ROUTING UNMANNED AERIAL VEHICLES
WHILE CONSIDERING
GENERAL RESTRICTED OPERATING ZONES

Darin T. Brown, 1st Lieutenant, USAF

AFIT/GOR/ENS/01M-04

DEPARTMENT OF THE AIR FORCE
AIR UNIVERSITY

AIR FORCE INSTITUTE OF TECHNOLOGY

Wright-Patterson Air Force Base, Ohio

AFIT/GOR/ENS/01M-04

ROUTING UNMMANNED AERIAL VEHICLES WHILE CONSIDERING GENERAL RESTRICTED OPERATING ZONES

THESIS

Presented to the Faculty

Department of Operational Sciences

Graduate School of Engineering and Management

Air Force Institute of Technology

Air University

Air Education and Training Command

In Partial Fulfillment of the Requirements for the

Degree of Master of Science in Operations Research

Darin T. Brown

1st Lieutenant, USAF

March 2001

APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED

AFIT/GOR/ENS/01M-04

ROUTING UNMANNED AERIAL VEHICLES WHILE CONSIDERING GENERAL RESTRICTED OPERATING ZONES

Darin T. Brown
1st Lieutenant, USAF

Approved

Raymond Hill

Raymond Hill, Lt Col, USAF (Advisor)
Assistant Professor of Operations Research
Department of Operational Sciences

16 Mar 01
date

James T. Moore

Dr. James Moore, PhD
Associate Professor of Operations Research
Department of Operational Sciences

16 March 2001
date

ACKNOWLEDGEMENTS

All the thanks in the world go to my wonderful wife. She carried the load at home allowing me to devote myself to AFIT. Thanks to my advisor, Lt Col Ray Hill, for his advice, understanding, and patience. Thanks to Dr. James Moore for insight. I also want to thank Lt Robert Harder and Capt Gary Kinney for paving the way; their good work was a joy to follow up.

Finally, a special thanks goes to all the wonderful friends I met here at AFIT. I am a better person for knowing all of you. Thank you.

Darin T. Brown

TABLE OF CONTENTS

LIST OF FIGURES

Page

LIST OF TABLES

AFIT/GOR/ENS/01M-04

ABSTRACT

US military forces employ unmanned aerial vehicles (UAVs) to conduct intelligence-gathering missions worldwide. For a typical mission, commanders may task UAV operators to gather imagery on 100 or more sites or targets. UAV operators must quickly prepare mission plans that meet the needs of their commanders while dealing with real-world constraints such as time windows, site priorities, imagery requirements, UAVs with different capabilities (i.e. imagery equipment, speed, and range), and UAVs departing from different bases. Previous AFIT research provided the UAV Battlelab with a tool, AFIT Router, for generating high-quality routes to aid mission planning. This research enhances the AFIT Router by providing the ability to define general restricted operating zones and to build routes that consider these zones. This research also examines and compares a probabilistic tabu search heuristic and two reactive tabu search heuristics for solving vehicle routing problems.

Keywords: Air Force Research, Operations Research, Unmanned Aerial Vehicles, Remotely Piloted Vehicles, Surveillance Drones, Routing Around Obstacles, Combinatorial Analysis, Algorithms, Heuristics, Tabu Search, Vehicle Routing Problem, Traveling Salesman Problem, Multiple Depots, Time Windows, Java.

ROUTING UNMANNED AERIAL VEHICLE WHILE CONSIDERING GENERAL RESTRICTED OPERATING ZONES

1. Introduction

1.1 Background

America's armed forces employ unmanned aerial vehicles (UAVs) to perform intelligence-gathering missions worldwide. A UAV, as its name implies, is an aerial vehicle with no onboard pilot that is capable of preprogrammed autonomous operation or operations received from a human operator in a control station located some distance from the vehicle (Renehan 1997). Currently, the US Air Force uses the Predator UAV (see Figure 1).

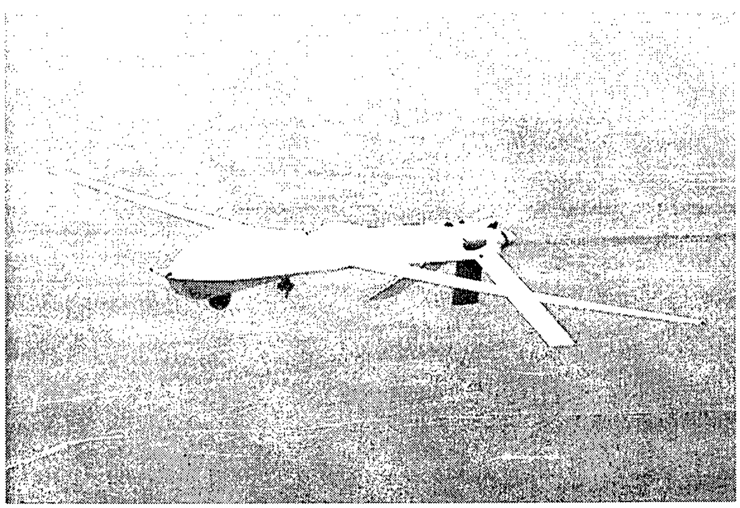

Figure 1: Predator UAV in flight

The Predator can remain airborne for extended periods—it has an endurance capability that exceeds 40-hours. With a cruising speed of 70-knots, this endurance translates to a 500-nautical mile operational radius. A UAV pilot, known as an air vehicle operator (AVO), remotely flies the Predator from a ground control station (GCS). The Predator transmits high-resolution video and synthetic aperture radar images of targets back to the GCS via line-of-sight communication and satellite link. Commanders then use these images for reconnaissance, surveillance, and target acquisition. In the Kosovo conflict, UAVs provided critical imagery that allowed more precise targeting and spared lives (Canan 1999).

The UAV mission is extremely dynamic. A typical UAV mission may require imaging hundreds of targets or sites within specified time windows. The AVOs must create flight plans for each of their UAVs. These flight plans must consider the following: time windows, time required to image a site, vehicle type, vehicle range, vehicle departing from different bases, no-fly zones, and bad weather. During the course of a mission, new targets may crop up. Thus, the AVOs must then reroute their UAVs in real-time to accommodate the additional requirements. AVOs have tools for route planning; however, these tools are unable to determine routes that minimize mission length and number of vehicles used.

1.2 Problem Statement

Previous research by O'Rourke (1999), Flood (1999), Kinney (2000), and Harder (2000) provided support tools to the UAV Battlelab that quickly generate near-optimal tours, minimizing mission length and number of vehicles used. The AFIT Router software created by Harder is the most recent of these support tools. The software uses an adaptive tabu search heuristic to find routes for UAVs to assigned sites. The software allows the user to model the UAV routing problem involving the following: multiple vehicle types, vehicles departing from different bases, time windows for site availability, time walls to model when visitation to a site is restricted, site priorities, and restricted operating zones (ROZ). A ROZ is used to model no-fly zones, threats, or areas of bad weather; basically, places where UAV flight is prohibited or restricted. The current model provides for restricted operating zones but merely uses them to restrict when sites within an active ROZ may be visited. Previous AFIT researchers defined the UAV routing problem as a vehicle routing problem with side constraints specific to operating UAVs. This research extends this definition by adding the task of building routes that account for flying around restricted operating zones and by examining extensions to the AFIT Router tabu search heuristic.

1.3 Scope and Contribution

This research continues the efforts by O'Rourke (1999), Kinney (2000), and Harder (2000) in support of the UAV Battlelab. This effort provides a modified version of the AFIT Router software that accounts for routing around ROZs. This effort also compares the tabu search heuristic provided by Harder with a probabilistic tabu search and two reactive tabu search heuristics.

