Solving the Aerial Fleet Refueling Problem using Group Theoretic Tabu Search

J. Wesley Barnes and Victor D. Wiley
Graduate Program in Operations Research and Industrial Engineering
The University of Texas at Austin

Department of Operational Sciences
Air Force Institute of Technology

James T. Moore

Department of Operational Sciences
Air Force Institute of Technology

David M. Ryer

Air Mobility Command
United States Air Force
Scott AFB, IL

Abstract

The Aerial Fleet Refueling Problem (AFRP) is concerned with the efficient and effective use of a heterogeneous set of tanker (refueling) aircraft, located at diverse geographical locations, in the required operations associated with the deployment of a diverse fleet of military aircraft to a foreign theater of activity. Typically, the "receiving" aircraft must traverse great distances over large bodies of water and/or over other inhospitable environs where no ground based refueling resources exist. Lacking the ability to complete their flights without refueling, each receiving aircraft must be serviced one or more times during their deployment flights by means of in-flight refueling provided by one of the available tanker aircraft. The receiving aircraft, aggregated into receiver groups (RGs) that fly together, have stipulated departure and destination bases and each RG's arrival time is bounded by a stated desired earliest and latest time. The excellence of a suggested solution to this very challenging decision making problem is measured relative to a rigorously defined hierarchical multicriteria objective function.

This paper describes how the AFRP for the Air Mobility Command (AMC), Scott AFB, IL is efficiently solved using Group Theoretic Tabu Search (GTTS). GTTS uses the symmetric group on n letters (S_n) and applies it to this problem using the Java^{TM} language.

1 Introduction

Air Refueling, the in-flight transfer of fuel between tanker and receiver aircraft, provides rapid response, increased range, and extended airborne operations for all aircraft. The capability to perform air refueling makes the United States (US) the dominant air power in the world today. This capability, coupled with the ability to efficiently employ air refueling assets, is essential for the US to maintain this dominance (AFDD 2-6.2 1999). The Air Mobility Command (AMC) of the United States Air Force (USAF) is responsible for determining the use of the US tanker fleet to meet the air refueling needs of the Air Force, Navy, Marines, US allies, and coalition partners. As part of this responsibility, AMC plans, schedules, tasks, and executes operations involving the use of its forces. Part of AMC's planning encompasses the intertheater movement of forces from the US to areas around the globe. This “deployment” of forces and its accompanied air refueling requirement is known as the Aerial Fleet Refueling Problem (AFRP).

As the agency responsible for air refueling, AMC addresses many questions involving the allocation and use of tankers. During Operations Desert Shield and Desert Storm, approximately 400 tankers off-loaded over 1.2 billion pounds of fuel to over 80,000 aircraft while flying over 30,000 sorties and logging over 140,000 hours of flight time (Gulf War Air Power Survey 1993). The many planning scenarios, like Desert Shield/Desert Storm, that must be considered create complex sets of questions whose answers demand the use of powerful analytical tools. Among these tools are the Combined Mating and Ranging Planning System (CMARPS), the Quick-Look Tool (QLT), and the Tanker Assignment Planning (TAP) Tool.

CMARPS is a computer program that helps analyze, plan, and schedule deployment of tankers and receiver aircraft in support of immediate and anticipated military operations. Unfortunately, this tool can take many weeks if not months to produce meaningful results. The QLT is a simple spreadsheet model used for determining gross tanker capabilities for supporting deployments. The TAP Tool is a spreadsheet model used for assigning tankers to refueling points. The simplicity of the QLT and TAP Tool makes them incapable of addressing the problem in required levels of detail. AMC possesses a compelling need for a tool that will provide the detailed analysis of CMARPS within a planning horizon comparable to that required by the QLT. Given a deployment scenario, examples of overview questions that require answers are:

- How many tankers are required to meet the air refueling requirements?
- How quickly can all the receivers be deployed to their final destinations?
- How far do the tankers and receiver aircraft have to travel?
- How much fuel is burned by both tankers and receiver aircraft?

In order to meaningfully answer upper level questions like these, as well as more probing questions relating to the efficiency of planning operations, a great many detailed operational aspects must be addressed.

2 Problem Statement

We assume that the following information is given:

- a known set of tankers and their associated original beddown (starting) bases
- a known set of receiver aircraft, each with an initial departure base and a final destination base, where one or more receiver aircraft are aggregated to form Receiver Groups (RG's). Each RG has a stipulated desired earliest arrival time and latest arrival time
- a known set of RG starting and ending bases and tanker beddown bases
- a known set of flight characteristics for each aircraft including flight speed, altitude, take-off weight, fuel capacity, and fuel-burn rates
- a known set of tanker specific characteristics including fuel-offload capacity and fuel-offload rates

Figure 1 pictures an example of a scenario with two tankers and two RGs where the RGs' flight paths are displayed in red. Waypoints (WPTs) define the physical locations and start times where each refueling of an RG will take place. In Figure 1, each RG has two WPTs (circles) scheduled before arrival at a common destination base. The yellow pentagons represent the beddown bases of tankers 1 and 2.

Figure 1. A Small Scenario

For a given deployment, the following decisions compose the solution to the AFRP.

- the waypoints (WPTs)
- the tanker(s) that will serve each WPT
- how much fuel the assigned tanker(s) should deliver to a WPT

To this end, this paper will demonstrate how S_n will be used to represent the assignment of tankers to WPTs as well as the specific order that the WPTs will be visited (see Section 4). The amount of fuel delivered is a function of the RG, the fuel it has used since last receiving fuel, and how much fuel it needs to arrive at its final destination.

We further assume that the decision maker has the authority to (a) stipulate the departure times of all RGs and tankers and (b) to require both tankers and RGs to "orbit" at specified locations to satisfy WPT requirements in terms of timing and location. These two assumptions allow an AFRP solution to satisfy a number of constraints while exhibiting a number of desirable qualities to include deconflicting schedules and reducing the number of tankers required.

Given these assumptions, this paper concentrates on determining the “best” assignment and ordering of WPTs with tankers while generating a flyable schedule for the tankers and RGs alike.

The determination of “best” is influenced by a large number of factors. Among these factors are feasibility related criteria such as providing the required amount of fuel at each WPT and respecting the precedence relationships along a RG’s flight path. Other factors include optimality related criteria such as the number of tankers used, the distance flown by the tankers, and the arrival time of RGs at their destinations.

The AFRP objective function, as implemented in this research, is multicriteria and hierarchical. The hierarchical ordering of the associated criteria is subject to redefinition in accordance with perceived mission priorities. For the purposes of this research, the following criteria, in the order given, define the specific hierarchical objective function addressed.

Minimize:

1. the number of unescorted RGs requiring escort between WPTs
2. the number of WPTs not serviced by a tanker
3. the number of WPTs serviced out of order for a RG's flight path
4. the amount of required fuel not available
5. the amount of time spent by RGs and tankers in "orbit" at WPTs
6. the amount of RG late arrival time, i.e., where one or more RGs arrive later than a desired latest time
7. the number of tankers used
8. the amount of tanker mission time required
9. the total distance flown by tankers
10. the amount of fuel used by tankers
11. the amount of fuel off-loaded by tankers
12. the amount of fuel used by RGs

For any pair of solutions, superiority is determined by strict hierarchical criteria comparison starting at index 1 and traversing through increasing indices. At any index, a lower criterion value (bounded below by zero) implies a better solution and the comparison process is stopped. Two solutions are equivalent only if all 12 criteria are equivalent.

When Criteria 1 to 4 exceed zero, the solution is not feasible because of policy violation or failure of aircraft to complete their required flights. Criteria 5 through 12 were determined by Maj David Ryer of the Studies and Analysis Division of AMC.

Figure 2 displays a specific solution to the scenario of Figure 1. Using the cyclic notation of S_n (see Section 4), this solution would be (T1, WPT3, T1’,WPT1)(T2, WPT4, WPT2). In this solution, the first tanker, T1, services RG2 at WPT3 and returns to its beddown base, denoted T1’. It then continues operations by flying to WPT1, servicing RG1, and then returning once again to its beddown base. The second tanker, T2, services RG2 at WPT4, then flies to WPT2 and services RG1 before returning to its beddown base. In this solution, the assignment of tankers to WPTs influences the timing of other physical activities including the departure time of each RG and the initial and subsequent departure times of the tankers during their service to the deployment.

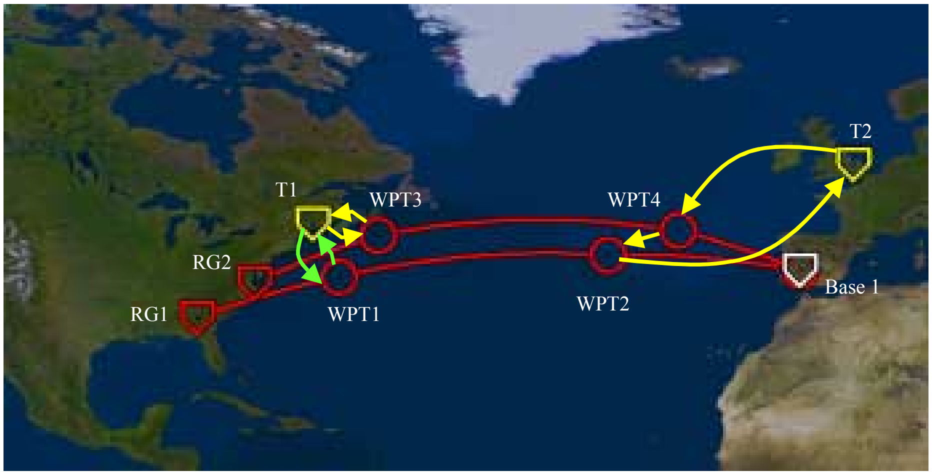

Figure 2. A Specific AFRP Solution

Thus, the solution to the AFRP is composed of a complex set of interrelated decisions involving time, space, and amounts of fuel. Usually, the effect of changing any individual decision will "ripple" through the AFRP structure forcing multiple associated changes in the solution.

