/factsheets/factsheet.asp?fsID=109. October 2003.

Annaballi, RonJon. *A Multiple Ant Colony Metaheuristic for the Air Refueling Tanker Assignment Problem*. MS Thesis. AFIT/GOR/ENS/02M-01, School of Engineering and Management, Air Force Institute of Technology, Wright-Patterson AFB OH, March 2002.

Barceló J. and J. Casanovas. "A Heuristic Lagrangean Algorithm for the Capacitated Plant Location Problem," *European Journal of Operational Research*, 15: 212-226, (1984).

Battiti, R. and G. Tecchiolli. "The Reactive Tabu Search," *ORSA Journal on Computing*, 6 (2): 126-140, (Spring 1994).

Ben-Daya, M. and Al-Fawzan, M. "A Tabu Search Approach for the Flow Shop Problem," *ORSA Journal on Operations Research Society*, 109: 88-95, (1998).

Böke, Cem. *Combining and Analyzing the Tanker and Aircrew Scheduling Heuristics*. MS Thesis. AFIT/GOR/ENS/03M-04, School of Engineering and Management, Air Force Institute of Technology, Wright-Patterson AFB OH, March 2003.

Boukhtouta A., Bedrouni A., Berger J., et. al. "A Survey of Military Planning Systems." 6 Jul 2004, Proc. of the 9th ICCRTS International Command and Control Research and Technology Symposium. Copenhagen, Denmark. 14 September 2004.

Brown, Darin T. *Routing Unmanned Aerial Vehicles While Considering General Restricted Operating Zones*. MS Thesis. AFIT/GOR/ENS/01M-04, School of Engineering and Management, Air Force Institute of Technology, Wright-Patterson AFB OH, March 2001.

Calhoun, Kevin M. *A Tabu Search for Scheduling and Rescheduling Combat Aircraft*. MS Thesis. AFIT/GOR/ENS/00M-06, School of Engineering and Management, Air Force Institute of Technology, Wright-Patterson AFB OH, March 2000.

Capehart, Shay R. *A Tabu Search Metaheuristic for the Air Refueling Tanker*

Assignment Problem. MS Thesis. AFIT/GOR/ENS/00M-07, School of Engineering and Management, Air Force Institute of Technology, Wright-Patterson AFB OH, March 2000.

Carlton, W. B. *A Tabu Search to the General Vehicle Routing Problem*. PhD dissertation. University of Texas at Austin, Austin TX. 1995.

Carlton, W. B. & Barnes, J. W. "Solving the Traveling-Salesman Problem with Time Windows Using Tabu Search", *IIE Transactions*, 28 (8): 617-629 (1996).

Cortinhal, Maria João and Maria Eugénia Captivo. "Upper and Lower Bounds for the Single Source Capacitated Location Problem." *European Journal of Operational Research*, 151: 333-351, (1993).

Cortinhal, Maria João and Maria Eugénia Captivo. "Genetic Algorithms for the Single Source Capacitated Location Problem," *Metahueristics: Computer Decision-Making*. Boston: Kluwer Academic Publishers, 2004. pp 187-216.

Cornuejols, G., R. Sridharan, and J.M. Thizy. "A Comparison of Heuristics and Relaxations for the Capacitated Plant Location Problem." *European Journal of Operational Research*, 50: 280-297, (1991).

Fiho V.J.M.F and R.D. Galvão. "A Tabu Search Heuristic for the Concentrator Location Problem," *Location Science*, 6: 189-209, (1998).

Gendreau, M., Laporte, G. & Potvin, G. *Vehicle Routing: Modern Heuristics*, Chichester: John Wiley & Sons, 1997.

Glover, Fred and M. Laguna. Ch 3. *Modern Heuristic Techniques for Combinatorial Problems*: 151-188. Ed. Colin R. Reeves, London: McGraw-Hill Book Company, 1995.

Glover, Fred and M. Laguna. *Tabu Search*. Boston: Kluwer Academic Publishers, 1997.

Harder, Robert. "OpenTS, An Open Source Framework for Tabu Search," Presentation. IBM Watson Research Center. Yorktown, New York. April 27, 2001.

Holmberg, K. "On the Convergence of Cross Decomposition." *Mathematical Programming*, 42: 269-296 (1990).

Iannuzzi, Phillip A. "50 Years Without Air Refueling Doctrine", Airlift/Tanker Quarterly, 5 (2), (Spring 1997).

Iannuzzi, Phillip A. "50 Years Without Air Refueling Doctrine – Part II", *Airlift/Tanker Quarterly*, 5 (3), (Summer 1997).

Jain, A. and Meeran, S. “Deterministic Job-Shop Scheduling: Past, Present and Future.” *European Journal of Operational Research*, 113: 390-434 (1999).

Jackson, Leroy A. *Facility Location using Cross Decomposition*. MS Thesis. Naval Postgraduate School. Monterey, CA. 1995.

Keuhn A. and M.J. Hamburger, “A Heuristic Program for Locating Warehouses,” *Management Science*, 9: 643-666, (1963).

Logicon. “Combined Mating and Ranging Planning System Overview.” Slide Presentation, Information Technology Group, Logicon Inc., 1996.

Michalewicz, Zbigniew and David B. Fogel. *How to Solve it: Modern Heuristics*. 2nd Ed. London: Springer-Verlag 2004**.**

Merriam-Webster Inc. *Webster’s School Dictionary*. Springfield MA: Merriam-Webster, 1986.

Moseley, Michael T. *Operation Iraqi Freedom: By the Numbers.* USCENTAF 30 April 2003 Report.. 12 January 2005 http://www.globalsecurity.org/military/library/report/2003/uscentaf_oif_report_30apr2003.pdf..