Like my predecessors, this research does not account for any flight profile planning aspects of the UAV mission such as turning radii or approach angles. This detail is left to the AVOs and their GCS tools. Also, we do not account for terrain. Terrain may affect route feasibility, but we will assume terrain has no effect in order to continue returning solutions quickly. However, with the improvements this research provides, it may be possible to model terrain using a ROZ.

1.4 Overview

Chapter 2 presents a brief review of literature relating to this research. Chapter 3 presents a proposed methodology for this research. Chapter 4 presents the testing and analysis of our algorithms. Chapter 5 provides a brief summary of this research and recommendations for future research.

2. LITERATURE REVIEW

2.1 Traveling Salesman and Vehicle Routing Problems

In the traveling salesman problem (TSP), a single salesman must visit a set of customers or cities—visiting every customer exactly once—and return home. A cost is associated with travel between two customers. Thus, the objective is to find the lowest cost tour. A tour is an ordered list of customers representing the salesman's cycle through the set of customers. For this single salesman TSP, we assume the salesman has unconstrained ability to pay the cost of the tour. Extensions to this basic problem include: multiple traveling salesmen and time windows for each customer. Lawler *et al* (1985) provides an extensive overview of the TSP.

The TSP forms the basis for the vehicle routing problem (VRP). Instead of a salesman, a vehicle must service a set of customers subject to side constraints. Servicing a customer could involve picking up or delivering a product but not both. These side constraints allow for more detailed modeling of real-world problems. For example, the side constraints can model vehicle service capacity, vehicle range, customer demands, or customer service times. Each tour must start and end at the same depot. The objective is to find a set of minimal cost tours that service all customers without violating any side constraints. Like the TSP, there are several extensions to the VRP. Carlton (1995) presents a hierarchical classification scheme for the general VRP (GVRP). His classification scheme defines the basic to the most complex variants of the TSP, VRP, and the pickup-and-delivery problem (PDP). The PDP extends the VRP by allowing a vehicle to make both pickups and deliveries along the same route.

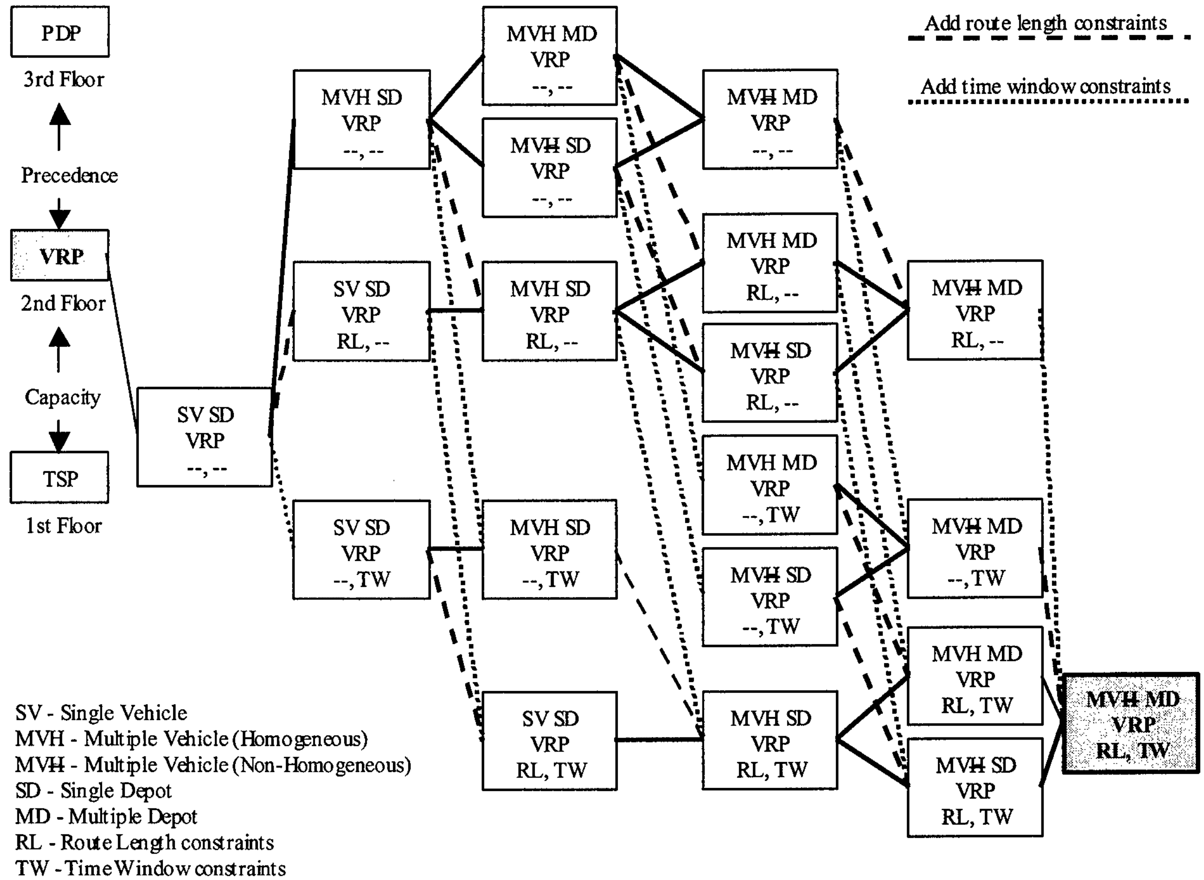

Figure 2: Hierarchical Classification Scheme (Carlton 1995) for the TSP, VRP and PDP

Using Carlton's classification scheme (See Figure 2), Harder (2000) and Kinney (2000) modeled the UAV routing problem as a MVH MD VRP with TW and RL. Additional constraints were added to handle customer priorities and restricted time windows.

2.2 Heuristics

The computational complexity of the TSP makes it very difficult to solve to optimality. The TSP falls into the NP-Hard class of problems (Parker and Rardin 1982). An NP-Hard problem has no solution algorithm whose solution time is a polynomial function of the problem size. Some solve NP-Hard problems with explicit enumeration of solutions to guarantee finding the optimal. As the number of customers in the problem increases, the number of possible solutions increases exponentially. Researchers have not yet found a polynomial-time algorithm for the TSP (Hall 1996). Laporte (1992a) surveys exact and approximate (heuristic) algorithms for the TSP. Recently, Helsgaun (2000) successfully implemented the Lin-Kernighan heuristic on the symmetric TSP. In a symmetric TSP, the travel cost from city i to city j equals the travel cost from city j to city i. Although the algorithm demonstrated success on extremely large problems with more than 1,000 cities, the strong assumptions of symmetry and single salesman limit the usefulness of this algorithm for the VRP and, especially, our UAV routing problem.

Since the VRP is an extension of the multiple TSP with side constraints, it is also difficult to solve, both from a theoretical and from a practical standpoint. In his overview of exact and approximate algorithms, Laporte (1992b) states that a variety of exact algorithms exist for the VRP, but these algorithms can only solve relatively small problems. This is evident in the following example. Hadjiconstantinou and Christofides (1995) developed an exact algorithm for a SD MVH VRP. Their algorithm computes higher-lower bounds based on an iterative combination of two problem relaxation techniques called q-paths and k-shortest paths. Hadjiconstantinou and Christofides solved this basic VRP involving up to 50 customers exactly and found tight lower bounds for problems involving up to 150 customers. Unfortunately, the algorithm falters on even a basic VRP when side constraints are added.