One consideration not taken into account in the example of Figure 2 is that fighter aircraft require continuous escort (deployment support) while flying over large bodies of water. The added consideration of deployment support further restricts the assignment of tankers to WPTs by forcing a tanker to follow a RG along segments of the RG's flight path. Tankers that provide escort typically do not immediately return to their original base. Instead, they often continue escort duties until passing the receivers on to another available tanker or proceeding to another enroute base.

Figure 3 provides an example of RG1 and RG2 being escorted during part of their flight to Base 1. T1 escorts RG1 from WPT1 to WPT2 and lands at Base 1 while T2 escorts RG2 from WPT3 to WPT4 and then returns to its beddown base. In this example, T1 is serviced at Base 1 and then returns to its beddown base. In other scenarios, T1 could have been refueled at Base 1 and continued to service the deployment without returning to its beddown base.

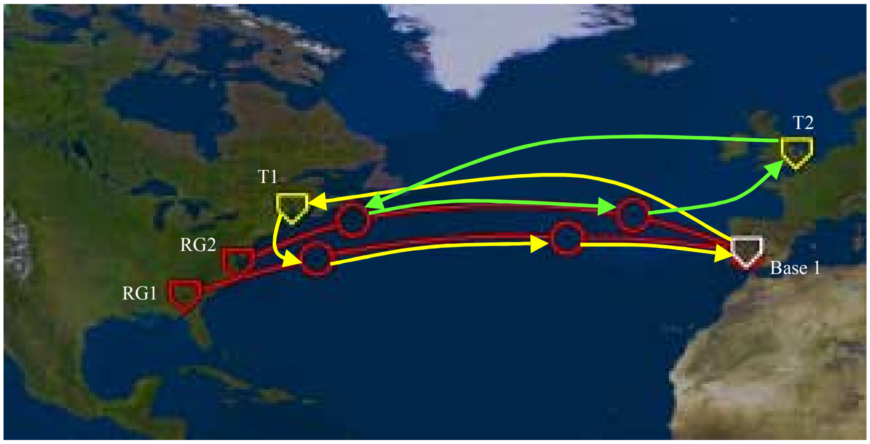

Figure 3. RG Escort Example

The AFRP solution is constrained by a large number of limiting factors. The safety of the crews and aircraft associated with the deployment is dominant, i.e., no tanker or receiver aircraft should have to divert from its flight path due to lack of fuel. Many other constraints must also be satisfied. For example, a tanker has limited fuel capacity and its crew has flight duration restrictions which affect the crew-tanker ability to travel long distances and to provide fuel. Certain bases have limited capacity for resident aircraft (known as *maximum on ground* or MOG). In the scenarios considered in this research, the important constraint of MOG was not a consideration, but could be addressed in future analysis. A tanker requires time to fly between any two locations and time to perform midair refueling. Hence, all tanker WPT assignments must be limited to those where the tanker is physically capable of being present at the specified WPT time.

The AFRP is unique, complicated, and extremely difficult to model and solve when viewed in its full practical context. As is discussed in Wiley (2001), the AFRP can be related to other classical combinatorial optimization problems by relaxing specific constraints or by fixing selected decision variables. Perhaps the most closely associated classical problem is a variation of the multi-vehicle, multi-depot Vehicle Routing Problem (VRP).

In the AFRP, we have finite capacity heterogeneous vehicles located at multiple depots (with route length/route duration upper bounds) that are required to deliver product to customers (such deliveries require a finite service time). In the notation of Carlton (1995), the AFRP is a variation of problem type $\alpha = (MV\overline{H}, MD, VRP, RL)$ which is known to be NP-hard (Gendreau, Laporte, Potvin, 1997).

However, there are several additional considerations present in the AFRP that are not present in problem α. These are :

1. In problem α, the customers' locations are fixed in space, requiring only that the customers be partitioned among the vehicles and that each vehicle's customers be ordered for service. Further, the amount of product to be delivered to each customer is a known 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.

   As in problem α, we must stipulate the responsible vehicle (tanker) and the ordering of any delivery. In addition, for all RGs, we must also stipulate the spatial location (longitude, latitude, and altitude in 3-dimensional space) and 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 WPT decisions are the decisions on the takeoff time of each RG and the possibly multiple takeoff times of each tanker.

If we desired to solve the AFRP by forcing it to be equivalent to problem α, all considerations of event timing must be relaxed; the spatial location and fuel requirement of each WPT must be fixed known constants; and each WPT becomes equivalent to a separate customer in the problem α formulation.

Problem α may be brought closer, in some respects, to the AFRP through the inclusion of a "time window" constraint on each of the WPTs, i.e., the stipulation of an earliest and latest time that refueling can begin at each WPT (customer).

The stipulation of time windows on the WPT refuelings would reintroduce 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.

The primary objective of the research documented in this paper was to develop methods for producing an "ensemble" of excellent solutions to any instance of the AFRP. To achieve this objective, a Group Theoretic Tabu Search (GTTS) approach was used. Our GTTS makes use of adaptive tabu search (Dell' Amico and Trubian 1993, Glover and Laguna 1997) to dynamically update the memory structures, promote diversification, and remove the need to empirically "tune" a static tabu tenure. GTTS represents a solution to the AFRP as an element from the symmetric group on n letters, S_n (Isaacs 1994), and creates move neighborhoods that can be represented by S_n acting under conjugation or multiplication upon itself. To address the issue of reusability and portability, the GTTS approach was implemented using the Java$^{\text{TM}}$ programming language. Our implementation makes extensive use of Wiley's (2001) Java$^{\text{TM}}$ class library for S_n. Secondary research goals were to investigate the effects of selecting different move neighborhoods both from a static and dynamic context.

How these goals were accomplished is discussed in detail in the remainder of this paper.

3 Literature Review

CMARPS was originally built in 1982 and has continued to evolve to its present form. It is currently maintained for AMC by the Northrop Grumman Company. CMARPS is a deterministic computer planning system that assists analysts and warplanners in developing and scheduling the deployment of tankers and receivers during peacetime, crisis, contingency, and wartime operations. Unfortunately, because of the great number of scenarios, as well as their formidable size, that must be explicitly constructed, this tool can take several analysts many weeks to produce meaningful results. An extremely large multivolume user's manual is available only in hardcopy and provides no detail on the methodology employed by CMARPS.

While CMARPS can assist in providing extensive, detailed, and accurate data for predicting receiver and tanker aircraft mission requirements, CMARPS marked complexities make quick and effective use difficult even for highly experienced users. The limitations of CMARPS caused several later studies that attempted to improve on these shortcomings.

With the aforementioned problems and time limits associated with CMARPS, AMC has been sponsoring research into aerial refueling for years. Initial attempts include Yamani (1986) and Hostler (1987). Yamani's research was limited to (a) consideration of RGs where only one WPT was required and (b) consideration of single aircraft that required two WPTs. Hostler (1987) allowed a tanker to refuel at most two WPTs and ignored the travel distance and fuel consumption limitations associated with the tankers.

More recent research includes Russina and Ruthsatz (1999) and Capehart (2000). Russina and Ruthsatz's QLT attempted to provide a more responsive program than CMARPS. The goals of this spreadsheet tool were to quickly estimate the number of tankers needed for a deployment and then determine how quickly that deployment could be achieved. The QLT incorporated several simplifying assumptions. Among these were (a) constant flight speeds for all aircraft, (b) the refueling tanker must provide any required escort duties, (c) all tankers were identical, (d) only one tanker could be assigned to a WPT (regardless of fuel requirements), (e) each tanker must return to its home base after a single refueling, and (f) the locations of all WPTs and the amount of fuel required at the WPT are assumed to be known constants and are part of the input data. The most limiting assumption in the QLT was that, when escort was not required, all tankers were assumed to take the same constant amount of time to travel to a WPT, complete refueling, and return to homebase. Thus, the QLT did not explicitly take into account the very important limitations of actual tanker travel and tanker fuel use. Useful for determining initial estimates, the QLT is incapable of addressing the problem in required levels of detail.

The work of Capehart (2000) extends the QLT through the use of a tabu search (TS) approach (Glover and Laguna 1997) within the TAP Tool. Capehart (2000) models the AFRP as an Assignment Problem (Kuhn 1955) with time windows. This formulation makes a number of simplifying assumptions which limit the solutions it produces. Among these limiting assumptions are: (a) a tanker providing fuel to a RG requiring escort must escort the RG to the next WPT, (b) each tanker must return to its home base after a single refueling, (c) light aircraft are escorted over their entire flight, not just when over large bodies of water, (d) the approach does not explicitly account for tanker reuse, but rather corrects time conflicts involving any individual tanker using a penalty structure, and (e) the locations of all WPTs and the amount of fuel required at the WPT are calculated based on a percentage of the full capacity of the RG.

The GTTS method described below possesses none of the above limitations. The next section provides a brief overview of how group theory and in particular, S_n, are incorporated into GTTS. The reader familiar with these topics may find this section superfluous.

4 Algebraic Concepts

A *group* is made up of any set of objects and an operation acting on this set which satisfies the properties of *closure, associativity, identity, and invertibility* (g^{-1} is the inverse of group element g). Well-known examples of groups include integers under addition, positive real numbers under multiplication, and n x n invertible matrices under matrix multiplication. Another group of particular interest is the Symmetric Group on n-letters (S_n).

