Solving the Theater Distribution Vehicle Routing and Scheduling Problem Using Group Theoretic Tabu Search

Major John R. Crino, Ph.D.
Department of Operational Sciences
Air Force Institute of Technology

James T. Moore, Lt Col, USAF (RET), Ph.D.
Associate Professor of Operations Research
Department of Operational Sciences
Air Force Institute of Technology

J. Wesley Barnes, Ph.D.
Cullen Trust for Higher Education Endowed Professor in Engineering
Graduate Program in Operations Research and Industrial Engineering
The University of Texas at Austin

Colonel William P. Nanry, Ph.D.
Director, Army Quadrennial Defense Review Office
Office of The Army G8, Pentagon

Abstract— The military "theater distribution vehicle routing and scheduling problem" (TDVRSP) is associated with determining superior allocations of required flows of personnel and materiel within a defined geographic area of operation. A theater distribution system is comprised of facilities, installations, methods, and procedures designed to receive, store, maintain, distribute, and control the flow of materiel between exogenous inflows to that system and distribution to end user activities and units within the theater. An automated system that can integrate multi-modal transportation assets to improve logistics support at all levels has been characterized as a major priority and immediate need for the US military services.

This paper describes both the conceptual context, based in a flexible group theoretic tabu search (GTTS) framework, and the software implementation of a robust, efficient, and effective prescriptive generalized theater distribution methodology. This methodology evaluates and prescribes the routing and scheduling of multi-modal theater transportation assets at the *individual asset operational level* to provide economically efficient time definite delivery of cargo to customers.

Keywords— vehicle routing, scheduling, theater distribution, tabu search, group theory, group theoretic tabu search, symmetric group

1 Introduction

In the last decade, serious shortcomings during military operations such as *Desert Shield/Desert Storm* and *Allied Force* dramatically illustrated the need for logistical process improvement (Kaminski, 1995 and Brooks, 2000). As a result, the creation of a methodology that integrates multi-modal transportation assets to more efficiently and effectively plan and execute logistics support within a theater has become a major priority and immediate need among the military services (Brooks, 2000).

Achieving superior logistical efficiency mandates that the military discard the outmoded “just-in-case inventory” approach (with its extremely costly long-term stockpiles of materiel (Schrady, 1990)), and move to a rapid and reliable transportation process that provides time definite delivery to customers. Responding to this requirement, the United States Air Force (USAF) initiated the development of an automated Theater Distribution Management System (TDMS) which enables logistics planners to react to rapidly developing contingency operations common to the expeditionary USAF (Brooks, 2000). The USAF also plans to develop a scalable model that meets the other services’ theater distribution modeling requirements.

Theater distribution is the flow of personnel and materiel within theater to meet the geographic combatant commander’s intent (JP 4-01.4, 2000). (A theater is a geographical location outside the continental United States for which a commander of a unified command is assigned military responsibility (JP 1-02, 1996).)

A theater distribution system is comprised of facilities, installations, methods, and procedures designed to receive, store, maintain, distribute, and control the flow of materiel between exogenous inflows to that system and distribution to end user activities and units within the theater (JP 1-02, 1993). Such a distribution system may be efficiently represented by a *network* where the associated physical components are categorized as nodes, modes and routes.

This paper describes a deterministic methodology that, for the *first time*, provides vehicle routing and scheduling of multi-modal theater transportation assets at the *individual asset operational level* to provide economically efficient time definite delivery of cargo to customers for generalized theater distribution problems. The model has been shown to be robust, flexible, and capable of solving a hierarchy of TDVRSP problems (theater distribution vehicle routing and scheduling problems) (Crino, 2002). Group theoretic tabu search (GTTS) was the underlying framework utilized in the development of this methodology and several significant new abstract algebraic concepts were brought to light during the associated research.

2 Literature Review

All current theater distribution models used by the military are simulation models. Among such models are the Enhanced Logistics Intratheater Support Tool (ELIST), Transportation Resource Assessment Network Simulator (TRANS), and the Mobility Analysis Support System (MASS). While these models assist in the understanding of distribution networks, none fulfill the defined requirements for theater distribution planning (Synergy, 2002). These requirements include prescribing routing and scheduling plans that optimize the utilization of theater assets while satisfying customer demand and time requirements.

One way to meet the stated requirements is to model the theater distribution problem as a vehicle routing and scheduling problem (VRSP) and use deterministic methods to solve the problem. VRSP's have been attacked using both optimization methods and heuristic methods. Classical optimization methods that find and verify an optimal solution require unacceptably large computation effort and time for practical sized problems. In addition, data describing a practical TDVRSP instance will be imprecise. Pristine optimal solutions based on such inputs will typically be inapplicable to the practical scenario. While modern metaheuristic search methods do not guarantee an optimal solution, they have been shown to find excellent robust solutions for real-world problems with dramatically less time and effort, usually within a real time planning horizon. Such robust solutions, appropriate to a wide spectrum of TDVRSP instances, are highly preferred by theater distribution movement planners. Additionally, a theater distribution plan is often not a static document. Unforeseen events can entail modifications to the initial plan requiring planners to respond in a short time horizon.

A well-established heuristic method for solving VRSPs is tabu search (TS) (Glover, 1997). A recent survey conclusively showed that TS outperforms other heuristics in finding solutions to established VRSP benchmark problems (Laporte and others, 2000). TS has successfully solved other military applications addressing a number of vehicle routing, scheduling, and assignment problems (Ryer, 1999; O'Rourke, 1999; Cullenbine, 2000; Hall, 2000; Calhoun et al., 2002; Kinney, 2000; Wiley, 2001; Brown, 2001; Crino, 2002; and Combs, 2002).

In a groundbreaking dissertation, Colletti pioneered the association of group theory with metaheuristics when he identified the symmetric group on n-letters, S_n, as a natural setting for solving permutation and ordering problems with metaheuristic techniques (Colletti, 1999). Wiley (2001) was the first to apply GTTS. Other related research efforts include Barnes, Colletti, and Neuway (2001), Colletti and Barnes (1999), Colletti, Barnes, and Dokov (1999) and Jones (2002).

3 Problem Statement

A TDVRSP methodology should provide military planners with a tool that provides economically efficient vehicle routing and scheduling plans that achieve time definite delivery of requested cargo to customers. As detailed below, modeling requirements for the TDVRSP extend well beyond those of the typical VRSP.

Within a typical theater, there are multi-modal vehicles with differing characteristics such as cruising speed, capacity, cruising length, fixed and variable costs, route type (air, ground, water), and service times. Vehicle modeling requirements include the ability to make one or more trips during the planning horizon, the ability to perform direct delivery from outside the theater, and the ability to refuel at customer locations. Scheduling considerations include vehicle availability, service times, load times, and unload times. All vehicles operate from a depot or a hub.

Theater distribution network nodes are depots, hubs, and customers. Depots, aerial ports of debarkation (APODs), and seaports of debarkation (SPODs) are the supply nodes. The hubs, or support areas (SAs) which occur at corps (CSA), division (DSA), and brigade (BSA) levels, are transshipment nodes, which receive, store and distribute cargo. Customers are sink nodes that receive cargo. All nodes have time window constraints, fuel storage capacity and maximum on the ground (MOG) constraints. Hubs have cargo storage constraints and customers have cargo demand requirements and time definite delivery requirements.

A TDVRSP has three types of time window constraints: early time definite delivery (ETDD), time definite delivery (TDD), and multiple time windows for non-departure and non-arrival times (MTW). An ETDD stringently defines a customer service starting time but does not constrain vehicle arrival or departure times. A TDD defines when a customer service must be complete but does not constrain service occurrence, or vehicle arrival and departure times. MTWs restrict vehicle arrival and departure at a node but do not stipulate when vehicles are loaded or offloaded.