Heuristic algorithms have become a popular alternative to exact algorithms mainly because of their ability to handle more complex vehicle routing problems, larger size problems, and numerous side constraints. Zanakis and Evans (1981) describe heuristics as simple procedures designed to provide good but not necessarily optimal solutions to difficult problems, easily and quickly. For the UAV routing problem, the air vehicle operators require timely solutions due to their dynamic working environment. Typically, a heuristic for the TSP and VRP is categorized as either a tour construction algorithm, which involves gradually building a solution at each step, or a tour improvement algorithm, which improves upon a feasible solution (Laporte 1992a). Gendreau *et al* (1992) developed a combination tour construction and improvement algorithm called GENIUS for the TSP. The generalized insertion proceduce (GENI) builds feasible tours while the unstringing and stringing procedure (US) improves feasible tours. Gendreau *et al* (1999) adapted GENIUS for the VRP. Hachicha *et al* (2000) developed three heuristics—two constructive algorithms and one combination algorithm—for the VRP: a modified savings algorithm derived from that of Clarke and Wright (1964), a modified sweep algorithm derived from one by Gillett and Miller (1974), and a route-first/cluster-second algorithm. Chiang and Russell (1996) built a simulated annealing metaheuristic for the VRP with time windows. They use a tour construction algorithm, based on Solomon's insertion heuristic, that builds tours in parallel instead of one at a time. Solomon (1987) created an insertion-based, tour construction heuristic capable of solving the MVH VRP TW. Solomon then uses simulated annealing as a tour improvement heuristic. Laporte *et al* (2000) provide a survey of both classical and modern heuristics for the VRP; their overview of modern heuristics is devoted to tabu search because of its success with the VRP. Since the UAV routing problem is an extension of the VRP, tabu search is an appropriate technique for solving this problem. Results by O'Rourke (1999), Kinney (2000), and Harder (2000) support this assertion.

2.3 Tabu Search

Tabu search (TS) is a metaheuristic developed by Glover (1986) that intelligently searches the solution space of complex problems. A metaheuristic is an overall strategy that guides other heuristics in its search for good solutions (Glover and Laguna 1997). For TSP applications, tabu search typically employs a tour construction algorithm to build a starting solution and then attempts to improve that solution by guiding a tour improvement algorithm. Many like Semet and Taillard (1993), Gendreau *et al* (1996), Tsubakitani and Evans (1998), and Gendreau *et al* (1999) have successfully implemented tabu search for the VRP.

Tabu search systematically uses memory structures to efficiently explore the solution space through responsive exploration. Responsive exploration means aggressively investigating regions with high quality solutions, then breaking away from these local optima to explore new regions. According to Glover and Laguna (1997), the use of memory along with responsive exploration of the solution space qualifies tabu search as an intelligent heuristic. To understand how tabu search uses memory structures, the concepts of moves and tabu lists must be defined.

A move is some change in a solution attribute yielding a new solution. Solution neighbors are one move from the current solution. Subsequently, a neighborhood is the set of all neighbor solutions that can be reached using one particular move type. Harder (2000) used four move types in his tabu search for the UAV routing problem: relocate a site within a tour, relocate a site to another tour, insert a site into a dummy tour, and remove a site from a dummy tour. A dummy tour contains sites not currently visited. At each iteration, tabu search builds a neighborhood for the current solution, and then chooses the best neighbor solution to become the new current solution. Tabu search explores the solution space by executing moves and continues exploring until some stopping criteria, such as reaching a specified number of iterations, is met.

During the search, tabu search maintains a list of recently used moves called the tabu list; these moves are considered forbidden and are usually avoided. Tabu moves remain on the tabu list for a specified number of iterations, called their tabu tenure. Neighbor solutions generated by tabu moves are restricted from being selected as the new current solution for the next iteration. By remembering recently used moves with the tabu list, tabu search avoids solution repetition and becoming trapped at local optima. These tabu restrictions force the search into previously unexplored regions of the solution space. Under user-specified conditions called aspiration criteria, the tabu status of a move can be ignored. A common aspiration criterion is when a tabu move results in a neighbor solution deemed the best found thus far in the search. Tabu tenure is a crucial factor affecting the performance of the search. If the tabu list is too short, tabu search can return to the same local optimum; this cycling produces an ineffective search. If the tabu list is too long, the search becomes too diverse possibly never finding a local optima and computational time is wasted determining if a move is tabu (Tsubakitani and Evans 1998).

Two components of tabu search called intensification and diversification add to the intelligent behavior of tabu search (Glover and Laguna 1997). During intensification, the search process generates neighborhoods that favor solutions with properties occurring in good solutions. Diversification is the counterpart to intensification. During diversification, the search process generates neighborhoods that favor solutions with properties varying from solutions already encountered. By alternating between intensification and diversification phases, tabu search achieves responsive exploration of the solution space. The use of memory permits the search to intensify or diversify. Battiti and Tecchiolli (1994) intensify and diversify their tabu search by adjusting the tabu list tenure. A short tabu list results in a larger neighborhood around a solution. The search has a greater chance of performing an improving move that intensifies the search within a particular region of the solution space. Conversely, a long tabu list results in a smaller neighborhood around a solution. The search has a greater chance of performing a non-improving move that takes the search to another part of the solution space.

2.4 Advance Tabu Search Topics

2.4.1 Hashing Functions

Many basic and advanced components of tabu search rely on the ability to determine if a solution has been previously visited. Keeping a list of all previous solutions and comparing trial solutions to that list is computationally inefficient and requires large amounts of memory. With hashing functions, a solution can be represented by an integer value (Woodruff and Zemel 1993). An effective hashing function is one that is easy to solve, allows for reasonable storage and comparison, and has a low probability of collision. A collision occurs when two different solutions produce the same hash function value. Woodruff and Zemel (1993) provide several hashing functions suitable for tabu search. Carlton (1995), Kinney (2000), Harder (2000), and Nanry and Barnes (2000) utilize hashing functions in their implementations of tabu search.

2.4.2 Reactive Tabu Search (RTS)

Battiti and Tecchiolli (1994) developed the reactive tabu search. Like many heuristics, tabu search has several parameters, such as tabu tenure and neighborhood size, that affect the performance of the search. A RTS dynamically adjusts these parameters based on how the search is performing. Since RTS tunes itself to the problem at hand, the user can achieve good balance between intensification and diversification without lots of prior experience with the problem or lots of testing to determine appropriate settings for the current problem. The RTS by Battiti and Tecchiolli controlled the tabu list size to intensify or diversify the search.

Nanry and Barnes (2000) used a RTS to solve a pickup-and-delivery problem. Nanry and Barnes developed a hierarchical search methodology based on the average duration of a tour from the current solution and the average length of the time windows for the customers. This search methodology dictated which types of moves to consider and when to consider them. For example, when the average time window length is large relative to the average duration of a tour, Nanry and Barnes state that a large number of feasible solutions exist. Therefore, their search methodology encourages more tour-"polishing" or improvement moves in comparison to moves that add or remove tours.

O'Rourke (1999) used a RTS for the UAV routing problem. His tabu search adjusted the tabu list size as well as a penalty coefficient. O'Rourke used an objective function that included penalties for missed time windows, exceeding vehicle capacity, and exceeding vehicle range. Controlling the penalty coefficient forced the search in and out of feasible regions of the solution space and acted as an additional diversification strategy. Harder (2000) and Kinney (2000) used a RTS for the UAV routing problem. Their tabu search not only adjusted the tabu list size, it actively determined how many iterations to spend improving a solution.