S_n is the group of permutations of the set $\{1,2,...,n\}$ of integers from 1 to n (Artin 1991). Specific elements of S_n may be represented in the long or cyclic form. The long form appears as a matrix with 2 rows and n columns. The first row represents the index position of a letter within the permutation. The second row represents the letter located at the index position of the first row. The cyclic form appears as a set of cycles delineated by parentheses. Here is the same permutation taken from S_6: first in the long form and then in the cyclic form[1]:

$$\begin{pmatrix} 1 & 2 & 3 & 4 & 5 & 6 \\ 3 & 6 & 5 & 4 & 1 & 2 \end{pmatrix} \leftrightarrow (1,3,5)(2,6)(4) \equiv (1,3,5)(2,6)$$

Figure 4. Permutation Representations

Permutations are composed of cycles and an m-cycle has m letters. A 2-cycle is also called a transposition and an n-cycle is a permutation with a single cycle containing all the letters of S_n (equivalent to a 1-TSP tour). Every permutation is a unique product of disjoint cycles, i.e., cycles sharing no common cities. The number of cycles and the cycle sizes define the permutation's cycle structure. The cycles partition the letters of S_n into mutually exclusive sets. The placement of letters within each cycle determines the ordering. Thus, S_n provides a means to represent any partitioning and ordering problem solution.

The AFRP will make use of S_n's ability to represent any partitioning and ordering problem by specifically assigning tankers and WPTs to cycles. Each cycle will then represent the assignment of WPTs to a single tanker and the specific order that these WPTs will be visited.

The operation associated with the collection of permutations of the set $\{1,2,\ldots,n\}$ satisfying the conditions necessary for a group is *function composition*. This research uses multiplication and conjugation as forms of function composition.

Multiplication and conjugation of S_n will be used extensively by the AFRP as the means to assign WPTs to tankers and order the WPTs.

4.1 Multiplication

Let $p = (1,3,5)(2,6)$, the permutation from the above example. Let $q = (4,6,5)$, be another permutation. The product pq is the composite function (multiplication) produced by applying p and then q, i.e., for letter x, $xpq=(xp)q$. If the letter of interest is 3, then $3pq = (3p)q = 5q = 4$, and so under pq, 3 is mapped into 4. Permutation multiplication is not commutative since $3qp =$

---

[1] “Unit” cycles, i.e. (4), are customarily dropped from the cyclic representation.

$(3q)p = 3p = 5$, and, therefore, $pq \neq qp$. Since a permutation represents a 1 to 1 mapping from the letters $\{1,...,n\}$ onto itself, multiplication is associative and closed. The inverse of a permutation reverses each cycle, i.e. $(1,3,5)^{-1} = (1,5,3)$, and the identity permutation maps each letter into itself, i.e., is composed of n unit cycles. Thus, the 4 properties of a group are satisfied and the set of all $n!$ permutations under multiplication is S_n.

4.2 Conjugation

Conjugation is a special form of multiplication that takes the form $q^{-1}pq$. Conjugation also satisfies the properties of an operation acting upon a set necessary for a group. Conjugation has the additional quality of maintaining the current cyclic structure of the permutation being acted upon, i.e. p. This quality allows the number of letters assigned to a cycle to remain constant while reassigning and reordering letters within the cycles.

4.3 Examples

Using the permutation p, if one wanted to move letter "5" from its current position and place it in between letters "2" and "6" then, using multiplication as defined in Section 4.1 and denoted by *, the following operation would be performed: $p*(1,6,5) = (1,3,5)(2,6)*(1,6,5) = (1,3)(2,5,6)$.

Likewise, if one wanted to rearrange the assignment and ordering of such that letter "3" occupied the position currently held by letter "6" and that "6" was placed after letter "5", then, using conjugation as defined in Section 4.2 and denoted by ^, the following operation would be performed: $p\hat{}(3,5,6) = (1,3,5)(2,6)\hat{}(3,5,6) = (1,5,6)(2,3)$.

S_n is particularly important in group theory since all groups can be isomorphically mapped into S_n. Hence, group theory can be thought of as the "algebra of permutations." For further insights, see Colletti (1999), Colletti, Barnes, and Dokov (1999), Colletti and Barnes (2000, 2001), and Barnes, Colletti, and Neuway (2002).

This concludes the short review of the mathematical background needed for the remainder of this paper. The next section presents a detailed discussion of the implementation of the adaptive GTTS methodology that was developed to solve the AFRP.

5. Implementation of the GTTS for the AFRP

Our implementation of GTTS for the AFRP is written in the Java^{TM} language and makes extensive use of the object-oriented view of that language. Wiley (2001) presents a detailed discussion of object-oriented programming and we have found this perspective to be very beneficial in our implementation.

In order to effectively construct a GTTS for the AFRP, five primary types of *implementation* objects are defined and used: (1) *Locations*; (2) *Aircraft*, (3) *RGs*; (4) *Tankers*; and (5) *Nodes.*

These objects store detailed information beyond the partitioning and ordering captured by S_n. The connection between S_n and these objects occurs through a mapping of node labels to the integer set. Based on the placement of these node labels (a permutation of S_n), the information, described in the following paragraphs, associated with tankers, RGs, and nodes will be determined. For example, in Figure 4 the permutation could be mapped by labels to the following solution: Two tankers (letters “1” and “2”) have been assigned to three WPTs (letters “3,5” and “6”, respectively). The first tanker (“1”) will serve its two assigned WPTs in order of placement of labels within its cycle, i.e. services letter “3” then letter “5”. Given this ordering, the timing and scheduling of activities will be determined.

A solution of the AFRP requires detailed information about (1) the physical locations of bases and WPTs, (2) the flight characteristics of aircraft, (3) the assignment of aircraft to RGs and the stipulation of the flight path that a RG will follow, and (4) the number of tankers available for refueling activities and their associated beddown bases. In addition, information about the actual AFRP solution is stored in the tanker and RG objects. This is accomplished in the following ways: (information not germane to a particular object is left blank)

Locations (Bases, WPTs): (a) unique ID, (b) code name, (c) coordinates, (d) MOG, (e) whether over open water (After all location objects are created, a symmetric distance matrix is generated using the “great circle” distances associated with every pair of locations (Capehart 2000)).

Aircraft (Light receivers, heavy receivers, tankers): (a) unique ID, (b) airframe type (light, heavy, tanker), (c) nominal true air speed, (d) total fuel capacity (e) fuel burn characteristics (required fuel reserve, nominal take-off fuel, nominal fuel flow coefficient, nominal altitude, empty weight, and nominal load weight)

Receiver Groups: (a) Unique ID, (b) Total fuel upload requirement, (c) List of aircraft IDs in the RG (determines escort requirement), (d) Flight Path Information (start and end base IDs, RG flight characteristics, Earliest Departure Time (EDT) & Latest Departure Time (LDT)),

(e) RG solution attributes

(e1) list of flight path WPT nodes
(e2) arrival, service, and departure times at each WPT node
(e3) amount of fuel burned between WPT nodes
(e4) amount of fuel required to completely refuel each member of the RG .
(e5) start and finish times

Tankers: (a) Unique ID (b) Off-load capacity,

(c) Tanker solution attributes (blank unless tanker used)

(c1) WPT nodes served and associated times (arrival, service, and departure),
(c2) fuel burned and fuel offloaded for each WPT served, and
(c3) tanker start and finish times for deployment

Nodes (Tanker, WPT, RTB[2]):

(a) a unique ID (implies node type), (b) spatial location and fuel requirement, (c) RG assigned, (d) precedence relations with other nodes, (e) whether linked to another node for escort duty, (f) tanker assigned, (g) the fuel demand

Since the node objects are used to represent disparate entities, some additional remarks are appropriate. A tanker node uses only field (a). Creating a feasible solution (where no tanker runs out of fuel), may require some of the tankers to return one or more times to a base capable of tanker maintenance and refueling. At such a base, completion of these operations allows the tanker to continue servicing WPTs. To allow for this, tanker Return to Base (RTB) nodes are created for each active tanker base (as needed). RTB nodes are distinguished by zero fuel demand (from a tanker) and the lack of an assigned RG. Created only if needed, an RTB node will possess a unique ID (field (a)), its spatial location (field (b)) will correspond to the

---

[2] Return to Base (RTB).

associated refueling base, and the tanker ID of the associated tanker will occupy field (f). (All other fields are blank for an RTB node.)

A WPT node uses all seven fields. Field (d) will contain the information to assure that WPT nodes are visited in precedence (temporal) order along the RG's flight path. If a consecutive pair of WPT nodes within a RG's flight path are both over water and the RG has escort requirements, field (e) will indicate that the flight "leg" between the WPT pair requires escort by a tanker. Field (f) remains blank until a tanker is assigned to the WPT node as part of a generated solution to the AFRP. Field (g) will contain the RG's fuel demand at that WPT.

In solving a particular AFRP, the tanker nodes are generated first and are sequentially assigned indices starting at index 0. The WPT nodes are then created in the ascending order by RG ID and then by ascending order of associated WPT ID's within the RG flight path. For each WPT, its WPT nodes are indexed sequentially starting with the next available index. For every WPT along each RG 's flight path , either 1, 2, or 3 WPT nodes are generated.

Only one WPT node is required if both of the following conditions hold true:
(a) the RG's fuel demand at that WPT can be satisfied by a single tanker and (b) the RG does not require an escort to its next WPT.

Two WPT nodes are required if either of the following two sets of conditions are true:

(a) the RG's fuel demand at that WPT requires multiple tankers and the RG does not require an escort to its next WPT
(b) the RG's fuel demand at that WPT can be satisfied by a single tanker and the RG requires an escort to its next WPT.