There are two types of maximum on the ground (MOG) constraints. A working MOG limits the number of vehicles that can simultaneously service a customer. A parking MOG limits the number of vehicles parked at a customer location.

The TDVRSP has a tiered distribution architecture. The first order tier contains the depots and customers/hubs served by the depots. Middle tiers consist of hubs that service customers/hubs. The last order tier consists of end customers served by a hub. Each tier is a self-contained distribution network. However, they are not independent of each other. Lower ordered tiers are dependent on higher ordered tiers. For example, the hubs in a lower ordered tier receive logistics as a customer within a higher ordered tier. Once a hub receives its cargo supply, it can then distribute the cargo to its customers.

Figure 1 presents an example of a TDVRSP. There are four Tiers within this network: Tier 0 is the APOD with customers (1,2,CSA); Tier 1 is the CSA hub with customers (DSA1, DSA2, 3); Tier 2 is the DSA1 hub with customers (BSA3, BSA4, and 5); and Tier 3 is the DSA2 hub with customers (BSA1, BSA2, and 4). Note that unlike a classical network hierarchy, both Tiers 2 and 3 derive from Tier 1.

Hubs distribute cargo after it is received and processed. Cargo, characterized by the amount delivered and time of delivery, is processed and prepared for delivery to its next customer. Cargo is either loaded directly onto available vehicles or stored for later delivery.

The TDVRSP primary objectives are to minimize unmet customer demand (demand shortfall), late deliveries (TDD shortfall), vehicle fixed costs, and vehicle variable costs. Late delivery times are weighted by the amount of demand delivered late. A fixed cost is charged for each vehicle used in the solution and variable vehicle costs are associated with vehicle travel.

4 TDVRSP Solution Methodology

GTTS has been used successfully to attack several difficult combinatorial optimization problems (Hall, 2000 ; Wiley, 2001; Combs, 2002; Crino, 2002). Crino shows that GTTS provides an efficient and effective means to find quality TDVRSP solutions. This section provides algebraic foundations, a description of how the TDVRSP is represented in the GTTS framework, and a description of the GTTS methodology employed.

4.1 Algebraic Foundations

Abstract groups are simply explained as sets of objects, together with a method of combining its elements that is subject to a few simple rules (Baumslag and Chandler, 1968).

A *semi-group* is a non empty set G together with a fixed binary operation $\oplus$ that satisfies the following conditions:

1. $x \oplus y \in G, \forall x,y \in G$; the operation is closed
2. $(x \oplus y) \oplus z = x \oplus (y \oplus z) \; \forall x,y,z \in G$; the operation is associative

A *group* is a semi-group that satisfies the additional conditions:

1. $\exists ! \, e \in G \ni \forall x \in G, e \oplus x = x \oplus e = x$; there exists a unique identity
2. For each $x \in G, \exists ! \, x^{-1} \in G \ni x^{-1} \oplus x = x \oplus x^{-1} = e$; there exists a unique inverse. Groups have some elementary properties that are used in this research. These properties, presented as a theorem with a proof in Barnard and Neill (1996), are presented below.

1. For $x, y \in G$, if $x \oplus y = e$, then $x = y^{-1}$ and $y = x^{-1}$
2. $(x \oplus y)^{-1} = y^{-1} \oplus x^{-1} \; \forall x,y \in G$
3. $(x^{-1})^{-1} = x \; \forall x \in G$
4. For $x, y, z \in$ G, if $z \oplus x = z \oplus y$, then $x = y$ and if $x \oplus z = y \oplus z$ then $x = y$

A *permutation of a set A* is a function from A into A, which is both one to one, and onto (Fraleigh, 1976).

The *symmetric group on n-letters, S_n* is the group of all permutations of set A if A is the finite set {1, 2, 3, … , n} (Fraleigh, 1976).

Since the TDVRSP is fundamentally a combinatorial optimization problem involving the partitioning and ordering of letters, S_n provides a natural setting for structuring the problem.

There are two different notations for S_n, the standard form and cyclic form. The standard form notation is a 2 by n array that represents a one to one and onto function whose domain (top row) and image (bottom row) are the integers $\{1,2,\ldots,n\}$ (Colletti, 1999). Let elements $\pi \in S_n$ be permutations. The notation is the array

$$\begin{array}{cccccc} 1 & 2 & 3 & 4 & 5 \; \ldots. & n \\ \pi(1) & \pi(2) & \pi(3) & \pi(4) & \pi(5) \; .\,.\,.\,. & \pi(n)\,. \end{array}$$

The cyclic form notation is a streamlined notation of S_n. Given $i \in \{1,2,\ldots,n\}$, $p \in \mathbf{N}$ and $\pi^p(i) = i$, the first cycle is represented as:

$$(i, \pi(i), \pi^2(i),\ldots,\pi^{p-1}(i)).$$

Equivalently, the cycle (i,j,k,l) means π sends i to j, j to k, k to l, and l back to i. The process continues by picking an element not in the cycle containing i and iterating the process until all members of $\{1,2,\ldots,n\}$ have been used. A cycle of length k is a cycle containing k elements (Sagan, 1991). Cycles with only one element are called unit cycles, where unit cycles can be implied and dropped from cyclic notation.

Examples of the standard form permutation and cyclic form permutation are presented below (Sagan, 1991).

If $\pi \in S_5$ is given by

$$\pi(1) = 2, \qquad \pi(2) = 3, \qquad \pi(3) = 1, \qquad \pi(4) = 4, \qquad \pi(5) = 5,$$

then the standard form is $\pi = \begin{pmatrix} x \\ \pi(x) \end{pmatrix} = \begin{pmatrix} 1 & 2 & 3 & 4 & 5 \\ 2 & 3 & 1 & 4 & 5 \end{pmatrix}$

and the cyclic form is $\pi = (1,2,3)(4)(5)$ or $\pi = (1,2,3)$. The length of the first cycle is 3 and the unit cycles are (4) and (5).

The binary operation for S_n is function composition. The product of two permutations, $\pi \oplus \sigma$, is composition. Given $x \in S_n$, then $(\pi \oplus \sigma)(x) = \sigma(\pi(x))$.

For example, let $\pi = \begin{pmatrix} x \\ \pi(x) \end{pmatrix} = \begin{pmatrix} 1 & 2 & 3 & 4 & 5 \\ 2 & 3 & 1 & 4 & 5 \end{pmatrix}$ and $\sigma = \begin{pmatrix} x \\ \sigma(x) \end{pmatrix} = \begin{pmatrix} 1 & 2 & 3 & 4 & 5 \\ 4 & 3 & 1 & 2 & 5 \end{pmatrix}$

then

$$\pi \oplus \sigma = \begin{pmatrix} 1 & 2 & 3 & 4 & 5 \\ 2 & 3 & 1 & 4 & 5 \end{pmatrix}\begin{pmatrix} 1 & 2 & 3 & 4 & 5 \\ 4 & 3 & 1 & 2 & 5 \end{pmatrix} = \begin{pmatrix} 1 & 2 & 3 & 4 & 5 \\ 3 & 1 & 4 & 2 & 5 \end{pmatrix}.$$

An operation used throughout this research as a means to execute TS moves is conjugation. Conjugation provides a relabeling of permutation letters while maintaining the original cyclic structure.

> *Conjugation* is an operation, denoted $x = y^k$, where $x = k^{-1} \oplus y \oplus k$ for some $x, y, k \in G$ where G is a group.