2.4.3 Elite List and Jump Search Strategies

As tabu search explores the solution space, solutions meeting some specified criteria—such as having a good objective function value—are stored in an elite list. Once created, the tabu search can revisit solutions on the elite list and thoroughly examine the neighborhoods around those solutions for even better solutions. Essentially, tabu search undergoes a diversification phase to build the elite list and then enters an intensification phase to reexamine the elite solutions. This process can repeat itself until a stopping criterion is met.

Jump search by Tsubakitani and Evans (1998) takes the concept of elite lists a step further. Jump search uses one or more heuristics to build an elite list, which Tsubakitani and Evans refer to as jump points. Tsubakitani and Evans used six different heuristics to create their jump points. Tabu search is applied to the best jump point. Once a local optimum is found, the search starts over with the next best jump point. This process continues until all jump points have been used or after a specified number of iterations. Kinney (2000) combines jump search with his reactive tabu search for the UAV routing problem. However, Kinney uses various parameter settings in the Solomon tour construction heuristic to build up to 176 starting solutions or jump points. Jump search allows the tabu search to start with a selection of good solutions and acts as a diversification strategy.

2.4.4 Probabilistic Tabu Search (PTS)

Tabu search, as it has been discussed thus far, is deterministic in selecting moves and adjusting tabu tenure. Probabilistic tabu search departs from its deterministic predecessor by opting for probabilistic move selection and tabu tenure. Moves are created and evaluated as usual, and then these evaluations are mapped into probabilities that favor moves receiving the best evaluations. Xu *et al* (1998) and Løkketangen and Glover (1998) use probabilistic tabu search. Both papers present a probabilistic move selection technique where moves are ranked in order of their evaluation and then, starting with the best one, are either accepted with probability p or rejected with probability $1-p$. Tabu tenure is randomly selected, usually between a specified upper and lower bound. Gendreau *et al* (1999) determine tabu tenure in this fashion. Løkketangen and Glover (1998) consider the controlled randomization of probabilistic tabu search as a means for obtaining diversity without relying on memory.

2.5 Routing Around Obstacles

In practical UAV routing problems, we must build routes that avoid traversing obstacles such as no-fly zones, threat areas, and bad weather. Asseo (1998) developed an algorithm for in-flight re-planning of routes to avoid threat zones of circular shapes. Using linear segments tangent to the threat periphery and circular segments along the threat periphery, this geometric construction algorithm obtains the shortest route between a start and destination point. The algorithm works with overlapping threats of varying sizes. This algorithm was designed for in-flight use; thus it provides quick solutions. Unfortunately, this algorithm only handles circular threats. In our problem, the majority of the ROZs will be polygon in shape.

Wilber (1998) outlines an approach to strategic route planning. Wilber defines strategic route planning as the process of finding an acceptable, low cost flight path from a start point to a goal that meets all imposed strategic criteria. These strategic criteria include avoiding threats. Wilbur uses an informed best-first search, also called the A* algorithm (pronounced "A star"), to create routes. To use the A* algorithm, the search space must be organized as a tree, which is a special case of a directed graph. The A* algorithm searches the tree by using a heuristic to estimate the cost of decisions already made plus the potential inherent in the remaining decisions. Nodes with low estimates are added to the shortest path. The search continues to build a path until the goal is reached.

Surprisingly enough, computer game programmers regularly use operational research-type methods within their games to provide a higher level of artificial intelligence and increased realism. The problem of routing around obstacles is extremely common in computer video games (Stout 1997). A review of techniques used by game programmers revealed that Dijkstra's algorithm is used for finding the shortest path between two points. Figure 3 shows how this problem of routing around obstacles can be modeled as a graph. A graph is a defined as a collection of two sets: a set of points called vertices and a set of lines called edges. In graph theory, the terminology is not completely standard; a vertex is often called a node while an edge is often called an arc.

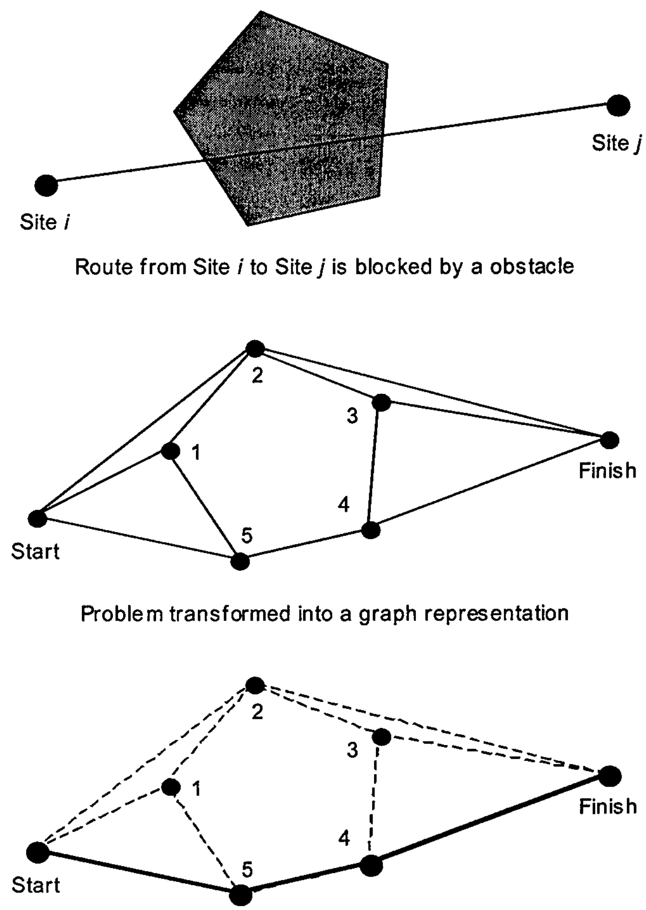

Figure 3: Transformation of obstacle avoidance problem to shortest path problem

Dijkstra's algorithm is a very simple and efficient algorithm for finding the shortest path between two specified vertices on a non-negative cost matrix graph (Christofides 1975). In general, this iterative algorithm relies on assigning temporary labels to vertices; these temporary labels represent the upper bound cost to the vertex from the start vertex. At each iteration, the algorithm reduces the labels and selects one label to become permanent, indicating that it is no longer an upper bound but the exact length of the shortest path from the start vertex to the vertex in question. The algorithm continues until the target vertex becomes permanently labeled or all vertices are permanently labeled. Due to the simplicity and efficiency of Dijkstra's algorithm, it is most appropriate for our application of building routes around obstacles in the UAV routing problem.

2.6 Conclusion

The TSP, VRP, and their variants represent more than just an academic exercise; these problems relate to everyday life and military applications. The vast amount of literature on these problems highlights this fact. The tabu search heuristic remains a widely used and effective tool for solving the TSP and VRP. Kinney (2000) and Harder (2000) demonstrated that tabu search could quickly provide high quality solutions for the UAV routing problem. The efficiency of tabu search is affected by how the algorithm uses knowledge of its progress. By using knowledge, tabu search can smartly make decisions that lead to better solutions faster. Those researchers who have used a reactive tabu search have shown the positive effect of using knowledge. Tabu search can also be efficient by avoiding the overhead of knowledge through randomly choosing among "good" decisions as those researchers employing a probabilistic tabu search have shown. Chapter 3 discusses the implementation of a reactive tabu search and probabilistic tabu search for the UAV routing problem.

Although a vast amount of literature exists on the TSP and VRP. The problem of routing around obstacles does not appear to have been combined with solving a TSP or VRP via a tabu search heuristic. Dijkstra's algorithm quickly solves the shortest-path problem between two vertices on a graph. Chapter 3 discusses how we incorporate this algorithm into the AFIT Router software to give UAV operators the ability to plan routes that avoid a general restricted operating zone.