If condition (a) holds, the WPT is represented by two WPT nodes where each is assigned one-half of the original WPT's fuel demand. With this representation, a different tanker can be assigned to each WPT node, jointly satisfying the total WPT demand.[3]

If condition (b) holds, the WPT is represented by two WPT nodes. The first node is assigned all of the WPT demand and the other is assigned a demand of zero. These WPT node creations and assignments are made so that one tanker can perform the refueling

---

[3] In this paper, two tankers could feasibly meet all of the demand at any WPT. This had the benefit of limiting the cardinality of the solution space by restricting the number of tankers that had to visit a WPT to at most two (without escorts). This assumption is consistent with the deployment scenarios considered herein.

function at the WPT and, if appropriate, another tanker can perform the escort duty to the next WPT. (The escorting tanker will provide the refueling function at the next WPT in the RG 's flight path.)

Three WPT nodes are required if the RG's fuel demand at that WPT requires multiple tankers and the RG requires escort to the next WPT. In this case, two of the three WPT nodes serve to allow the required refueling by two different tankers and it is possible that another tanker will assume the escort duty to the next WPT by being assigned to the third WPT node. When the WPT nodes are created, the fuel burned between a RG's adjacent WPTs is calculated and becomes the fuel demand for the latter WPT.

RTB nodes are generated as needed and are assigned the next available index.

Once the AFRP node objects have been created, they are used in the GTTS implementation using the OpenTS framework of Harder (2000). As explained in detail in Wiley (2001), Harder's framework requires the creation of the following objects: solutions, moves, neighborhoods, objective function, tabu list, and aspiration criteria. In this paper, we assume that the reader is familiar with classical tabu search techniques as described in Glover and Laguna (1997). Harder's framework provides the mechanism for implementing in the TS process as illustrated in Figure 5. A description of OpenTS is provided at http://oss.software.ibm.com/developerworks/opensource/coin/OpenTS/.

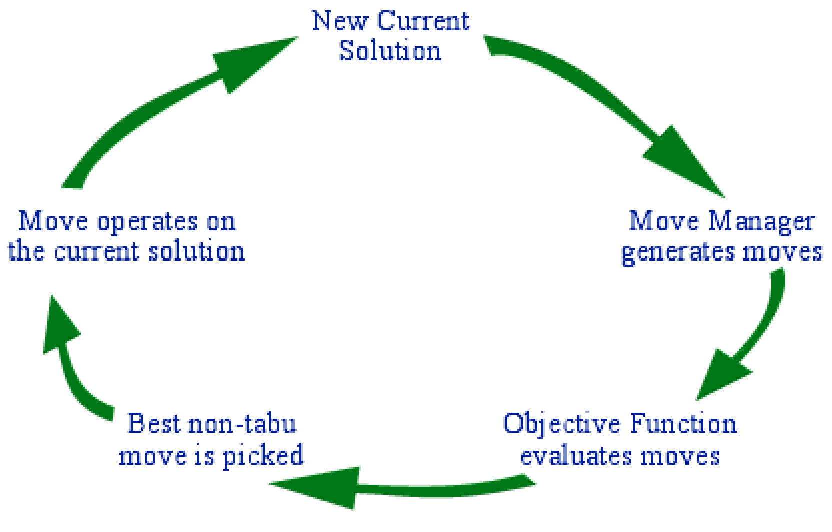

Figure 5. An iteration under the OpenTS architecture

A solution in our AFRP GTTS approach is represented by a permutation, $s \in S_n$, where n is the cardinality of the set of Node objects defined for the current instance of the AFRP being solved. Each letter in a solution, s, corresponds to the index of a AFRP Node object and each solution is stored as a symmetric group object as defined by Wiley's Symmetric Group Class Library (2001). An in depth description of the Symmetric Group class and the accompanying abstract Group class is presented in Wiley (2001). A java archive file, "jar," with the executable code and the associated Java$^{\text{TM}}$ documentation is available at

http://www.me.utexas.edu/~orie/techrep.html.

Each solution, s, embodies methods for creating moves, for performing conjugation and multiplication of elements of S_n and the current tanker assignments are implied in the solution's cyclic structure and the node indexing scheme described above. For example, the cycle, (0,15,16,17), would be interpreted as "Tanker 0 flies to (services) AFRP nodes 15, 16, and 17 and then returns home". All other tanker assignments are also represented with additional cycles. Indeed, all cycles in a solution will contain a single tanker node, which, by convention, will be placed in the first position of each cycle in the permutation representation.

Since S_n acts upon itself, all AFRP moves are also represented as permutations in S_n and move permutations act directly on solution permutations to produce different AFRP solutions. At each iteration, a defined neighborhood of eligible moves is evaluated. The neighborhoods generated by the AFRP are move-based neighborhoods rather than the more traditional solution-based neighborhoods. A solution-based neighborhood considers solutions reachable by one move. Rather than storing solutions, the moves that transform the current solution to a neighboring solution may be stored. Since function composition in S_n is a bijection of the set $\{1,2,\ldots,n\}$ onto itself, the move neighborhoods for the AFRP are composed of permutations in S_n and the move-based neighborhood is a natural extension of this concept.

After a candidate list (Glover and Laguna 1997) of moves is stipulated, the solution resulting from each move is evaluated relative the hierarchical objective function. This evaluation automatically performs any operations required to deconflict the schedules of the tankers and RGs in the new solution. Any solution that is the best solution found so far (within the current iteration), causes a check on the tabu status of the associated move. If the move is not tabu, that move is stored as the iteration's current best move found. Once the best available move is selected, that move is implemented and recorded within the tabu structure and the TS procedure begins a new iteration.

When an aspiration criterion is used, a move satisfying the associated criterion will be accepted regardless of its tabu status. An aspiration criterion object allows the definition of a criterion for any particular solution/move combination. The most common aspiration criterion used states that when a new solution is found that is superior to any found earlier in the search, the new solution is accepted regardless of its tabu status.

6 Solution Methodology

We now discuss some specific procedures and concepts unique to the GTTS and the AFRP: (a) construction of the initial solution, (b) dynamic AFRP move neighborhoods, (c) AFRP move evaluation criteria, (d) tabu memory structures used, and (e) consistent WPT generation.

6.1 Construction of the Initial Solution

For the current discussion, we assume that the WPT locations for all RGs have been supplied by an external source. This case prevails under previous AFRP solution techniques. (We present an adaptive tabu search approach for the WPT location problem later in the discussion.)

The initial solution created for the AFRP assigns all WPT nodes to the first tanker (tanker 0). Usually this will produce a highly infeasible solution (i.e., the tanker’s capacity will be exceeded). If a problem had 15 tankers and 32 WPT nodes, the initial solution would be (0,15,16,17,…,44,45,46). To overcome this infeasibility, the *Tanker Insert* (TKI) move neighborhood is used to insert unused tankers into the current employed tanker's WPT node assignments. In addition to reducing any infeasibility, the insertion point is strongly influenced by the requirement that some RGs must be escorted over open water. For the generation of TKI moves, the tankers are placed in a “pool” for each beddown base. If a beddown base has unassigned tankers in its pool, then one of the unassigned tankers is selected. For each selected tanker, moves that insert the selected tanker into the current solution are generated. Available insertion points start before the second waypoint node assigned to any tanker and continue up to before that tankers last waypoint node. An example of a TKI move, given the above initial solution, is “Insert tanker 5 in front of node 25”. Using permutation post-multiplication, this move is represented by (0,5,25).

Performing the multiplication, (0,15,16,…,24,25,26,…,46)*(0,5,25), yields (0,15,16,…,24) (5,25,26,…,46). Now tanker 0 is assigned nodes 15 through 24 and tanker 5 is assigned nodes 25 through 46

Placement of the remaining tankers continues until there are no available tankers or until a feasible solution is obtained. For example, if there were 14 required escort arcs between the 32 WPTs, the 15 tankers might be assigned to the WPTs yielding the following initial solution:

(0,15)(1,16,17)(2,18,19)(3,20,21)(4,22,23,24)(5,25,26)(6,27,28)(7,43,44)(8,29,30)(9,45,46)
(10,31,32,33,34) (11,35,36)(12,37,38)(13,39,40,41,42)

where 14 tankers are used before TKI moves are no longer used.

For every iteration where the TKI move neighborhood is used, a TKI move will be generated for each WPT node and tanker base with available tankers combination for a total of |WPT nodes| * |tanker bases| moves.

The choice of the above approach was motivated by two benefits: (1) the approach is easy to implement and (2) the associated move evaluation process drives the initial solution towards excellent solutions while feasibility is simultaneously being achieved.

6.2 Dynamic AFRP Move Neighborhoods

Once the initial TKI moves have performed their function, additional move types are invoked based on the current search status and solution. The move types discussed below were determined by both historical successes with the VRP (Carlton 1995) and by the need to develop new move types that would exploit the specific structure of the AFRP. These move neighborhoods include: (1) Return to Base Inserts (RTBI), (2) Restricted Inserts (RI), (3) Escort Pair Inserts (EPI), (4) Return to Base Deletes (RTBD), (5) Tanker Swaps (TKS), (6) Restricted Swaps (RS), and (7) Return to Base Swaps (RTBS).

Each move neighborhood creates a set of eligible moves to be evaluated and all use permutation post-multiplication. Post-multiplication allows the search to investigate AFRP solutions from different conjugacy classes in S_n, where a conjugacy class represents the number of tankers and their assigned WPTs within the current AFRP solution. In the following sections, each move neighborhood cited above is described with the conditions that cause them to be invoked.