An example of conjugation as applied in TS is the operation between two permutations where one represents an incumbent solution, $x = (1,3,2)(4)(5)$, and the other a two-letter swap move, $k = (2,4)$. Conjugating x by k results in the solution $x^k = k^{-1} \oplus x \oplus k = (1,3,4)(2)(5)$, where the cyclic structure is maintained and only letters in the conjugator permutation are moved.

Group theory provides for the partitioning of groups into conjugacy classes, group actions and orbits. In general, any abstract group G inherently self-partitions into mutually exclusive and exhaustive conjugacy classes.

> *A conjugacy class,* CClass $(G, g) = \{g^h: h \in G\}$, is the set of all elements $\{h^{-1} \oplus g \oplus h: \ h \in G\}$ for $g \in G$.

Another feature of group theory that partitions S_n solution spaces are group actions and orbits. Given an abstract group G and a set X, a *group action*, denoted $_GX$ defines how G-elements operate upon X-elements to create X-elements (Colletti, 1999).

For $g \in$ G and $x \in$ X, let x^g denote the unique X-element that satisfies:

- $x^e = x$, $\forall\, x \in$ X (e is the G-identity)
- $\forall\, g,h \in$ G and $x \in$ X, $(x^g)^h = x^{gh}$

The "conjugation-like" notation $x^g \in X$ denotes the result of $g \in G$ acting upon $x \in X$. The user defines G and the result of x^g must be closed on X.

Let G be a group and X be a non empty set. For $x \in X$, an orbit is the set of all elements in X to which x can be moved by $g \in$ G. i.e., $x^G \equiv \{x^g\!: g \in G\}$. A group action inherently partitions X into *orbits*.

The *orbit* of $x \in X$ is x^G and denotes the G-orbit of x, Orbit (G, x). Since any two orbits of $_GX$ either coincide or are disjoint, orbits of a group action partition X. If X is a conjugacy class in S_n, orbits will exhaustively and exclusively partition the conjugacy classes (Colletti, 1999).

Specifically for this research, the set X is a CClass (S_n, x) which comprises the solution space of the TDVRSP. The group G is a user-defined subgroup of S_n that moves solution letters. The G-orbit of $x \in X$ defines a TS move neighborhood of x. A particular x^g defines a move neighbor of x. Each orbit is the neighborhood of any one of its permutations, and the members of an orbit share the same neighborhood (Colletti, 1999).

4.2 Using S_n to represent the TDVRSP

Applying GTTS to the TDVRSP requires that solutions be represented as elements of S_n. For a TDVRSP solution, each cyclic factor in the solution's disjoint cyclic structure represents a vehicle trip. For p = (1,4,5)(2,3) $\in S_5$, (1,4,5) represents the following tour: $1 \rightarrow 4 \rightarrow 5 \rightarrow 1$. For the TDVRSP, the first letter in the tour represents a vehicle. Subsequent letters represent the customers serviced by that vehicle. In this example, a vehicle leaves a depot/hub and services two customers before returning back to the depot/hub. A unit cycle represents either a vehicle letter not leaving the depot/hub or a customer letter not serviced. Permutations having a single factor represent TDVRSP solutions having one tour. Permutations with k factors represent k vehicles conducting at most k tours. The k vehicles could be from the same depot/hub or different depot/hubs depending on the vehicle location data.

In group theoretic notation, let C and V be the disjoint customer and vehicle letter-sets, respectively, where $|V| < |C|$, and let X be the permutations in S($C \cup V$) whose factors each contain a sole V-letter. The *first position* of any cycle in X contains the cycle's single V-letter, and the factors of $x \in X$ are arranged in ascending V-letter order, thus implying lexicographic ordering.

In a TDVRSP problem, vehicles can make multiple trips and customers can receive multiple services within a time period. A letter is allocated for each vehicle trip and each customer service. Letters are assigned sequentially for each vehicle or customer. For example, if vehicle A has the potential to make three trips, three letters are assigned to vehicle A's trips. If customer D has the potential to receive two services, two letters are assigned. Table 1 shows a letter assignment to vehicles *A*, *B*, and *C* and customers *A*, *B*, and *C*. Vehicle *A* has the potential for 3 trips, vehicle *B* can make 2 trips, and vehicle *C* can make 4 trips. Customer *A* has the potential for servicing 2 times, customer *B* can be serviced 3 times, and customer *C* may be serviced only once.

The network representation of Table 1's TDVRSP solution is presented in Figure 2. For the solution tour, vehicle trips occur in numerical order but customer services do not necessarily occur in numerical order. The solution is (1,10)(2,11,13)(4,12)(6,15) where there are 5 unused vehicle trips and one unused customer service letter.

Figure 3 presents another example of a TDVRSP solution. The hubs in this TDVRSP require a letter assignment scheme based on the tier levels within the distribution structure. The highest tiers are the depots and the customers they serve. The second highest tiers are the customers served by the hubs in the highest order. The third highest tiers are the customers serviced by the hubs in the second order. This hierarchical structure is replicated until all nodes are considered. Letter assignments are first given to the highest tier's vehicles. Subsequent assignment of vehicle letters moves to the next highest tier. This continues until vehicles for the lowest tier are assigned letters. Next, an identical procedure is followed for the assignment of customer letters.

The solution given in Figure 3 is

(1,14)(2,15,16)(3,17)(5,18)(7,19)(8,21)(9,22)(10,24)(11,25,26)

The implied unit cycles (4), (6), (12), and (13) are unused vehicle trip letters and unit cycles (20) and (23) are unused customer service letters.

In addition to yielding an efficient and effective solution methodology for the TDVRSP, the research documented here also provided *significant* advances in the development of GTTS as applied to vehicle routing and scheduling problems. These GTTS advances, described in detail in Crino (2002), included, for the *first* time, the formulation of move neighborhoods defined using groups to generate orbits. These orbits were used as a means to efficiently search the solution space. They eliminated cycling, prevented solution reevaluation, and avoided entrapment in local optima. This methodology prevents the search from being trapped in a subset of the solution space topology and eliminates the need for a restart mechanism. This technique allowed exhaustive non-repetitive search of each partition and, by placing a utilized orbit on an orbit tabu list, prevented reevaluation.

A unique solution space partition hierarchy was developed using the symmetric group on n-letters. Conjugacy classes, cyclic form structures, orbital planes, and orbits were defined that partition the solution space. Solution space partitions were exploited in the diversification and intensification process. In addition, neighborhoods were constructed to intelligently traverse the partitions and enable a potential search of the entire space. Group move neighborhoods steered the search between different orbits. Swap move neighborhoods traversed the search between different orbital planes. Insert and extraction move neighborhoods moved the search to different conjugacy classes and cyclic form structures.

Orbital planes were defined and used as a primary search mechanism of the GTTS. Orbital planes are orbits partitioned by orbits. They provide a more atomic partitioning of the solution space permitting partial or exhaustive search. Using orbital planes is a highly efficient special case of Colletti's (1999) orbital transversal method because the search may re-examine an orbit (orbital plane) without reevaluating solutions within the orbit (orbital plane).

4.3 TDVRSP Modeling Assumptions

It is a recognized fact that no "real-world" problem's inherent complexities can be captured in a usable model of manageable size. For this reason, while preserving the methodology's capability for practical planning purposes, a number of assumptions have been incorporated into the GTTS representation of the TDVRSP. The primary assumptions are listed below.