3. METHODOLOGY

3.1 Overview of Modifications to the AFIT Router

Harder (2000) proposed a general software architecture for optimization applications to promote software reuse amongst analysts and researchers. He then applied this proposed architecture in the development of the AFIT Router, a tool for routing unmanned aerial vehicles. The AFIT Router allows the user to model the UAV routing problem and generate high quality routes that minimize the number of vehicles used and the total travel time. The AFIT Router consists of four components: the graphical-user interface (GUI), core AFIT router kernel, the universal vehicle router (UVR), and the solvers. The core AFIT router kernel tracks data for sites, vehicles, bases, and restricted operating zones. We modified the core AFIT router kernel to calculate travel times for routes that skirt around restricted operating zones. The following section describes the new data objects within the core AFIT router kernel and how they are used to calculate travel times.

Harder also provided his adaptive tabu search with the AFIT Router. Harder designed this algorithm to provide good solutions very fast (i.e. short CPU run time). We compared three tabu search algorithms as possible replacements for Harder's adaptive tabu search: probabilistic tabu search, reactive tabu search I, and reactive tabu search II. Our goal was to only replace the adaptive tabu search if one of our three algorithms either outperformed the adaptive tabu search in terms of solution quality while maintaining similar CPU running times or outperformed the adaptive tabu search in terms of CPU running times while maintaining similar solution quality.

3.2 Modifications for Routing Around ROZs

The core AFIT router kernel interacts with the user via the GUI application and any solvers via the UVR. Prior to our modifications, the core AFIT router kernel constructed a time matrix for each vehicle. These time matrices store travel times for paths to and from all sites or bases. The UVR passes these times on to any solver being employed. The core AFIT router kernel calculates these times based on the vehicle's cruising speed, the great circle distance between the two points, and the effect of wind. Great circle distance accounts for the effect of traveling over a large sphere, the earth. Previously, a restricted operating zone (ROZ) aided in assigning time windows and time walls to all sites located within the geographic region defined by the ROZ. Now, a ROZ defines a geographic region where UAV operation is prohibited or restricted.

Our modifications change how these times are calculated by accounting for any extra time required to fly around any ROZ blocking the point-to-point line. We added three additional data objects to the core AFIT router kernel to handle this task.

Table 1 lists each object and the data it tracks and/or functions it performs. The *Waypoint* object models a point (latitude and longitude coordinates) on the ground. The *Waypoint* allows this obstacle avoidance problem to be converted into a shortest path problem. The *Route* object simply models a route between two points by storing an ordered set of *Waypoints*. The *Route Builder* acts as the main component for routing around restricted operating zones. The *Route Builder* determines if a path between two *Waypoints* is blocked by a ROZ, converts the obstacle avoidance problem into a shortest path problem, and solves for the shortest path producing a *Route* object that is used to determine the estimated travel time between two points. Figure 4 depicts the logic for the Route Builder.

Table 1: New data objects for routing around obstacles

DATA OBJECT	DATA TRACKED/FUNCTIONS
Waypoint	Stores latitude, longitude, and a reference to site or ROZ from which this point originates.
Route	Stores start point, finish point, and array of *Waypoints* to represent path between start and finish.
Route Builder	Requires the following parameters: start *Waypoint*; finish *Waypoint*; and a list of restricted operating zones. Determines if straight-line path between start and finish is blocked by a ROZ. Converts obstacle avoidance problem into a graph, if necessary. Produces a *Route* object that represents the path between start and finish.

3.2.1 Detecting Intersections

The *Route Builder* has to be able to detect when paths cross a ROZ. The Java 2 programming language has built in methods for determining when two line segments intersect. Unfortunately, we cannot employ this built-in method alone. Since we build shortest-paths based on the vertices and edges of a ROZ, this built-in method detects an intersection between the line segments of the shortest-path and the ROZ. If we used the built-in method alone, the *Route Builder* could never construct line segments that touch the ROZ vertices or edges. Since we do need to build shortest-paths that touch the vertices or edges of a ROZ, we developed a technique to detect when a path or path segment crosses the interior of a ROZ.

To determine when a path or path segment crosses a ROZ, we count the number of intersections. If the path connects one site to another and if we find any intersections, then the path crosses the interior of the ROZ. If the path connects one site to a ROZ vertex and if we find more than two intersections, then the path crosses the interior of the ROZ. Finally, if the path connects one vertex from a ROZ to another vertex from a ROZ and if we find more than four intersections, then the path crosses the interior of the ROZ.

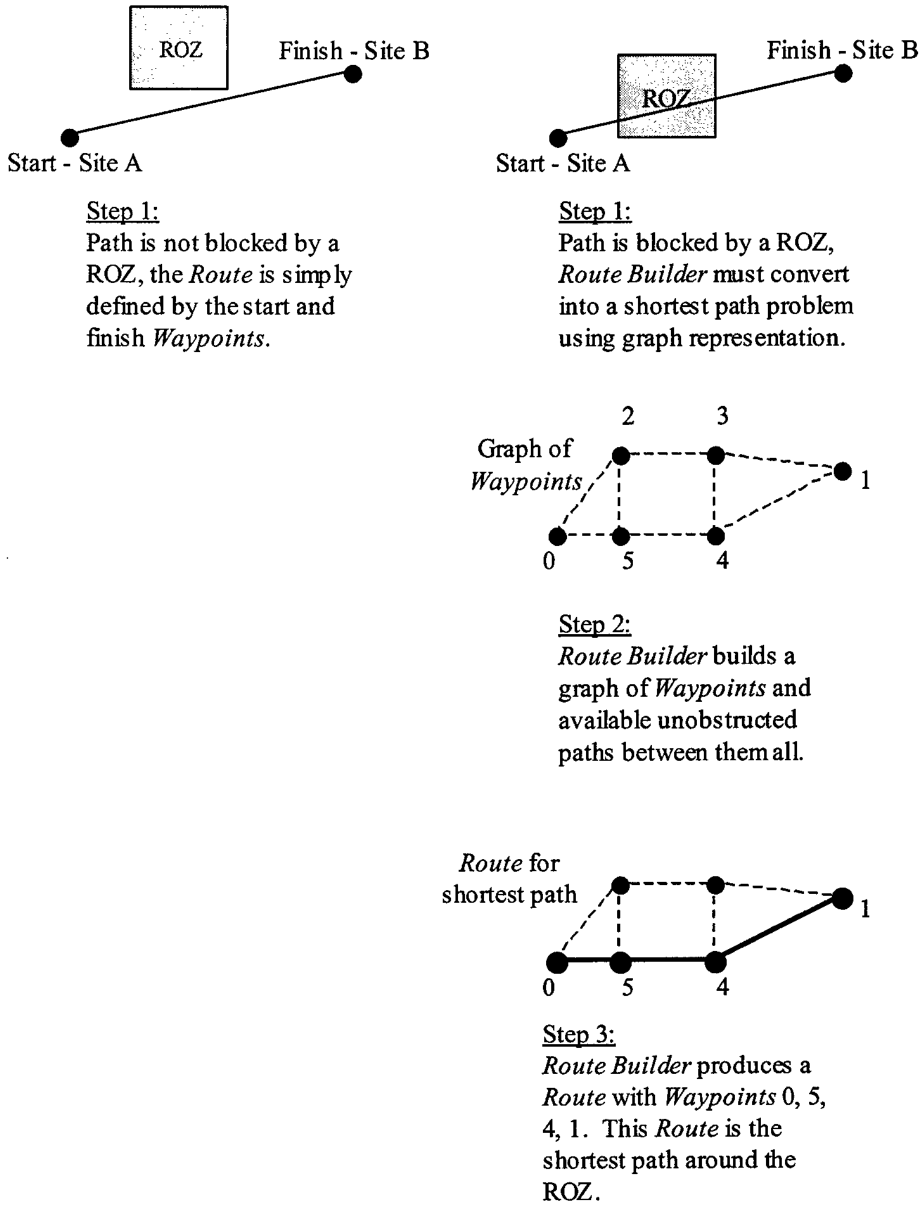

Figure 4: *Route Builder* Logic

3.3 Move Definitions

For the probabilistic tabu search and the two reactive tabu search algorithms, we use five move types: moves that relocate customers within the tour, moves that relocate customers to other tours, moves that add customers from actual tours to the dummy tour, moves that remove customers from the dummy tour and adds then to actual tours, and moves that swap a pair of customers in a tour.