Return To Base Insert Moves (RTBI)

After the TKI strategy is complete, the solution may still be infeasible, primarily due to a lack of tanker capacity. After the initial solution is achieved, the RTBI neighborhood is invoked whenever the current solution is infeasible. An RTBI move can be used to reduce infeasible fuel shortage at the cost of delaying one or more RGs, i.e., the tanker is unavailable until it has been refueled and allowed to reenter service.

The RTBI neighborhood is implemented by making available an RTB node for each active tanker base in the deployment scenario. These RTB nodes may be inserted within each of the current tankers' assignments. Available insertion points start at the first waypoint node assigned to a tanker within the solution and continue until after the last waypoint node. Suppose an RTB node is placed in tanker 4's assignment before node 24 for the solution

(0,15)(1,16,17)(2,18,19)(3,20,21)(4,22,23,24)(5,25,26)(6,27,28)(7,43,44)(8,29,30)
(9,45,46)(10,31,32,33,34)(11,35,36)(12,37,38)(13,39,40,41,42)

The move that will accomplish this is (24,49). The only affected cycle in this solution contains tanker 4, so only changes in that cycle need to be shown. (This convention of showing only the affected cycles will be followed throughout the rest of this paper.) Performing this move yields (4,22,23,24)*(24,49)=(4,22,23,49,24). Tanker 4 is now allowed to return to base, after servicing nodes 22 and 23, for refueling and ancillary operations before completing its assignment by servicing node 24.

For every iteration where the RTBI move neighborhood is used, a RTBI move will generated for each node (WPTs and tankers) part of the current solution and tanker base combination for a total of |WPT nodes + active tanker nodes| * |tanker bases| moves.

Restricted Insert Moves (RI)

Following initial solution construction, the RI neighborhood is invoked at each iteration of the search and allows an individual node to be inserted in a different position in its cycle or to be inserted in another tanker's cycle. An RI move can either reorder a tanker's assignment or change the partitioning of the letters among the tankers. The “restriction” on this type of move limits the allowable “distance” that a letter can be moved within the current permutation solution representation. “Distance” is defined as the number of positions a letter may move from its current position either to the left or right. For the results presented in this paper, the distance was set at 5 positions. This parameter setting and similar parameter settings discussed below were found through empirical experimentation for the example problems studied.

Consider the following two RI example moves. First we insert a node within its current cycle, i.e., “Insert node 24 in front of node 22 within tanker 4's current assignment.” The move and changes in tanker 4's assignment are achieved by (4,22,23,24)*(4,22,24)=(4,24,22,23).

Next we insert a node into another tanker's cycle., i.e., “Insert node 24 from tanker 4's assignment in front of node 25 in tanker 5's assignment”. The move and changes in the tankers’ assignments are: (4,22,23,24)(5,25,26)*(4,25,24)=(4,22,23)(5,24,25,26).

For every iteration where the RI move neighborhood is used, a RI move will be generated for every node (WPTs and RTBs) and position combination within 5 places, left or right, of the node. This will produce a total of |WPT nodes + RTB nodes| * (Distance * 2) moves.

Escort Pair Insert Moves (EPI)

After the initial solution construction, the EPI neighborhood is invoked for each iteration of the search. The EPI neighborhood inserts the two nodes associated with an escort arc. (An escort arc connects the last node of an earlier WPT to the first node of the adjacent WPT later in the path of a RG requiring escort. In any feasible solution, this pair of nodes must be adjacent and in the correct temporal order.)

Consider the following two clarifying examples: First, we “Insert the pair (31,32) after node 34 in tanker 10's assignment.” The move and changes in tanker 10's assignment are (10,31,32,33,34)*(10,31,33)=(10,33,34,31,32).

Second, we “Insert the pair (31,32) after node 46 in tanker 9's assignment.” The move and changes in the tankers' assignments are:

(9,45,46)(10,31,32,33,34)*(9,31,33)=(9,45,46,31,32)(10,33,34)

This reassigns the escort arc duty from tanker 10 to tanker 9.

For every iteration where the EPI move neighborhood is used, an EPI will be generated for each Escort Pair and position combination within 5 places, left or right, of the node. This will produce a total of |Escort Pairs| * (Distance * 2) moves.

Both the RI and EPI neighborhoods can be viewed as variants of the traditional *k*-OrOpt move.

Return To Base Delete Moves (RTBD)

RTBI moves help the search progress towards feasibility while increasing the number of nodes being used in the solution. Once the number of letters reaches 1.5 times the original number of nodes, the RTBD move neighborhood is invoked. This neighborhood is used to remove any extra, i.e. nonbeneficial, RTB nodes from the solution.

An illustrative example of an RTBD move is “Remove the return to base node 49 from tanker 4's assignment.”

The move and change in tanker 4's assignment is: (4,22,23,49,24)*(24,49)=(4,22,23,24).

For every iteration where the RTBD move neighborhood is used, a RTBD move will be generated for each RTB node in the current solution for a total of |active RTB nodes| moves.

Tanker Swap Moves (TKS)

In addition to the RTBD neighborhood, the TKS neighborhood is invoked when the number of nodes in the solution has grown to 1.5 times the original number. This neighborhood allows idle tankers to be exchanged with active tankers (from different bases) to reduce travel and fuel usage. For each beddown base that has unassigned tankers in its tanker pool, a move that exchanges an assigned tanker from a different beddown base within the current solution is generated.

An example of a TKS move is “swap tanker 14 for tanker 10.” The move and changes in the solution are (10,31,32,33,34)*(10,14,31)=(14,31,32,33,34).

Additionally, the TKS neighborhood allows an idle tanker to be exchanged with an active tanker that has been relocated to the idle tanker's beddown base. This may occur in two ways: (1) a tanker may relocate, provide service and then return to an active tanker base (which differs from the relocation base and beddown bases) and (2) the tanker relocates, provides service, and then returns to the relocation base.

An example of the first case is “Swap idle tanker 10 for active tanker 7.” Tanker 7 has relocated to tanker 10's beddown base prior to servicing WPTs. After servicing nodes 36 and 37, tanker 7 relocates to another active tanker base. If the original assignment for tanker 7 is (7,67,35,36,65) (idle tanker 10’s assignment is (10)), the move and change is (7,67,35,36,65)*(7,10,35,67)=(10,35,36,65). Note that RTB node 67 also has been removed from activity.

An example of the second case is “Swap idle tanker 5 for active tanker 13.” Tanker 13 has relocated to tanker 5’s beddown base, provided service, and then returned to tanker 5’s beddown base. Suppose the original assignment for tanker 13 is (13,51,41,42,57). The move and changes are (13,51,41,42,57)*(5,41,51,13,57)=(5,41,42) and two RTB nodes, 51 and 57, have been removed.

For every iteration where the TKS move neighborhood is used, a TKS move will be generated for each tanker in the current solution and tanker base combination for a total of |active tankers| * |tanker bases| moves. Additionally, a TKS move will be generated for each active relocated tanker and tanker base combination for a total of |relocated active tankers| * |tanker bases| moves. In total, |active tankers + relocated active tankers| * |tanker bases| moves are generated.

Restricted Swap Moves (RS)

During the search, if a specified number of iterations have passed (20 iterations for the results of this paper) without a new best solution being identified, the RS neighborhood is invoked. RS moves allow an individual node within a tanker's assignment to be swapped either with a node in its cycle or with a node in another tanker's cycle. This maintains the current cardinality of the partitioning of the letters amongst the tankers. The "restriction" of this neighborhood limits the allowable distance (5 positions for this paper) that any letter can be moved.

An example of an RS move where we swap a letter with another in its current cycle is "Swap node 31 for node 34 in tanker 10's assignment." The move and changes of tanker 10's assignment is (10,31,32,33,34)*(10,32)(31,34)=(10,34,32,33,31).

An example of swapping one tanker's node with another tanker's node is "Swap node 34 from tanker 10's assignment with node 24 of tanker 4's assignment." This yields (4,22,23,24)(10,31,32,33,34)*(4,10)(24,34)= (4,22,23,34)(10,31,32,33,24).

For every iteration where the RS move neighborhood is used, a RS move will be generated for each node (WPTs and RTBs) and position combination within 5 places, left or right, of the node. This will produce a total of |WPT nodes + RTB nodes| * (Distance * 2) moves.

Return To Base Swap Moves (RTBS)

Like the RS neighborhood, the RTBS neighborhood is invoked when a specified number of iterations (20 for this paper) have passed without a new best solution being identified. This neighborhood allows the RTB nodes to be exchanged with other RTB nodes. This allows the solution to adjust the locations of the return to base nodes to fit the current set of tanker assignments. For each beddown base, a RTB node is available. For each selected RTB node, a move that exchanges an RTB node from a different beddown base within the current solution is generated.

An example of an RTBS move is “Swap node 48 for node 56 in tanker 13's assignment.” If tanker 13's assignment was (13,34,48), the move and change in tanker 13's assignment is (13,34,48)*(13,48,56)=(13,34,56).

For every iteration where the RTBS move neighborhood is used, a RTBS move will be generated for each RTB node and tanker base combination for a total of |active RTB nodes| * |tanker bases|.

6.3 Move Evaluations

AFRP move evaluations determine, first and most importantly, whether a move yields a feasible solution and, second, whether a move yields a superior solution. We discuss below, first, a number of feasibility conditions that effect the hierarchical move evaluations for the AFRP. Any violations of these conditions are noted and counted, one at a time, during the hierarchical evaluation of an AFRP solution. Item (d) does not appear in the hierarchical list presented in section 2. Its presence here is due strictly to algorithmic concerns in the implementation of the GTTS.