Neebe A.E. and M.R. Rao. “An Algorithm for the Fixed-Charge Assigning Users to Sources Problem,” *Journal of the Operational Research Society*, 34 (11): 1107-1113, (1983).

Newman, Richard J. “Tankers and Lifters for a Distant War.” *Air Force Magazine*, 85 (1): 56-60 (January 2002).

Pinedo Michael. *Scheduling: Theory, Algorithms, and Systems*. Englewoods Cliffs NJ: Prentice Hall, 1995.

Russina, B. and Ruthsatz B. *The Quick Look Tool for Tanker Deployment: Technical report*, St. Louis MO: Center for Optimization and Semantic Control, Washington University, 1999.

Savelsbergh, M. W. P. & Goetschalckx, M. “A Comparison of the Efficiency of Fixed versus Variable Vehicle Routes.” *Journal of Business Logistics*, 16 (1): 163-187, (1992).

Simpson, Richard. “Command of Theater Air Mobility Forces During the Air War Over Serbia.” *Airlift/Tanker Quarterly*, 8 (3): 10-13 (Summer 2000).

Sun Tzu. *The Art of War*. Boston: Shambala Publications Inc, 1988.

Tekelioglu, Umit H. *A Reactive Tabu Search Metaheuristic Extension of the Air Refueling Tanker Assignment Problem*. MS Thesis. AFIT/GOR/ENS/01M-16. School of Engineering of Management, Air Force Institute of Technology, Wright-Patterson AFB OH, March 2001.

Wiley, Victor D. *The Aerial Fleet Refueling Problem*. PhD Dissertation. University of Texas at Austin, Austin TX, 2001.

Woodruff, D.L. and Zemel, E. "Hashing Vectors for Tabu Search," *Annals of Operations Research,* 41: 123-137, (Issue 1-4 1993).

Zanakis, S. H. and J. R. Evans. "Heuristic 'Optimization': Why, When, and How to Use It," *Interfaces*, 11 (5): 84-91 (October 1981).

Vita

Second Lieutenant Jeffrey Miller graduated from Ramona High School in Ramona, California. He accepted a Congressional appointment to the United States Air Force Academy in Colorado Springs, Colorado where he graduated with a Bachelor of Science degree in Economics and Operations Research in May 2003. He graduated with academic distinction after being on the Dean's List for all eight semesters and received a regular commission in the United States Air Force.

His first assignment was to achieve his graduate degree in Operations Research. In August 2003, he entered the Graduate School of Engineering and Management, Air Force Institute of Technology. Upon graduation, he will be assigned to The Air Force Studies and Analyses Agency at the Pentagon.

REPORT DOCUMENTATION PAGE	*Form Approved* *OMB No. 074-0188*

1. REPORT DATE *(DD-MM-YYYY)*	**2. REPORT TYPE**	**3. DATES COVERED** *(From – To)*
09-03-2005	**Master's Thesis**	Sep 2003 - Mar 2005

4. TITLE AND SUBTITLE
A CAPACITATED FACILITY LOCATION APPROACH FOR THE TANKER EMPLOYMENT PROBLEM

5a. CONTRACT NUMBER

5b. GRANT NUMBER

5c. PROGRAM ELEMENT NUMBER

6. AUTHOR(S)

Miller, Jeffrey R., Second Lieutenant, USAF

5d. PROJECT NUMBER

5e. TASK NUMBER

5f. WORK UNIT NUMBER

7. PERFORMING ORGANIZATION NAMES(S) AND ADDRESS(S)

Air Force Institute of Technology
Graduate School of Engineering and Management (AFIT/EN)
2950 Hobson Street, Building 642
WPAFB OH 45433-7765

8. PERFORMING ORGANIZATION REPORT NUMBER

AFIT/GOR/ENS/05-12

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

AFOSR/NM, Attn: Dr. Neal Glassman, 4015 Wilson Boulevard Mail Room 713
Arlington, Virginia 22203-1954
Phone: (703) 696 - 9548
e-mail: neal.glassman@afosr.af.mil

10. SPONSOR/MONITOR'S ACRONYM(S)

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

12. DISTRIBUTION/AVAILABILITY STATEMENT

APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED.

13. SUPPLEMENTARY NOTES

14. ABSTRACT

Air refueling is conducted to provide rapid response, increased range, and extended airborne operations for bombers, fighters, airlift, command and control, and intelligence, surveillance, and reconnaissance aircraft. The planning and scheduling of limited tanker resources during employment operations is a major concern for Air Mobility Command (AMC). AMC does not currently have a simple tool that runs in a short amount of time to aid in planning operations.

The tool developed allows AMC to input several sorties consisting of various aircraft types and armaments. Each sortie contains a base of origin, and is assumed to be attacking or patrolling in an engagement zone defined by the user. The user is also able to specify the locations of military tanker aircraft. The main goal of the tool is to assign the tankers to anchor areas, surrounding the engagement zone so that all receivers can be refueled during their attack operations. Secondary goals include minimizing the number of tankers required (or maximizing the number of receivers supported), and limiting the total flight distance for the tanker aircraft. The TET tool uses the heuristic technique tabu search to determine an assignment of tankers and sorties to anchor areas during employment.

15. SUBJECT TERMS

Tabu Search, Heuristics, Metaheuristics, Tanker Scheduling, Employment Problem

16. SECURITY CLASSIFICATION OF:			**17. LIMITATION OF ABSTRACT**	**18. NUMBER OF PAGES**	**19a. NAME OF RESPONSIBLE PERSON**
a. REPORT	**b. ABSTRACT**	**c. THIS PAGE**			Dr. James T. Moore, AFIT/ENS
U	**U**	**U**	**UU**	152	**19b. TELEPHONE NUMBER** *(Include area code)* (937) 255-3636, ext 4528; e-mail: james.moore@afit.edu