When relocating customers within a tour, we consider each customer in each tour. We generate all feasible moves that do not violate any constraints (i.e. time windows, vehicle range, and matching vehicle capability with customer requirement). This move type can improve the ordering of customers within a particular tour. However, this move type will not change the solution structure too much in terms of number of tours.

When relocating customers to others tours, we consider each customer in each tour. We generate all moves that place a customer into another tour without violating any constraints. This move type can drastically change the solution structure by increasing or decreasing the total number of tours in the solution.

For each customer in each tour, we generate moves that place a customer from an actual tour into the dummy tour. The move type can drastically change the solution structure by decreasing the total number of tours in the solution.

For each customer in the dummy tour, we generate moves that place the customer into every feasible position on every tour. For feasibility, we check whether any time windows were violated or the new distance is greater than the vehicle range. Employing this move type ensures that we attempt to avoid skipping any customers.

For each tour, we generate all possible pairs to swap. This move type results in the generation of a number of moves.

3.4 Implementation of Probabilistic Tabu Search

We implement a probabilistic tabu search (PTS) that employs the following concepts: probabilistic move selection and tabu tenure. Like Kinney (2000) and Harder (2000), the search uses the Solomon tour construction heuristic to produce starting solutions. However, the PTS algorithm only uses the single best starting solution. The PTS algorithm executes 100 iterations. At each iteration, the search probabilistically selects a move. Depending on whether or not the selected move improves the current solution being explored, the search randomly decreases or increases the tabu tenure. Table 2 outlines the basic steps for PTS.

Table 2: Steps for Probabilistic Tabu Search (PTS)

STEP	ACTION
PTS 0	Initialize settings and get a single starting solution.
PTS 1	If a new best solution was found during the previous iteration, generate moves that relocate customers within a tour, relocate customers to other tours, and remove customers from the dummy tour. Otherwise, generate moves that swap two customers within the same tour and moves that insert customers into the dummy tour. Probabilistically choose (See Table 3) the next move from the candidate list of moves.
PTS 2	Add the chosen move to the tabu list.
PTS 3	If we have not performed 100 iterations, then continue on to PTS 1. Otherwise, stop searching.

3.4.1 Probabilistic Move Selection

Typically, tabu search selects the move with the best evaluation. Løkketangen and Glover (1998) suggest that purely greedy move evaluations may have a "noise level" causing them to be imperfect. Therefore, the move with the best evaluation may not be the best move. Probabilistic move selection attempts to account for this noise level by biasing move selection towards the move with the best evaluation. In addition, probabilistic move selection provides diversification without the reliance on memory. Table 3 shows how we employ probabilistic move selection for PTS.

Table 3: Probabilistic Move Selection (PMS)

STEP	ACTION
PMS 1	Get list of candidate moves; rank order the moves starting with the best one.
PMS 2	Select the first (potentially best) move from candidate list to be the current move.
PMS 3	Move acceptance: a. Accept the current move if the aspiration criteria are satisfied and stop. b. Reject the current move if it is tabu and continue to step 4. c. Generate a random number, r, between 0 and 1; if $r \le p$ (where p is between 0 and 1), accept the current move and stop. Otherwise, reject the current move and continue to PMS 4.
PMS 4	Select the next move on the candidate list to be the current move and go to PMS 3.

The probability of selecting a move is p; conversely, $1 - p$ is the probability of not selecting a move. Thus by ignoring tabu status and aspiration criteria, $(1 - p)^k$ is the probability of not selecting one of the first k moves and $1 - (1 - p)^k$ is the probability of selecting one of the first k moves. For example, if p = 0.40, the probability of selecting one of the top five moves is about 0.92 and the probability of selecting one of the top ten moves is about 0.99. This simple technique results in a search that aggressively chooses among the best moves.

We set p equal to 0.75. Løkketangen and Glover state that higher threshold probabilities should be used when one can more accurately assess the worth of a move. For the VRP, we can accurately calculate the immediate change in the objective function for a given move. However, we cannot predict how a specific change will affect the search several iterations later; this represents our "noise level".

3.4.2 Probabilistic Tabu Tenure

After a move is selected using the PMS procedure, PTS stores the move on a tabu list for a specified number of iterations called the tabu tenure (also the size of the list). The actual move is not stored; instead, a hashing function yields a hash value based on the customer being moved. We initially set the tabu tenure to 35% of the number of customers based on the empirical analysis by Kinney (2000). The minimum allowed tabu tenure is set to 20% of the number of customers and the maximum allowed tabu tenure is set to 50% of the number of customers. When a new local best solution is found during an iteration, PTS decreases the tabu tenure to intensify the search around the new best solution using Equation (1). When an un-improving move is made, PTS increases the tabu tenure to diversify the search using Equation (2). For the following equations, T_k is the tabu tenure for iteration k, $T_{\min}$ is the minimum allowed tabu tenure, $T_{\max}$ is the maximum allowed tabu tenure, and i is the current iteration while $i+1$ is the next iteration.

$$T_{i+1} = \lfloor (T_i - (T_i - T_{\min}) \cdot r) + 0.5 \rfloor \tag{1}$$

$$T_{i+1} = \lfloor (T_i + (T_{\max} - T_i) \cdot r) + 0.5 \rfloor \tag{2}$$

This tabu list behaves reactively by decreasing or increasing tabu tenure depending on the nature of the search. However, the degree to which the tabu tenure is either decreased or increased is determined randomly.

3.5 Implementation of Reactive Tabu Search

We implement two versions of a reactive tabu search. The first version, RTS-I, executes a minimum of 100 iterations and continues (in increments of 50 iterations) until 50 iterations fail to find a new best solution. Like PTS, we base tabu status on the customer being moved. The length of the tabu list is initially set to 35% of the number of customers. We adjust the tabu list according to how the search is performing. If 25 iterations pass without finding a new best solution, we then increase the tabu list length by one. We continue increasing by one until a new best solution is found or the tabu list reaches a length of 50% of the number of customers. Once a new best solution is found or the tabu list length exceeds 50% of the number customers, we reset the tabu list length back to 35% of the number of customers. Table 4 outlines the basic steps of RTS-I.