(a) *Unescorted Arcs* - If a RG requires escort, all escort legs of its flight path must be accompanied by a tanker. Each escort arc must be represented as adjacent nodes within a tanker assignment. The number of unescorted arcs is returned for the purpose of objective function evaluation.

(b) *Uncovered RG Demand* - For every RG, a set of WPTs with fuel demands are generated. Within the current tankers' assignments, feasibility requires that the entire set of WPT fuel demands be covered. During the tabu search process it is possible to have solutions that leave some WPT fuel demands uncovered. The number of uncovered WPT fuel demands is returned.

(c) *Misordered Tanker* Services - For every tanker assignment, the order of service for the assigned WPT fuel demands is extremely important. A tanker cannot first provide service to a RG at a WPT that is farther along the RG's flight path and later service a WPT earlier along the same flight path. The number of misordered WPT fuel demands is returned.

(d) *Bad Tanker Assignments* - As the tabu search progresses, there are a number of items that logically should not occur. These items include: (a) "Tanker base hopping" (adjacent RTB nodes), (b) Two tankers within a single cycle in the AFRP solution representation, and (c) a tanker providing repeat servicing to the same RG.

Tanker base hopping occurs during the search when return to base nodes become adjacent to one another within a tanker's assignment. This has the effect of repositioning the tanker from one location to another without providing any useful service. The number of adjacent return to base nodes is recorded.

During the search, a tanker may be assigned within another tanker's route. For the purposes of this model, a tanker must either be providing service or receiving service as part of a RG, but not both. The number of tankers assigned within another tanker's route is recorded.

A tanker providing repeat servicing captures the condition that a tanker could be assigned to provide service to a specific RG, return to an active tanker base, and then provide service to the same RG. This pattern is considered undesirable since a tanker, when it returns to an active tanker base, must remain at that base for a minimum amount of service time (4 hours for this research). During this service, the RG is continuing along its flight path and, more than likely, cannot be feasibly serviced again by the same tanker. The number of tanker repeat servicings is recorded.

The sum of the three items is returned.

(e) *Infeasible Fuel Usage* – If any tanker or any RG attempts to use more fuel than is available to it in an AFRP solution, the total amount of such infeasible "phantom" fuel used is returned.

In addition to feasibility, criteria (f) through (m) of the AFRP objective function effect the move evaluation.

(f) *"Orbit" times* - During the search, scheduled times for a tanker's arrival to an assigned WPT may not coincide with the RG's arrival at that WPT. When this occurs, either the tanker must "orbit" (fly in circles) until the RG arrives or vice versa. In either case, the total amount of time spent in "orbit" by all tankers and RGs is returned.

(g) *RG late arrival* - It is not always possible for all RGs to reach their final destinations by their desired latest times. The sum of all RG late arrival times is returned.

(h) *Tankers used* - The actual number of tankers assigned to one or more RG WPTs is returned.

(i) *Tanker mission time* – The total of all tanker usage times, end of mission time minus start of mission time, is returned.

(j) *Tanker distance flown* -. The sum of all tanker distances flown in the solution is returned.

(k) *Tanker fuel usage* - The sum of all the fuel consumed by tankers in the solution is returned.

(l) *Tanker fuel delivered* - The sum of all fuel delivered to RGs in the solution is returned.

(m) *RG fuel usage* - The sum of all the fuel used by RGs in the current solution is returned.

6.4 Tabu Memory Structure

Part of the power of tabu search is achieved by using the solution and/or move history. The memory structure used in GTTS stores attributes of the moves that have been previously selected and changes the length of the "short term memory" (tabu tenure) using an adaptive procedure. Move attributes are stored in a square matrix of dimension *n*, the number of solution nodes. As moves are implemented, the letters of the move are recorded in the matrix. For each letter of the move, the row index is the letter, the column index is the image of the letter (the "next" letter in the cyclic context of the move), and the value placed in the matrix is the current iteration plus the current tabu tenure. Suppose *n*=3. At iteration zero, the tabu matrix would be as (a) in Figure 6. A tabu tenure of 7 and a move of (1,2,3) yields, at iteration 1, the tabu matrix (b) in Figure 6.

$$\underset{(a)}{\begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}} \qquad \underset{(b)}{\begin{bmatrix} 0 & 7 & 0 \\ 0 & 0 & 7 \\ 7 & 0 & 0 \end{bmatrix}}$$

Figure 6. Tabu Memory Structure

For future iterations, any move that contains one or more of the letter-to-image-of-letter (row to column) combinations with nonzero entries in the tabu matrix will be considered tabu.

Whenever the value of n changes, the tabu matrix is automatically redimensioned and the current tabu structure is preserved.

As the search progresses, the tabu tenure is adaptively modified based on the status of the current solution (Chambers 1996, Dell'Amico and Trubian 1993). If the current solution is the best solution found so far, the tabu tenure is reset to a specified default value (7 for this paper). If the current solution is a better move than the last, but not the best solution so far, the tabu tenure remains at its current value. If the current solution is not better than the previous move, the tabu tenure is increased by one.

6.5 An iteration in the GTTS

A GTTS iteration begins relative to the current solution. Based on the characteristics of this solution and the current status of the tabu search, move-based neighborhoods are generated. At any point in the search, one or more of the neighborhoods described in section 6.2 is invoked. Their union forms the composite search neighborhood for the current iteration.

Once the generation of moves is completed, the evaluation of these moves begins. A move's value is the difference between the current solution and the solution created by the move. Initially, the first move evaluated is considered the best move found. All other moves are compared against the current best move found by using the hierarchical objective function described earlier.

Items (a) through (d) listed in section 6.3 require only the specific partitioning and ordering of the waypoint nodes to the tankers as contained in the S_n representation. Each of these may be evaluated without the determination of the specific timing of events.

The remaining items in section 6.3 require the computation of specific timing of events for evaluation. Once the timing of events is determined, the evaluations of each of the hierarchical objectives is straightforward; therefore, the remainder of this section concentrates on the determination of the timing of events for a particular move being applied to the current solution.

The determination of the timing of events occurs through the application of the following three steps:

1. Determine each RG's flight path times regardless of tanker assignments
2. Determine each tanker's route times regardless of assigned RG times
3. Deconflict schedules determined in steps 1 and 2

The first step takes each individual RG and, based on its EDT with no delays, determines the arrival, service, and departure times at each waypoint node along its flight path as well as its destination arrival time. Similarly, the second step takes each individual tanker and, with no delays, determines the arrival, service, and departure times at each of its assigned waypoint nodes.

The third and final step in determining the timing of events for a solution deconflicts the schedules created in the first two steps through an iterative approach. This iterative approach involves two primary stages. The first stage considers each RG and determines the largest difference between its arrival at any waypoint node and the arrival of the tanker assigned to that waypoint node. For each RG whose largest difference is positive, the starting time of the RG is adjusted by the amount of this largest difference. The second stage considers each tanker and its associated route assignment. The difference between the tanker arrival at its first waypoint node and the associated RG arrival is determined. The tanker's schedule is adjusted to ensure that it either arrives at the same time as the RG or “orbits” until the RG arrives. Once the first waypoint node in the tanker's assignment is deconflicted, the procedure continues with the next waypoint node in its assignment. This iterative approach continues until all nodes within the tanker's assignment have been deconflicted. When all tankers have been deconflicted, step 3 repeats. This iterative process continues until no conflicts are found or until 10 iterations of step 3 have been completed. If 10 iterations are reached, the current candidate move is removed from consideration.

The choice of placing all “orbit” time in the tanker schedules is based on the capacity of the airframes that comprise the RGs, particularly fighters. For smaller aircraft, their fuel capacity minimizes their allowable orbit times. With the timing of events determined, the remaining objectives are evaluated. Once the best move is determined by the objective function, it is placed within the tabu structure. After this placement, the best move selected is implemented on the current solution, creating the next solution to be used in the tabu search iterative process.

6.6 Consistent WPT Generation

The above description of the GTTS assumed that the RG WPTs were provided by an external source and were consistent, i.e., feasible (flyable) solutions could be found when those WPTs were used. Of course, externally supplied WPTs are not necessarily consistent. To account for this possibility, a modified form of the GTTS, the GTTS Preprocessor (GTTSP), has been developed to determine consistent WPTs for a single RG's flight path. GTTSP is also an adaptive tabu search method developed specifically to find consistent active WPT node sets for a single RG. Reasons for using the GTTSP include: (a) no externally supplied WPTs are available, (b) the provided WPTs are untested, and (c) a comparison of the supplied WPTs against the GTTSP WPTs is desired

GTTSP requires the use of a set of candidate WPTs (nominally 100 nautical miles (NM) apart) throughout the selected RG's flight path. From this set of candidate WPTs, a subset is selected as “active”. This active subset of WPTs defines the actual WPTs for the RG's flight path in the AFRP to be solved by the GTTS.

The active subset of WPTs is selected based on objectives identical to those for the GTTS and imposes the same tanker resource configuration that is used by the GTTS on the AFRP. The application of the GTTSP differs depending on whether a RG requires escort.

For the no escort case, each candidate WPT is associated with a single candidate WPT node. Once the set of WPT nodes are available, an initial WPT node selection set for the individual RG is constructed by first selecting the “midway” WPT node along the RG flight path. If the number of candidate WPTs is odd, the midway WPT is uniquely determined. If the number of candidate WPTs is even, the midway WPT is taken to be the earliest WPT of the two midway WPTs. The amount of fuel required at this active WPT node is computed to be equal to the RG fuel used to reach the midway WPT node's location. Depending on the consistency of this initial WPT node selection, one of two moves is employed.