1. Sufficient supplies of required materiel are available at time 0 at all depot locations.
2. All cargo is aggregated into a single commodity and vehicle capacity restrictions are enforced.
3. All airfields accommodate all aircraft types
4. Each aircraft or ground vehicle type uses the same loading and unloading materiel handling equipment, personnel, and docking space for assessing working MOG constraints.
5. Unloading and loading requirements are independent of customer and load conditions.
6. Vehicle travel costs are independent of load weight.
7. Working MOG constraints are constant for each customer and depot.
8. Vehicles depart from and return to the same depot/hub.
9. All hubs have no supplies of materiel in storage at time 0.
10. Vehicles wait to depart hubs with full loads, unless customer demands require less than a full load.
11. Customers within the same tier are serviced exclusively from a single hub.
12. For a hub, working MOGs that unload vehicles are independent of working MOGs that load vehicles.

4.4 An Overview of GTTS

In this section, the GTTS architecture (Harder, 2000) used to solve the TDVRSP is presented. Figure 4 graphically depicts this architecture which is partitioned into a pre-TS phase and the TS phase. Figure 4 pictures the major components of each procedure within each phase. Due to space limitations, only an overview of the GTTS for the TDVRSP can be presented here. (The reader *desiring greater detail*, including algorithmic and code implementation is referred to Crino (2002) which may be requested from James T. Moore (james.moore@afit.edu)).

4.4.1 The Pre-TS Phase

The pre-TS phase achieves the following: (1) sets parameter values and assimilates a text file containing the customer and vehicle specifications for the current problem to be solved, (2) generates the group neighborhoods and (3) creates and evaluates an initial solution.

Generation of the group neighborhoods partitions the customer service letters C into disjoint subsets C_i, $i=1,2,\ldots,n$, where different customers are represented in each C_i. The process then generates the mutually exclusive groups, $G_i = S(C_i)$, the symmetric group on the letters in C_i. The G_i are stored within the *move neighborhood generator* for future use as move neighborhoods.

A greedy assignment heuristic creates the initial solution by assigning prioritized customers to vehicles that best meet their demands. Customers are assigned priority ratings using the following equation:

$$CustomerPriorityRating = \frac{custDist}{avgDist} * \sum_{i=1}^{n} [custDemand_i * (periodLength / TDD_i)(n-i)]$$

where i = TDD requirement index per customer
n = number of TDD requirements per customer
TDD_i = customer TDD requirement for index i
custDist = distance of customer to nearest depot/hub
avgDist = average distance of all customers to their nearest depot
$custDemand_i$ = customer demand
periodLength = total model time period

Vehicles are ordered based on the ratio of their capacity and average trip time.

$$VehCapPerAvgTripTime = vehCap/(2* avgDist/speed + loadTime + unloadTime + servTime)$$

where *vehCap* = vehicle capacity
speed = vehicle cruising speed
loadTime = time to load a vehicle
unloadTime = time to unload a vehicle
servTime = time to service a vehicle

Once created, the initial solution, or first *incumbent solution*, is evaluated according to an objective function that assesses the demand filled shortfall, TDD shortfall, fixed costs, variable costs and other penalties.

4.4.2 The TS Phase

Running until a termination criterion is satisfied, each iteration of the TS phase passes through five major components; *move neighborhood generator, solution evaluator, TS strategy manager, tabu list manager,* and *perform move operator*. An iteration begins by generating and applying a move neighborhood to the incumbent solution and ends when the *perform move operator* creates a new incumbent solution.

The *move neighborhood generator* creates and applies move neighborhoods to the incumbent solution. Move neighborhoods are generated and employed based on solution attributes and data collection. The types of move neighborhoods generated are:

- *Intra orbital plane mutually exclusive group action neighborhood*
- *Inter orbital plane swap move neighborhood*
- *Fill demand insert move neighborhood*
- *Inter conjugacy class insert move neighborhood*
- *Inter conjugacy class extraction move neighborhood*

The *intra orbital plane mutually exclusive group neighborhood* is a collection of move neighborhoods generated by mutually exclusive groups created in the pre-TS phase. They are the primary search mechanism used by the TS strategy during normal search cycles. The *inter orbital plane swap move neighborhood* is a collection of 2-letter swap moves, where each letter is from a disjoint set of customer letters, C_i, $i = 1,2,\ldots,n$. This neighborhood provides move combinations not offered by the group neighborhoods. The *fill demand insert move neighborhood* is a collection of moves that reduce the demand shortfall. The *inter conjugacy class insert move neighborhood* consists of *inter cycle insert moves* that diversify the search by moving to a new conjugacy class. This neighborhood also uses solution attributes to create moves that potentially decrease the solutions' demand shortfall or TDD shortfall. Finally, the *inter conjugacy class extraction move neighborhood* consists of extraction moves that diversify the search by moving into new conjugacy classes. This neighborhood extracts customer letters from cycles in order to reduce fixed and variable costs. It is specifically called when excess customer letters reside in cycles and when the *TS strategy manager* dictates implementing super diversification measures.

Working closely with the *tabu list manager*, the *solution evaluator* determines the objective function value for each p-neighbor, q, where $q = (p \oplus move, p^{move})\ \forall\ move \in MoveNeighborhood$. The goal of this process is to find the best non-tabu q to replace the incumbent solution p. The *solution evaluator* has four main components: the permutation preprocessor, the vehicle loader/scheduler heuristic, the objective function evaluator, and the solution comparison method. The *solution evaluator* iterates through the four components for all q.

The permutation preprocessor simply transforms the incumbent solution's permutation representation into that of a defined neighboring solution. The vehicle loader/scheduler heuristic determines the vehicle loads and delivery schedules by using constraint-programming techniques to determine the loads and schedules for each vehicle. A vehicle is loaded to its specified capacity unless no more supply exists at the depot/hub or the customers in the tour do not demand a full load. The amounts of supply customers receive are determined by one of two options. The first option, until the vehicle is empty, provides a customer with his total demand based on the trip order. The second option equally splits the vehicle load between all customers in the trip. If a customer does not demand its share, then its share is distributed equally among the remaining customers. Vehicle schedules are determined based on travel times, service times, time constraints, and working MOG constraints. A vehicle schedule begins at its available time and is influenced thereafter by such things as customer time and MOG constraints along its trip.

The final results provided by the vehicle loader/scheduler heuristic's are complete details for each customer regarding each servicing vehicle's visit and complete details for each vehicle used including timing and routing information, and customers served on each vehicle trip. (The vehicle loader/scheduler heuristic clearly does not guarantee that the optimal use of vehicles is found for the neighborhood solution being investigated. Fortunately, this heuristic performed sufficiently well for the overall GTTS algorithm to provide excellent solutions to the overall optimization problem.)

As stated earlier, the objective function evaluator measures demand shortfall, time definite delivery (TDD) shortfall, fixed costs, variable costs, and penalty costs. Penalty costs include parking MOG violations and hub storage capacity violations.

The *tabu list manager* uses the *orbit* and *move* lists to interact with the *solution evaluator* to prevent cycling within the TS process. The orbit list tracks traversed orbits and the move list tracks recent diversification moves. Both lists allow an element's tabu status to be determined and changed.

The *TS strategy manager* determines whether to continue with normal TS processes, to intensify or to diversify the search. Decisions are based on collected search data and search parameters. Intensification occurs in conjugacy classes that generate good solutions. Diversification, in a normal or "super" form, occurs when the search process needs to explore different conjugacy classes. Normal diversification imparts a new search direction towards a different solution space partition. Super diversification departs the current local search areas where moves tend to be un-improving.

Figure 5 presents an outline pseudocode of the GTTS algorithm.