The second version, RTS-II, uses the reactive search methodology inspired by the work of Nanry and Barnes (2000). The RTS-II algorithm first calculates the average time window length using Equation (3) where n equals the number of customers, e_i is the earliest arrival time for customer i, and l_i is the latest departure time for customer i. When the average time window length is greater than 25% of the average duration of all tours in the current solution, the algorithm generates the following types of moves for a single iteration: moves that relocate customers to other tours and moves that remove customers from the dummy tour. If we fail to create any feasible moves, we generate moves that insert customers from actual tours into the dummy tour. For the next $\left\lfloor n/10 + 0.5 \right\rfloor$ iterations, the algorithm generates moves that relocate customers within a tour. If at any point we fail to create any feasible moves, then for every tour we generate all possible moves that swap two customers within a tour. When the average time window length is less than 25% of the average duration of all tours in the current solution, the algorithm again generates the following types of moves for a single iteration: moves that relocate customers to other tours and moves that remove customers from the dummy tour. If we fail to create any feasible moves, we generate moves that insert customers into the dummy tour. For the next $\left\lfloor \frac{n}{25} + 0.5 \right\rfloor$ iterations, the algorithm generates moves that relocate customers within a tour.

$$atwl = \frac{\sum_{i=1}^{n}(l_i - e_i)}{n} \tag{3}$$

When using a single starting solution, the RTS-II algorithm executes a minimum number of iterations equal to 50% of the number of customers and continues (in increments of iterations equal to 50% of the number of customers) until the number of iterations equal to 50% of the number of customers fails to find a new best solution. When using multiple starting solutions, the RTS-II algorithm, for each starting solution, executes a minimum number of iterations equal to 25% of the number of customers and continues (in increments of iterations equal to 25% of the number of customers) until the number of iterations equal to 25% of the number of customers fails to find a new best solution. This algorithm employs the same tabu list technique used in RTS-I. Table 5 outlines the basic steps of RTS-II.

Table 4: Steps for Reactive Tabu Search, Version I (RTS-I)

STEP	ACTION
RTS-I 0	Initialize settings and get next available starting solution.
RTS-I 1	If a new best solution was found during the previous iteration, generate moves that remove customers from the dummy tour. Otherwise, generate moves that relocate customers within a tour and between tours. Also, generate moves that insert customers in the dummy tour every fifth iteration.
RTS-I 2	Add the chosen move to the tabu list.
RTS-I 3	If we have not performed 50 iterations or have found a new best solution in the last 50 iterations, then continue on to RTS-I 1. Otherwise, go to RTS-I 4.
RTS-I 4	If there are starting solutions available and we have explored less than 5 starting solutions without finding a new global best solution, go to RTS-I 0. Otherwise, stop.

Table 5: Steps for Reactive Tabu Search, Version II (RTS-II)

STEP	ACTION
RTS-II 0	Initialize settings and get next available starting solution.
RTS-II 1	Generate moves as described in section 3.5 referring to RTS-II.
RTS-II 2	Add the chosen move to the tabu list.
RTS-II 3	If we have not performed the number of iterations equal to 50% of the number of customers or have found a new best solution in the last number of iterations equal to 50% of the number of customers, then continue on to RTS-II 1. Otherwise, go to RTS-II 4.
RTS-II 4	If there are starting solutions available and we have explored less than 5 starting solutions without finding a new global best solution, go to RTS-II 0. Otherwise, stop.

Both reactive tabu search algorithms either start with a single solution or a set of solutions generated by the Solomon tour construction heuristic.

3.6 Conclusion

We modified the AFIT Router to considered ROZs as no-fly zones. A ROZ is transformed into a shortest path problem using a graph representation. A shortest path algorithm then finds the best path around the ROZ.

We provide a probabilistic tabu search (PTS) that aggressively explores the solution space while maintaining a degree of diversity during move selection. We also provide two reactive tabu search heuristics, RTS-I and RTS-II. RTS-I employs a move strategy and tabu list that adjusts depending on the how the search is performing. RTS-II employs a move strategy that adjusts depending on the characteristics of the problem as well as how the search is performing. RTS-II uses a reactive tabu list as well. Chapter 4 details the empirical testing used to compare the performance of PTS, RTS-I, and RTS-II.

4. Analysis and Results

4.1 Solomon Problem Set

We use the Solomon SD MVH VRP (with TW and RL) problem set to compare different configurations of our heuristic and to test the performance of our heuristics against other published solutions. The Solomon problems were randomly generated to account for several factors such as the geographic location of customers, the vehicle capacity for servicing customer demands, the percentage of customers with time windows, and the size of time windows for customer availability.

The Solomon problem consists of 56 problems categorized into six different types: R1, C1, RC1, R2, C2, and RC2. Each problem has data for 100 customers where 25, 50, 75, or 100% of them have time windows. Each problem type has between 8 and 12 problems. In the R1 and R2 problem sets, the customers have random locations. In the C1 and C2 problem sets, the customers have clustered locations. In the RC1 and RC2 problem sets, some customers have random locations while others are clustered together. The R1, C1, RC1 problems have small time windows and a vehicle with a small capacity. The R2, C2, RC2 problems have large time windows and a vehicle with a large capacity (Solomon 2000).

4.2 Analysis of PTS, RTS-I, and RTS-II

We compared our algorithms to the adaptive tabu search provided by Harder (2000) and the best know solutions for the Solomon problem set compiled by Kinney (2000). We also compared our algorithms against each other. The distance and CPU run time values generated by the algorithms are not normally distributed. However, we assume the distance or CPU run time values generated by each algorithm have equal variances. Consequently, we used a nonparametric statistical test called the Wilcoxon signed-rank test to make statistical comparisons. We test whether the differences in distance or CPU run time for two algorithms equals zero. All testing was performed on a Pentium III 650 MHz computer with 128 MB of RAM. Equation (4) was used to calculate the percent difference. For following tables that compare two algorithms, x_1 refers to distance or CPU run time for the algorithm in the left and x_2 refers to distance or CPU run time for the algorithm in the right column of each table.

$$\% \text{ Difference} = \frac{x_2 - x_1}{x_1} \cdot 100 \tag{4}$$

Table 6 contains the distances and CPU run times for the RTS-I and RTS-II algorithms using a single starting solution. In terms of distance, RTS-II proved to outperform RTS-I with a p-value of less than 0.001. In terms of CPU run time, we failed to determine any difference. Table 7 contains the distances and CPU run times for the same algorithms using multiple starting solutions. In terms of distance, RTS-II again proved to outperform RTS-I with a p-value of less than 0.001. But in terms of CPU run time, RTS-I proved to outperform RTS-II with a p-value of less than 0.001. For the 56 Solomon problems, RTS-I had an average CPU run time of 98 seconds. RTS-II had an average CPU run time of 224 seconds—this average is more than two minutes longer than RTS-I.

Table 8, Table 9, Table 10, and Table 11 compare the RTS-I and RTS-II algorithms against Harder's adaptive tabu search algorithm for single and multiple starting solutions. All versions of our reactive tabu search on average outperform Harder's adaptive tabu search in terms of distance. Figure 5 shows the average difference from the best-known solutions in terms of distance. Both RTS-I and RTS-II produce solutions that on average were within 10% of the best-known solutions whereas Harder's adaptive tabu search only averaged within 17% of the best-known solutions.

Table 12 compares the PTS algorithm against Harder's adaptive tabu search algorithm. Although PTS proved to run faster than Harder's adaptive tabu search with a p-value of less than 0.001, this is due to the fact that the PTS algorithm only runs a 100 iterations whereas Harder's adaptive tabu search runs until there is no improvement for several iterations and restarts with new starting solutions at least three times. In terms of distance, we fail to determine any difference between the two algorithms. PTS did not perform as well as the two reactive tabu search algorithms. The probabilistic move selection technique fails to move the search into good regions of the solution space.