A Preprocessor TKI (PTKI) move neighborhood is used if the WPT node set is inconsistent. Three conditions can cause such an inconsistency: (a) no tanker can fly to the midway point, provide service to the RG, and then return to its original beddown base without running out of fuel, (b) the RG can not fly to the midway WPT node without running out of fuel or (c) the RG can not fly from the midway WPT node to its destination base without running out of fuel.

If any of these conditions exist, then a PTKI move adds a WPT node to the current active set, assigns a tanker to that node, and checks for consistency. If the active WPT node set is still inconsistent, the PTKI move neighborhood is employed again. In this manner, the PTKI neighborhood determines the cardinality of the active node set. Once a feasible solution is found, the Preprocessor WPT Node Adjacent Swap (PAS) neighborhood is employed.

PAS moves serve to iteratively improve upon the consistent “active” node set. It achieves this by taking each “active” node and generating a move that replaces it with either its predecessor or successor along the flight path. For example, if the consistent solution is (0,50)(6,75), then (0,49)(6,75), (0,51)(6,75), (0,50)(6,74), and (0,50)(6,76) are reachable in a single PAS move. The best active WPT node set found by iteratively applying the PAS neighborhood within the GTTSP then serves to define the flight path WPTs for the RG in the AFRP.

Now suppose that escort is required for a RG. If a WPT is not located over open water, then a single candidate WPT node is generated. If a WPT node is located over open water and the adjacent WPT farther along the flight path is also over open water, then two candidate WPT nodes are generated. The *escort range* is a single contiguous interval starting at the first WPT escort node and ending at the last WPT escort node.

The initial solution is constructed as (tanker ID, first WPT node ID, last WPT node ID), i.e., the tanker flies to the first WPT node, provides service to the RG, escorts the RG to the last WPT node, provides service to the RG, and then returns to its beddown base. The fuel requirements at each of these “active” WPT nodes are determined by the GTTSP and, depending on consistency, either PTKI and/or PAS moves are employed until consistency is achieved. This process, which is very similar to the no escort required case, is discussed in detail in Wiley (2001).

7 Example Results

Wiley (2001) presents results for a small example with 3 RGs and 18 tankers, for a benchmark example with 9 RGs and 60 tankers and for a “typical middle east deployment” with 99 aircraft in 26 RGs serviced by 120 tankers.

The small example, depicting a fictitious deployment from the continental US to Saudi Arabia, was solved to enhance understanding of the complex interactions embodied in the solution to any AFRP and to clarify how such a solution may be presented to a prospective user of the GTTS solution. The details on the RGs are given in Table 1 for an assumed deployment start designated as time 0 (zero).

Table 1. Departure and Destinations Information

RG ID	AC Type	No. AC	Origin	Destination	EDT	RDT
0	F15	6	KLFI - Langley AFB, VA	OERY - Riyadh AB	8	40

1	F15	6	KLFI - Langley AFB, VA	OERY - Riyadh AB	0	40
2	F117	6	KHMN - Holloman AFB, NM	OEDR - King Abdul Aziz AB	0	40

The RGs may be serviced by any of 18 KC-135R tankers located at KBGR (6 tankers - Bangor IA, ME), EGUN (6 tankers - Mildenhall AB, UK), KGSB (3 tankers - Seymour-Johnson AFB, NC) and PAEI (3 tankers - Eielson AFB, AK). The scenario for this small deployment is pictured in Figure 7 where the tanker and RG departure bases are presented as pentagons and the RG flight paths are shown as WPTs and destination bases (circles) connected by lines. The results pictured in Figure 8 and Figure 9 are based on applying the GTTS on an AMD Athlon 950 Mhz machine. The best results were achieved at iteration 369 (in about 25 minutes) of a 30 minute run where 430 iterations were completed. Figure 8 shows how the number of tankers used (left vertical axis) and the hours of orbit delay time at WPTs (right vertical axis) changed during the iterations (horizontal axis) of the search procedure.

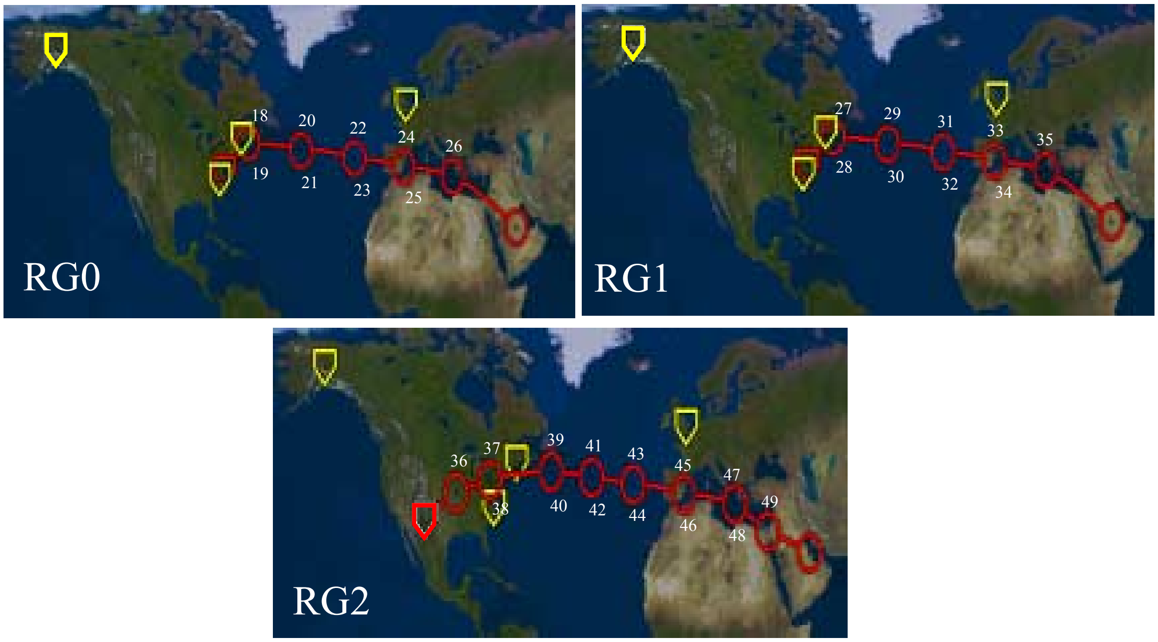

Figure 7. Scenario for Small Deployment

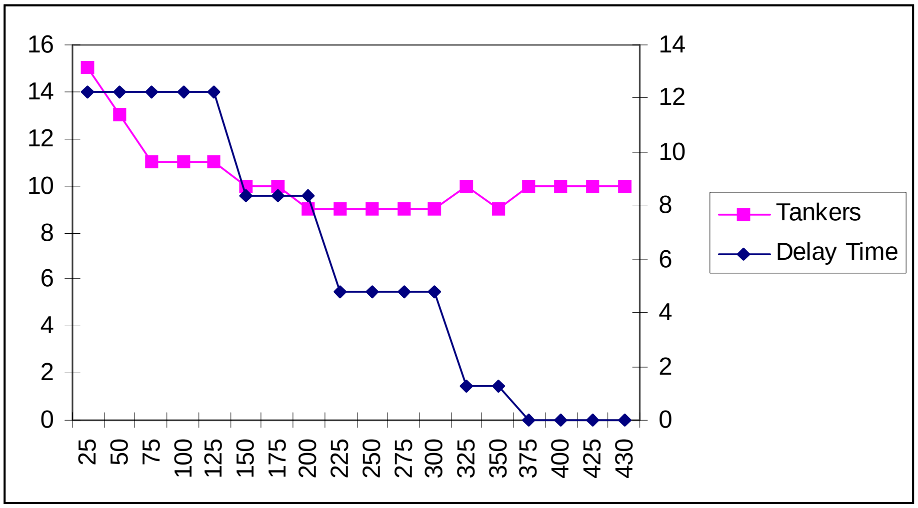

Figure 8. Solution Snapshot Plot

Figure 9 presents the best solution found from a "temporal assignment" viewpoint. As may be observed, 12 of the 18 tankers are employed in refueling and escort activities over the 40 hour time horizon of the deployment. In addition, each RG's and tanker's activities can be observed, in *detail*, from a careful examination of Figure 9. For example, tanker 6, based at EGUN, is idle until hour 4 when it flies to a WPT to provide service to RG1 between hours 7 and 10; it returns to EGUN, takes off at hour 15 and provides service to RG0 and then returns to EGUN for the remainder of the deployment. Similarly, RG2's activities can be determined. RG2 takes-off in time to meet tanker 2 for service, beginning at the Atlantic coast at hour 20. It is subsequently serviced by tankers 2, 3, 11, 7, 8, and 9. After service by tanker 9, RG2 completes its flight to OEDR.