5 A Detailed TDVRSP Example with Computational Results

Prior to the research documented here, no benchmark problems existed for the TDVRSP. Crino (2002) used valid experimental design methodology to construct a set of 39 problems that effectively test the robustness of the GTTS TDVRSP algorithm. The problems are composed of three TDVRSP types: the Air Force multiple trips multiple services (MTMS) without hub, the Joint MTMS without hub, and the Joint MTMS with hub and other defining constraints. The 39 problems were further categorized by problem size (small, medium and large), delivery restriction density (low, medium, and high), and demand to capacity ratio (low, medium and high).

Figure 6 presents Crino's benchmark problem 37, one of the large Joint MTMS problems with hub and other defining constraints. Only customers 0, 1, 2, and 3 can accommodate Air Force cargo aircraft and all aircraft are located at the APOD. Table 2 gives the tier structure for the MTMS with hub instance. Tables 3, 4, and 5 display data for the 31 customers whose demands range from 65 to 1800 tons. The working and parking MOGs vary in size from 1–3 for the working MOG-for-aircraft (MOGA) and MOG-for-ground-vehicles (MOGG) and 2 – 6 for the parking MOGA. The APOD is restricted to a working MOGA = 6 and the SPOD has a working MOGG = 8. Some customers have ETDD constraints > 0 and some have time windows that restrict arrival and departure times. Table 4 presents TDD requirements for the customers with more than one TDD. Table 5 displays time windows for the APOD and customer 2, which restrict aircraft arrival and departure within those windows.

An example of the vehicle data is displayed in Table 6. There are a total of 90 vehicles. There are two aircraft types and five ground vehicle types. The aircraft closely resemble the C-17 and C-130 cargo aircraft. The ground vehicles represent heavy and medium fleet assets. The capacity is in tons, speed is in miles per hour, and service, load, and unload times are in hours. Available time is in hours relative to the model start time and the cruise lengths are in miles.

The summary problem results are given in Table 7. Examples of detailed vehicle loads, schedules, and customer deliveries are displayed in Tables 8 and 9. Due to route length constraints, some vehicles refueled at customer locations to have enough fuel to return to their depot/hub. Customers 0, 1, and 2 provided 178.6, 146.3 and 384.3 gallons of fuel to ground vehicles.

The total run time for 1,000 iterations was 110 minutes, where the best solution was found at iteration 709 (62.8 minutes). However, competitive satisfactory solutions, where all customer demands were satisfied, were found much earlier in the search process. Two such solutions are documented in Table 10.

Problem 37's results are indicative of the timely, effective, and robust results provided by the GTTS approach for all 39 benchmark problems. The GTTS method has clearly displayed its ability to solve the multiple objective TDVRSP problem.

Figure 7 displays the TS objective function values for each of the 1,000 iterations. The chart evidences a "healthy" TS search process that moves through different areas of the solution space as reflected by the variation of the objective function values.

6 Concluding Remarks

The development of an automated solution methodology to the TDVRSP problem has been characterized as a major priority and immediate need for the US military services.

This paper documents groundbreaking new theoretical results based in a flexible group theoretic tabu search (GTTS) framework and presents a powerful software implementation of those results. This marriage of theory and application has resulted in a robust, efficient, and effective generalized theater distribution methodology. This methodology evaluates and determines the routing and scheduling of multi-modal theater transportation assets at the *individual asset operational level* to provide economically efficient time definite delivery of cargo to customers.

Many software programs are available that "perform" theater distribution modeling. However, all, but the technique presented in this paper, are simulation models and simulations can not prescribe highly effective, near optimal routes and schedules for all vehicular transport assets. Therefore, this model is the *first* of its kind to offer this functionality.

In a recent study funded by HQ USAF (HQ USAF/ILA, 2002), a number of recommendations were made to the Air Force on types of models that could support an automated theater distribution management system. The purpose of the system is to "optimize" the entire theater distribution system, from the number of bases and vehicles, down to the vehicle routes and schedules. They concluded that vehicle routing and scheduling was very difficult to optimize, and development of an optimization approach is considered a high-risk methodology. Therefore, they proposed a low risk method using simulation. Unfortunately, they lose the functional requirements of prescribing detailed vehicular routes and schedules when using simulation. The HQ USAF study (HQ USAF/ILA, 2002) validates the importance and magnitude of this research. What was regarded as *too difficult* has been successfully ***created and developed*** in this research.

Bibliography

Barnard, T. and H. Neill. *Mathematical Groups*. Berkshire: Cox and Wyman Ltd, 1996.

Barnes, J. W., B. Colletti, and D.L. Neuway. "Using Group Theory And Transition Matrices For Conjugative Move Disciplines In Multiple TSP's With Known Solution Structure," *European Journal of Operational Research*, 138, (2002)

Baumslag, B. and B. Chandler. *Group Theory: Schaum's Outline Series*. McGraw Hill, 1968.

Brooks, Roger. "Position Paper on the Theater Distribution Management System (TDMS) Initiative" Headquarters, United States Air Force, Washington D.C., 26 May 2000.

Brown, Darin T. *Routing unmanned aerial vehicles while considering general restricted operating zones*, Masters Thesis, Air Force Institute of Technology, 2001

Calhoun, Kevin M. *A tabu search for scheduling and rescheduling combat aircraft*, Masters Thesis, Air Force Institute of Technology, 2000.

Calhoun, Kevin M., Richard F. Deckro, James T. Moore, James W. Chrissis, and John C. Van Hove, "Planning and Re-planning in Project and Production Scheduling", *Omega* 30 (3): 155-170 (2002).

Capehart, S. R. *A Tabu Search Metaheuristic for the Air Refueling Tanker Assignment Problem*, Masters Thesis, Air Force Institute of Technology, 2000.

Colletti, B. W. *Group Theory and Metaheuristics*. Ph.D. dissertation, The University of Texas at Austin, 1999.

Colletti, B. W., J.W. Barnes, and S. Dokov. "A Note on Characterizing the k-OPT Neighborhood Via Group Theory," *Journal of Heuristics:* 5, (1999).

Crino, John R. *A Group Theoretic Tabu Search Methodology For Solving Theater Distribution Vehicle Routing and Scheduling Problems*, Ph.D. Dissertation, AFIT, 2002.

Crino, J. , J. Moore, Colletti, B. , and W Barnes, "Using Group Theory to Construct Powerful Search Neighborhoods for Partitioning and Ordering Problems", working paper July 2002, UT-Austin & AFIT.

Cullenbine, Christopher A. *A tabu search approach to the weapons assignment model*, Masters Thesis, Air Force Institute of Technology, 2000.

Department of the Chairman of the Joint Chiefs of Staff. *Department of Defense Dictionary of Military and Associated Terms.* Joint Publication 1-02.

Washington: CJCS, 9 September1993.

Department of the Chairman of the Joint Chiefs of Staff. *Joint Tactics, Techniques, and Procedures for Joint Theater Distribution.* Joint Publication 4-01.4. Washington: CJCS, 22 August 2000.

Fraleigh, John B. *A First Course in Abstract Algebra*, Addison-Wesley, Reading MA, 1976.

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

Gomory, Ralph E. "An Algorithm to Integer Solutions to Linear Programs," In *Recent Advances in Mathematical Programming*, R. Graves, P.Wolfe (eds), McGraw Hill, New York, 269-302 (1963)

Gomory, Ralph E. "Outline of an Algorithm for Integer Solutions to Linear Programs," *Bulletin of the American Mathematical Society*: 64, 275-278 (1958).

Hall, S. N.. *A Group Theoretic Tabu Search Approach to the Traveling Salesman Problem.* Masters Thesis, Air Force Institute of Technology, 2000.