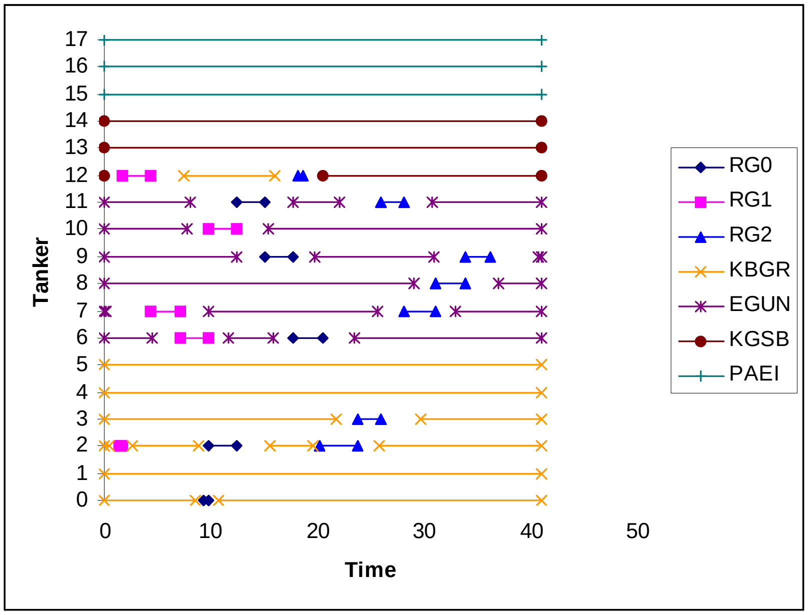

Figure 9. Tanker Assignments for a Small Deployment

The benchmark problem was studied to provide a comparison of the TAP Tool to the GTTS method. The TAP Tool was run 6 *times* for 90 minutes (with different static tabu tenures) on a Intel Pentium II 350 Mhz machine (a total of 9 *hours*). Using the same problem structure and assumptions as those embodied in the TAP Tool, the GTTS was run *once* for 30 minutes on an Athlon 950 Mhz machine. The best results from the 6 TAP Tool runs yields a composite solution using 24 tankers traveling 215,204 miles with the latest RG arrival requiring 69.1 hours. The best solution from the single GTTS run used 23 tankers, flying 106,227 miles, with the latest RG arriving at 69.2 hours. Thus the GTTS used one fewer tankers, less than half the tanker fuel consumption, with only one-tenth of an hour added to the latest RG arrival time.

The typical middle east deployment scenario was constructed with the aid of AMC to provide information on GTTS performance on an AFRP of practical size. The best solution, found in two hours and 16 minutes, employed 95 tankers flying 326,968 miles and allowed the latest RG to arrive in 55 hours. The tanker assignments in this solution were obtained at the *detailed asset operational level* (individual tankers or receivers) and yield excellent assignments in terms of such metrics as minimizing tanker usage, tanker travel, and total deployment time. In general, depending on the desire of the user, Wiley's AFRP method can provide either a single "best" solution to a particular problem or, as indicated in Figure 6, a set of robust, comparable solutions to a problem instance. The fact that the best previous method, CMARPS, requires several analysts weeks, and occasionally months, to achieve a single feasible solution strongly underscores the power that GTTS methods bring to the solution of USAF logistics problems. Thus, with this new tool, analysts and decision makers not only can have the flexibility and added insight provided by multiple solutions, they also can make critical decisions in a tremendously shortened planning horizon.

9 Conclusions and Directions for Future Work

This research has yielded three major contributions: (a) a reusable, portable code for implementing and applying group theory to P|O combinatorial problems, (b) a demonstration of the effectiveness of using dynamic search methodologies has been shown for the AFRP, and (3) the development of a very effective solution methodology for the AFRP.

Future work could be addressed at both improving the scope and efficiency of Wiley's (2001) Java^{TM} class library for S_n. In addition, the efficiency of the GTTS code could be improved both in the construction of the initial solution and in the algorithmic implementation of the search technique. The current GTTS uses an adaptive approach to adjusting the tabu tenure; the more powerful reactive approach of Battiti and Tecchiolli (1994) could possibly yield significant improvements in the solution methodology. An investigation of more sophisticated tabu memory structures also might be beneficial.

Further, the AFRP is just one of a myriad of problems being addressed by AMC. Other important AMC problems are associated with (a) refueling needs in intra-theater employment, (b) once tanker schedules have been determined, assigning aircrews to the tankers, (c) assigning the aircrews for RGs.

In addition to USAF logistics problems, the GTTS, or just the SymmetricGroup class, can be applied to any problem where the partitioning and ordering of elements is important. Colletti (1999) describes the use of the S_n for representing the m-TSP. As the GTTS has demonstrated, far more complicated problems can be solved using S_n as the basic representation of solutions and for constructing varied move neighborhoods.

These move neighborhoods can be compactly represented using the group actions of multiplication and conjugation. Any k-OrOpt move can be represented by three letters from S_n combined with right-multiplication. The first of the three letters represent the beginning of the k-length pattern, the second represents the letter immediately following the end of the k-length pattern, and the third letter represents the letter to insertion point of the *k*-length pattern. Any m-letter rearrangement move can be represented by *m*-letters from S_n combined with conjugation. An excellent discussion of these rearrangement moves can be found in Colletti (1999). In fact, S_n facilitates the construction of very complex move neighborhoods in a compact representation through the use of conjugation and multiplication as described and demonstrated in Wiley (2001), Combs (2001), and Crino (2002).

In conclusion, the computational results reported here indicate that the GTTS is efficient and effective, providing excellent results with no tuning. The procedures are sufficiently robust to embrace variants of the basic AFRP with modified objectives and constraints. Future researchers should be able to build on the foundation provided by this GTTS application to the AFRP to construct highly effective approaches to other combinatorial optimization problems.

References

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

Artin, M. (1991). *Algebra*, Prentice-Hall, Inc., Englewood Cliffs, New Jersey 07632.

Barnes, J. W., B. Colletti and D. Neuway, "Using Group Theory and Transition Matrices to Study a Class of Metaheuristic Neighborhoods,"*European Journal of Operational Research*, 138 (2002) 531-544.

Bartalos, I., Dudas, T. & Imreh, B. (1995). On a Tour Construction Heuristic for the Asymmetric TSP, *Acta Cybernetica* 12: 209-216.

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

Byrd, V. B. (1994). Passing Gas: The History of Inflight Refueling, Byrd Publishing Company, Chico, CA.

Capehart, S.R. (2000). A Tabu Search Metaheuristic for the Air Refueling Tanker Assignment Problem, Master's thesis, Air Force Institute of Technology, Wright-Patterson AFB, OH.

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

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

Chambers, J.B. (1996). *Classical and Flexible Job Shop Scheduling by Tabu Search*, PhD thesis, The University of Texas at Austin.

Colletti, B., Barnes, J. & Dokov, S. (1999). A Note on Characterizing the K-OPT Neighborhood Via Group Theory, *Journal of Heuristics* **5**: 51-55.

Colletti, B. & Barnes, J. W. (2001). Local Search Structure in the Symmetric Traveling Salesperson Problem under a General Class of Rearrangement Neighborhoods, *Applied Mathematical Letters* **14**(99-06): 105-108.

Colletti, B. W. (1999). *Group Theory and Metaheuristics*, PhD thesis, The University of Texas, Austin, TX.

Colletti, B. W. & Barnes, J. W. (2000). Linearity in the Traveling Salesman Problem, *Applied Mathematics Letters* **13**(3): 27-32.

Combs, T. (2001). A Group Theoretic Tabu Search Approach to Tanker Crew Scheduling, *Technical Report AFIT/DS/ENS/02*, Air Force Institute of Technology.

Crino, John R., A Group Theoretic Tabu Search Methodology for Solving the Theater Distribution Vehicle Routing and Scheduling Problem, PhD Dissertation, Air Force Institute of Technology, 2002.

Dell'Amico, M. & Trubian, M. (1993). Applying Tabu Search to the Job-Shop Scheduling Problem, *Annals of Operations Research* **41**: 231-252.

Gaglione, A. M. (1992). An Introduction to Group Theory, *Technical Report PU/NRL/5350-92-231*, Naval Research Laboratory, Washington, D.C.

Gendreau, M., Laporte, G. & Potvin, G. (1997). *Vehicle Routing: Modern Heuristics*, John Wiley & Sons, Chichester, chapter 9, pp. 311-336.

Gilmore, P. C., Lawler, E. L. & Shmoys, D. B. (1985). *Well-solved Special Cases*, John Wiley & Sons, New York, chapter The Traveling Salesman Problem: A Guided Tour of Combinatorial Optimization, pp. 87-143.

Glover, F. & Laguna, M. (1997). *Tabu Search*, Kluwer Academic Publishers, Boston, MA.

Gulf War Air Power Survey (1993). Vol. III, Logistics and Support, Washington, D. C.

Harder, R.W. (2000). *A Jump Search Enhanced Reactive Tabu Search in a Universal Vehicle Router Applied to Routing Unmanned Aerial Vehicles*, Master's thesis, Air Force Institute of Technology, Wright-Patterson AFB, OH.

Hostler, H. (1987). *Air Refueling Tanker Scheduling*, Master's thesis, Air Force Institute of Technology, Wright-Patterson AFB, OH.

Isaacs, I.M. (1994). *Algebra*, Brooks/Cole Publishing, Pacific Grove, CA.

Karp, R. M. (1979). A Patching Algorithm for the Nonsymmetric Traveling Salesman Problem, *SIAM Journal of Computing* **8**(4): 561-573.

Kuhn, H. (1955). The Hungarian Method for the Assignment Problem, *Naval Res. Logistics Quart.* **2**: 83-97.

Moncrief, K. W. (1996). Tactical Air Refueling: Undocumented Past, Future Requirement, *Technical Report Graduate Research Paper AFIT/GMO/LAC/96N-9*, Air Force Institute of Technology.

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

Smith, R. K. (1998). *75 Years of Inflight Refueling Highlights*, 1923-1998, U.S. Government Printing Office, Washington, D.C.

Sunderman, J.F. (1961). *Early Air Pioneers*, Franklin Watts, Inc., New York, NY.

Taillard, E., Badeau, P., Gendreau, M., Guertin, F. & Potvin, J. (1997). A Tabu Search Heuristic for the Vehicle Routing Problem with Soft Windows, *Transportation Science* **31**: 170-186.

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

Yamani, A. A. (1986). *Analysis of an Air Transportation System*, PhD thesis, University of Florida.