HQ USAF/ILA. "Theater Distribution Management System Requirements and Feasibility Study", 15 January, 2002.

Jones, Steven K. *A Highly Effective and Efficient Method for Neighborhood Evaluation in Metaheuristic Search*, Masters Thesis, University of Texas at Austin (2002).

Kaminski, Paul G. "The Revolution in Defense Logistics", Keynote address of the USAD for Acquisition and Technology, Alexandria, VA, 31 October 1995.

Kinney, Gary W. *A hybrid jump search and tabu search metaheuristic for the unmanned aerial vehicle (UAV) routing problem*, asters Thesis, Air Force Institute of Technology, 2000.

Laporte, G., M. Gendreau, J.Y. Potvin and F. Semet. "Classical And Modern Heuristics For The Vehicle Routing Problem," *International Transactions in Operational Research:*7(4-5), 285-300 (2000).

O'Rourke, Kevin P. *Dynamic unmanned aerial vehicle (UAV) routing with a Java-encoded reactive tabu search metaheuristic*, Master's thesis, Air Force Institute of Technology, 1999

Ryer, D. M. *Implementation of the Metaheuristic Tabu Search in Route Selection for Mobility Analysis Support System*, Masters Thesis, Air Force Institute of Technology, 1999.

Schrady, David. *Combatant Logistics Command and Control for the Joint Force Commander.* Navy report, 1999.

Synergy, *Theater Distribution Management System Requirements and Feasibility Study*, 2002.

Wiley, V. D. *The Aerial Refueling Problem – A Doctoral Dissertation Proposal*. The University of Texas at Austin, 2000.

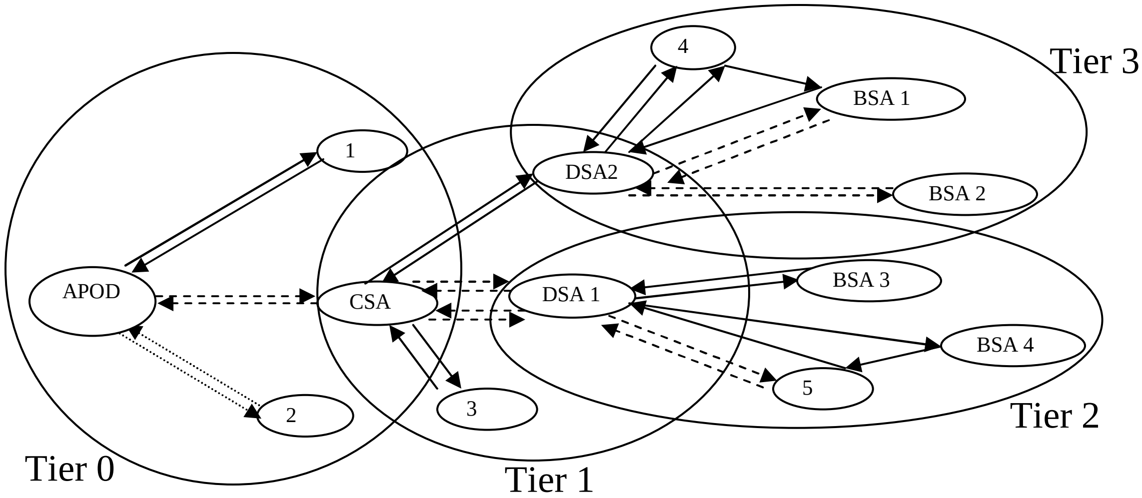

Figure 1 Example TDVRSP

Table 1. S_n Letter Assignment Example

	Number of trips/services	Assigned letters
Vehicle *A*	3	1,2,3
Vehicle *B*	2	4,5
Vehicle *C*	4	6,7,8,9
Customer *A*	2	10,11
Customer *B*	3	12,13,14
Customer *C*	1	15

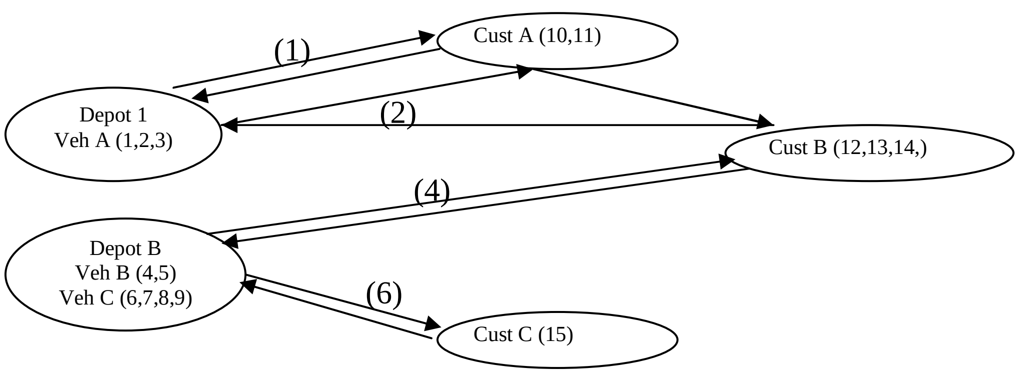

Figure 2 TDVRSP Solution Tour Example

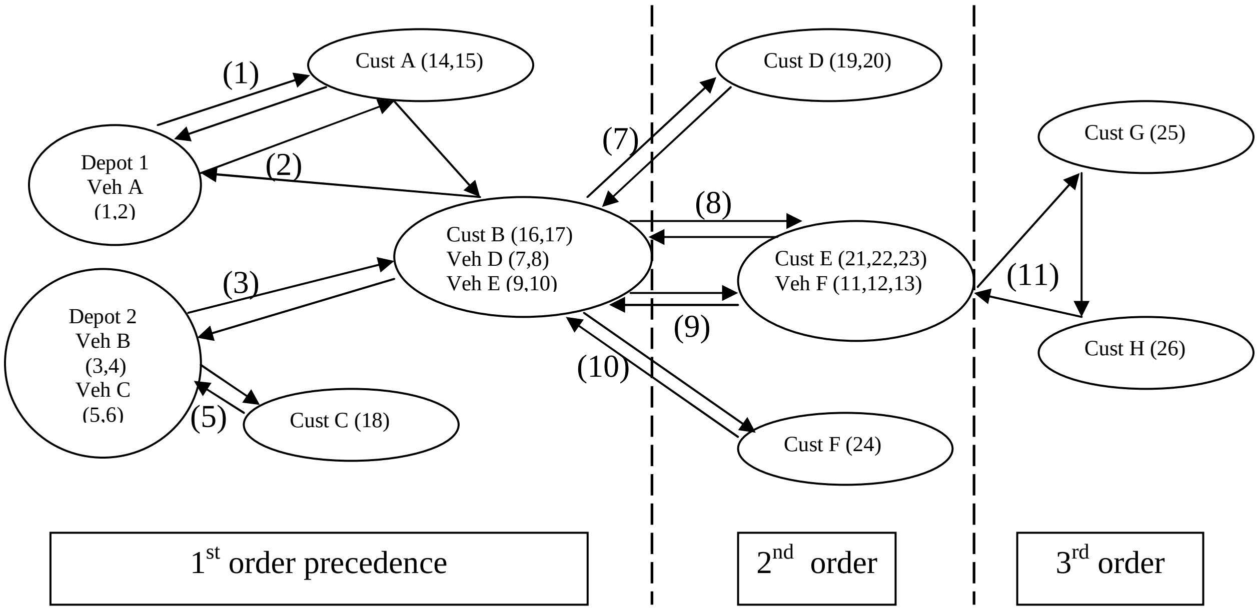

Figure 3 Solution Tour With Hubs Example

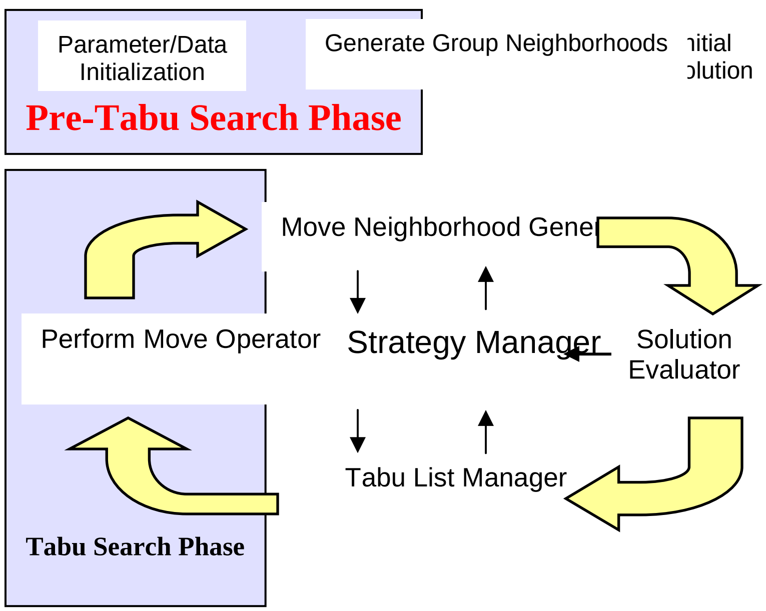

Figure 4 Group Theoretic Tabu Search Architecture

Obtain initial solution & initialize parameters
p = Incumbent solution = initial solution
WHILE iteration_count <= MaxLoops, do:

    Increment iteration_count
    Select MoveNeighborhood

    a. If *super diversification counter* ≥ *superDiversifyTolerance* and *diversifyingCounter* ≤ *superDiversifyMoves*
        i. Select *inter- conjugacy class extraction move neighborhood*
    b. Else If *worsening move counter* ≤ *worseningMoveTolerance* or *constant move counter* ≤ *constantMoveTolerance*
        i. Select next sequential group neighborhood, or
        ii. Select swap move neighborhood at end of group sequence
        iii. If strictly intensifying in an orbital plane, do not include step ii.
    c. Else If intensificationOn = true
        Select swap move neighborhood
    d. Else (select and generate a diversification move neighborhood)
        i. If solution demand shortfall > 0
            Select *fill demand move neighborhood*
        ii. Else If unnecessary vehicle trips > 1 or customer services > 2
            Select *inter- conjugacy class extraction move neighborhood*
        iii. Else If TDD shortfall > 0
            1. Select *inter- conjugacy class extraction move neighborhood*, or (alternate)
            2. Select *inter- conjugacy class insert move neighborhood*
        iv. Else
            Select *inter- orbital plane swap move neighborhood*

    Evaluate $q = p \oplus move$ or $q = p^{move}$ $\forall$ $move \in$ *MoveNeighborhood*.
    Select best *q* based on objective function values and tabu status.
    Register evaluated orbit, conjugacy class, and/or move in tabu list
    Perform $p \oplus move$ or p^{move} and generate a new incumbent solution, *p*
    Determine TS type, normal or intensification

    e. If *TS counter* = *iterations*,
        i. Begin intensification search process
        ii. Set *p* = an elite list permutation
        iii. *IntensificationOn* = true
    f. Else If *intensification TS counter* = *intensificationIterations*,
        i. Begin normal TS process
        ii. Set *intensificationOn* = false.

End while
End

Figure 5 – An Outline Pseudocode of the GTTS TDVRSP Algorithm

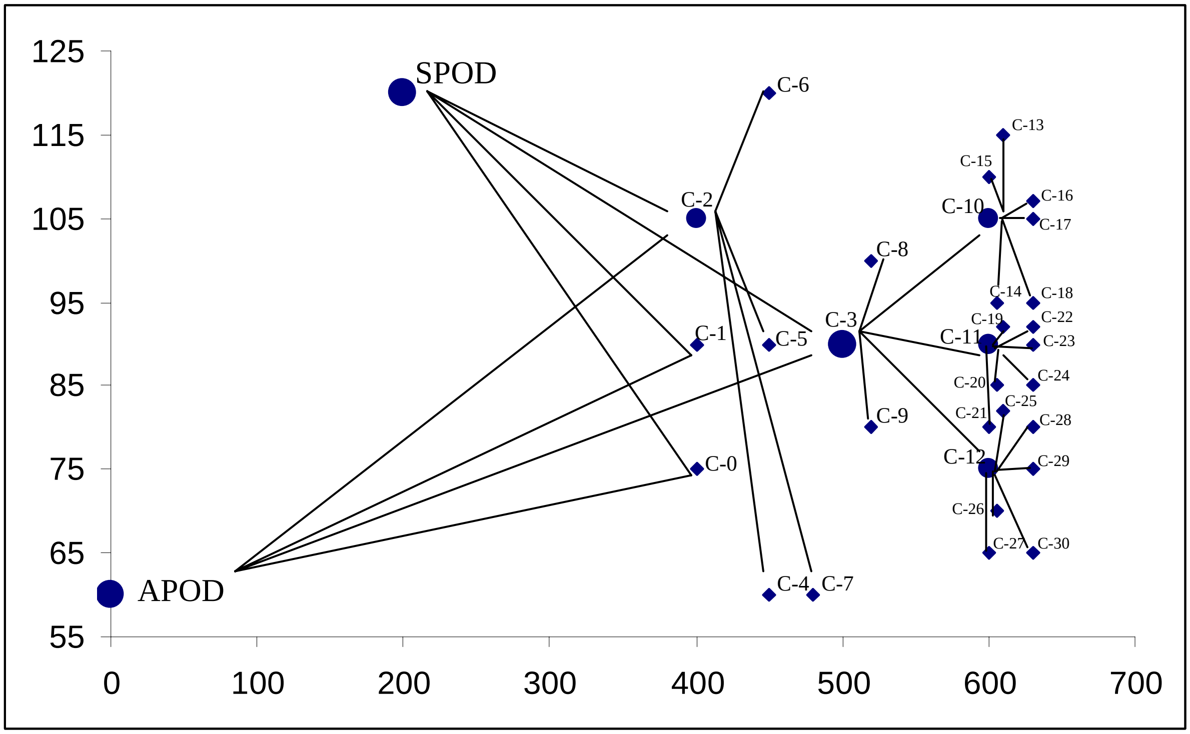

Figure 6 TDVRSP Depot, Hub, and Customer Locations

Table 2 TDVRSP Tier Structure

Tier	Supply source	Customers
0	APOD, SPOD	C-0, C-1, C-2, C-3
1	C-2	C-4, C-5, C-6, C-7
2	C-3	C-8, C-9, C-10, C-11, C-12
3	C-10	C-13, C-14, C-15, C-16, C-17, C-18
4	C-11	C-19, C-20, C-21, C-22, C-23, C-24
5	C-12	C-25, C-26, C-27, C-28, C-29, C-30

Table 3 TDVRSP Customer Data

Cust	Location Coordinates		Demand	wMOGA,wMOGG,pMOGA			eTDD	TDD	Hub	Tier	Prior	Storage	ACfuel	Gfuel
0	400	75	300	1	3	2	0	96	0	0	1	0	0	200
1	400	90	300	2	3	4	0	96	0	0	1	0	0	200
2	400	105	800	3	3	6	0	96	1	0	1	200	2000	400
3	500	90	1800	3	3	6	0	96	2	0	1	300	8000	500
4	450	60	200	0	2	0	0	96	0	1	1	0	0	200
5	450	90	200	0	2	0	0	96	0	1	1	0	0	200
6	450	120	200	0	1	0	36	96	0	1	1	0	0	200
7	480	60	200	0	1	0	24	96	0	1	1	0	0	200
8	520	100	75	0	1	0	0	48	0	2	1	0	0	50
9	520	80	75	0	2	0	0	72	0	2	1	0	0	50
10	600	105	550	0	3	0	0	96	3	2	1	0	0	200
11	600	90	550	0	3	0	0	96	4	2	1	0	0	200
12	600	75	550	0	3	0	0	96	5	2	1	0	0	200
13	610	115	65	0	1	0	0	24	0	3	1	0	0	50
14	605	95	70	0	2	0	0	36	0	3	1	0	0	50
15	600	110	70	0	2	0	0	72	0	3	1	0	0	50
16	630	107	115	0	3	0	0	30	0	3	1	0	0	50
17	630	105	115	0	3	0	0	66	0	3	1	0	0	50
18	630	95	115	0	3	0	0	96	0	3	1	0	0	50
19	610	92	65	0	1	0	24	70	0	4	1	0	0	50
20	605	85	70	0	2	0	24	96	0	4	1	0	0	50
21	600	80	70	0	3	0	24	96	0	4	1	0	0	50
22	630	92	115	0	1	0	24	96	0	4	1	0	0	50
23	630	90	115	0	3	0	24	36	0	4	1	0	0	50
24	630	85	115	0	3	0	24	90	0	4	1	0	0	50
25	610	82	65	0	1	0	48	96	0	5	1	0	0	50
26	605	70	70	0	2	0	48	96	0	5	1	0	0	50
27	600	65	70	0	3	0	48	96	0	5	1	0	0	50
28	630	80	115	0	2	0	48	72	0	5	1	0	0	50
29	630	75	115	0	3	0	48	96	0	5	1	0	0	50
30	630	65	115	0	3	0	48	72	0	5	1	0	0	50

Table 4 Customers with Multiple TDD Requirements

cust	cummulative demand	ETDD	TDD
0	69	0	36
0	142	0	60
0	219	0	84
0	300	0	96
1	70	0	36
1	150	0	60
1	215	0	84
1	300	0	96
4	90	0	36
4	200	0	96
5	110	0	60
5	200	0	96
7	80	24	60
7	200	24	96

Table 5 Customer Time Windows

cust	AC-timewindows		G- time windows	
APOD	24	30	0	0
APOD	48	54	0	0
APOD	72	78	0	0
2	24	30	0	0
2	48	54	0	0
2	72	78	0	0

Table 6 Vehicle Data

Veh Type	Capacity	Speed	Fixed Cost	Var Cost	**Cruise Length**	Service Time	**Load Time**	**Unload Time**
A	85	470	10	.05	9000	2	4	2
A	12	330	2	.02	3000	1	2	1
G	28	45	1	.01	300	2	3	1
G	16	60	1	.01	300	1	2	1

Table 7 Evaluation Summary

Total Demand Shortfall	0.0
Sum TDD Shortfall (weighted)	3.41
Fixed Cost	106.5
Variable Cost	112.98
Storage Constraint Penalty	3.0
MOG Parking Penalty	0.0
Total cost	**225.88**

Table 8 Vehicle Schedule Example

Vehicle	Cust/dep	Arr Time	Start Off/Onload	End Off/Onload	Depart	Delivery
4		0	0	4	4	
4	3	5.07	5.07	7.07	7.07	85
4		8.13	10.13	14.13	14.13	
4	3	15.2	15.2	17.2	17.2	85
4		18.26	20.26	24.26	30	
4	3	31.07	31.07	33.07	33.07	85
4		34.13	36.13	40.13	40.13	
4	3	41.2	41.2	43.2	43.2	85
4		44.26	46.26	50.26	54	
4	3	55.07	55.07	57.07	57.07	85
4		58.13	60.13	64.13	64.13	
4	3	65.2	65.2	67.2	67.2	56

Table 9 Customer Delivery Example

Customer	Vehicle	Delivery amount	Arr	Offload	Dep
6	39	4	22.98	36	37
6	34	28	27.32	37	38
6	38	4	31.87	38	39
6	37	8	37.79	39	40
6	39	4	41.26	41.26	42.26
6	34	28	44.59	44.59	45.59
6	36	12	59.83	59.83	60.83
6	33	28	61.15	61.15	62.15
6	35	16	64.49	64.49	65.49
6	36	12	64.77	65.49	66.49
6	37	8	65.74	66.49	67.49
6	38	4	65.78	67.49	68.49
6	33	28	68.74	68.74	69.74
6	35	16	69.44	69.74	70.74

Table 10 Example Solutions

Demand Shortfall	TDD Shortfall	Fixed Cost	Variable Cost	Total Cost	Time	Iteration
0.0	9.02	108.0	113.91	251.94	11.5	145
0.0	4.71	107.5	113.43	230.64	24.5	290
0.0	3.41	106.5	112.98	225.88	62.75	709

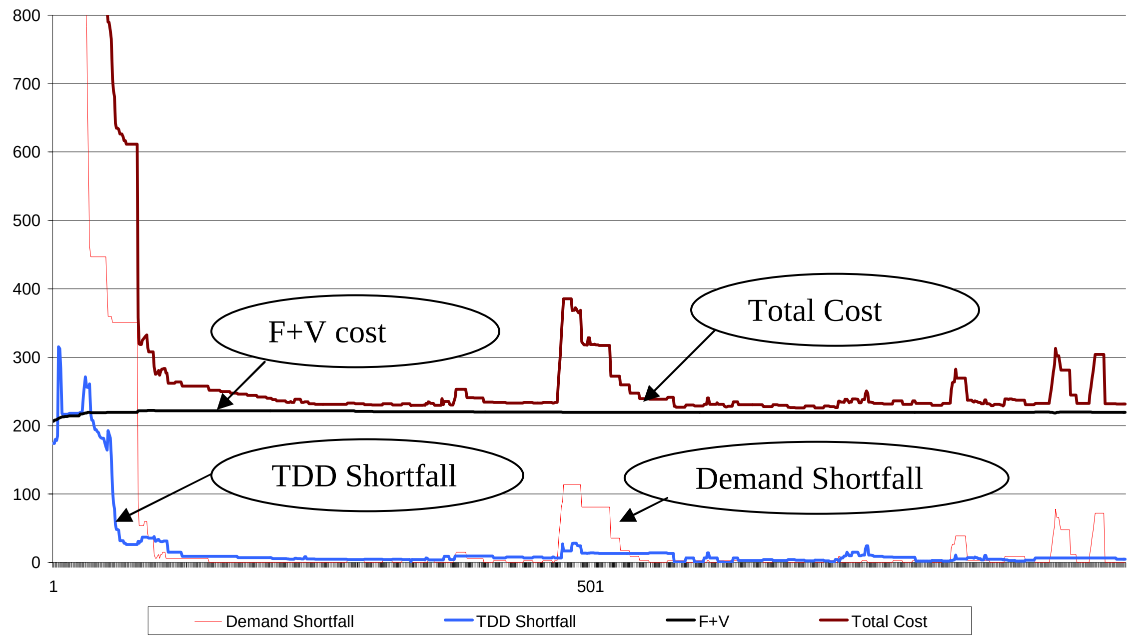

Figure 7 TDVRSP Objective Function Values