Table 6: Comparing RTS-I and RTS-II using Single Starting Solution

	RTS-I (Single Starting Solution)				RTS-II (Single Starting Solution)				% Difference	
Problems	Distance	Vehicles	Time (sec)	Best (sec)	Distance	Vehicles	Time (sec)	Best (sec)	Distance	Time
C101	828.94	10	73	11	828.94	10	23	12	0.00%	-68.49%
C102	911.73	10	56	15	838.73	10	40	33	-8.01%	-28.57%
C103	946.31	10	82	17	868.03	10	57	46	-8.27%	-30.49%
C104	1034.77	10	56	19	1034.77	10	39	29	0.00%	-30.36%
C105	828.94	10	55	9	828.94	10	24	12	0.00%	-56.36%
C106	828.94	10	56	10	828.94	10	23	14	0.00%	-58.93%
C107	828.94	10	83	11	828.94	10	24	15	0.00%	-71.08%
C108	828.94	10	62	17	828.94	10	23	16	0.00%	-62.90%
C109	870.91	10	58	18	870.91	10	37	26	0.00%	-36.21%
C1 Average	**878.71**		**65**	**14**	**861.90**		**32**	**23**	**-1.91%**	**-50.09%**
C201	591.56	3	48	14	591.56	3	32	14	0.00%	-33.33%
C202	591.56	3	61	23	591.56	3	41	30	0.00%	-32.79%
C203	712.46	3	79	30	707.96	3	59	45	-0.63%	-25.32%
C204	680.78	3	69	40	679.08	3	67	50	-0.25%	-2.90%
C205	588.88	3	55	18	588.88	3	39	20	0.00%	-29.09%
C206	588.49	3	55	20	588.49	3	42	21	0.00%	-23.64%
C207	599.49	3	56	19	588.29	3	43	19	-1.87%	-23.21%
C208	588.32	3	85	30	588.32	3	51	28	0.00%	-40.00%
C2 Average	**617.69**		**64**	**24**	**615.52**		**47**	**28**	**-0.35%**	**-26.38%**
R101	1749.36	20	118	27	1728.24	20	66	54	-1.21%	-44.07%
R102	1637.60	18	80	38	1627.27	18	80	73	-0.63%	0.00%
R103	1366.11	15	109	34	1361.18	15	75	66	-0.36%	-31.19%
R104	1125.57	11	75	29	1125.87	11	65	51	0.03%	-13.33%
R105	1505.58	15	80	25	1500.92	15	57	46	-0.31%	-28.75%
R106	1427.10	13	76	23	1396.03	13	68	54	-2.18%	-10.53%
R107	1179.58	12	112	25	1178.44	12	54	43	-0.10%	-51.79%
R108	1135.53	11	76	23	1045.54	11	108	97	-7.92%	42.11%
R109	1298.83	13	76	31	1298.83	13	69	58	0.00%	-9.21%
R110	1261.33	12	79	35	1243.66	12	69	56	-1.40%	-12.66%
R111	1251.27	12	75	31	1198.26	12	186	170	-4.24%	148.00%
R112	1059.87	11	74	27	1057.86	11	165	125	-0.19%	122.97%
R1 Average	**1333.15**		**86**	**29**	**1313.51**		**89**	**74**	**-1.47%**	**3.11%**
R201	1559.44	4	69	25	1399.01	4	111	96	-10.29%	60.87%
R202	1245.92	4	62	37	1300.40	4	120	103	4.37%	93.55%
R203	1105.50	3	91	57	1086.25	3	150	132	-1.74%	64.84%
R204	877.20	3	105	79	855.55	3	148	133	-2.47%	40.95%
R205	1421.91	3	64	31	1229.54	3	85	71	-13.53%	32.81%
R206	1036.86	3	70	46	1014.42	3	85	67	-2.16%	21.43%
R207	994.28	3	107	59	983.90	3	119	102	-1.04%	11.21%
R208	782.40	3	116	91	775.72	3	127	112	-0.85%	9.48%
R209	1055.44	3	71	46	1040.22	3	103	91	-1.44%	45.07%
R210	1209.17	3	75	43	1152.45	3	99	79	-4.69%	32.00%
R211	930.30	3	100	68	919.85	3	166	138	-1.12%	66.00%
R2 Average	**1110.77**		**85**	**53**	**1068.85**		**119**	**102**	**-3.77%**	**41.18%**
RC101	1831.99	16	110	16	1830.15	16	41	33	-0.10%	-62.73%
RC102	1562.99	14	74	24	1545.89	14	64	50	-1.09%	-13.51%
RC103	1427.47	12	102	37	1408.01	12	73	65	-1.36%	-28.43%
RC104	1427.47	12	76	39	1408.01	12	72	64	-1.36%	-5.26%
RC105	1764.79	15	87	30	1705.93	15	56	45	-3.34%	-35.63%
RC106	1450.37	13	74	22	1443.56	13	47	37	-0.47%	-36.49%
RC107	1383.18	12	110	28	1348.14	12	59	48	-2.53%	-46.36%
RC108	1213.45	11	80	18	1211.75	11	55	39	-0.14%	-31.25%
RC1 Average	**1507.71**		**89**	**27**	**1487.68**		**58**	**48**	**-1.33%**	**-34.50%**
RC201	1742.40	4	68	22	1619.74	4	62	46	-7.04%	-8.82%
RC202	1550.54	4	59	36	1510.68	4	105	92	-2.57%	77.97%
RC203	1319.78	4	91	44	1246.30	4	137	122	-5.57%	50.55%
RC204	988.74	3	89	60	977.33	3	130	109	-1.15%	46.07%
RC205	1680.31	4	58	31	1565.42	4	135	118	-6.84%	132.76%
RC206	1432.17	4	60	28	1375.99	4	111	96	-3.92%	85.00%
RC207	1196.16	4	84	39	1143.49	4	101	87	-4.40%	20.24%
RC208	1004.26	3	90	60	963.45	3	94	76	-4.06%	4.44%
RC2 Average	**1364.29**		**75**	**40**	**1300.30**		**109**	**93**	**-4.69%**	**46.08%**

Table 7: Comparing RTS-I and RTS-II using Multiple Starting Solutions

	RTS-I (Multiple Starting Solutions)				RTS-II (Multiple Starting Solutions)				% Difference	
Problems	Distance	Vehicles	Time (sec)	Best (sec)	Distance	Vehicles	Time (sec)	Best (sec)	Distance	Time
C101	828.94	10	72	11	828.94	10	74	12	0.00%	2.78%
C102	911.73	10	73	15	838.73	10	124	70	-8.01%	69.86%
C103	946.31	10	75	17	915.76	10	144	34	-3.23%	92.00%
C104	870.17	10	98	50	901.46	10	216	77	3.60%	120.41%
C105	828.94	10	65	9	828.94	10	40	11	0.00%	-38.46%
C106	828.94	10	69	11	828.94	10	53	13	0.00%	-23.19%
C107	828.94	10	69	11	828.94	10	61	15	0.00%	-11.59%
C108	828.94	10	80	17	828.94	10	65	16	0.00%	-18.75%
C109	828.94	10	84	40	828.94	10	131	75	0.00%	55.95%
C1 Average	**855.76**		**76**	**20**	**847.73**		**101**	**36**	**-0.94%**	**32.55%**
C201	591.56	3	64	14	591.56	3	55	23	0.00%	-14.06%
C202	591.56	3	69	23	591.56	3	133	30	0.00%	92.75%
C203	712.46	3	79	31	707.96	3	170	45	-0.63%	115.19%
C204	680.78	3	87	41	679.08	3	288	49	-0.25%	231.03%
C205	588.88	3	64	18	588.88	3	73	19	0.00%	14.06%
C206	588.49	3	67	20	588.49	3	140	21	0.00%	108.96%
C207	588.29	3	72	31	588.29	3	83	19	0.00%	15.28%
C208	588.32	3	81	30	588.32	3	81	27	0.00%	0.00%
C2 Average	**616.29**		**73**	**26**	**615.52**		**128**	**29**	**-0.13%**	**75.47%**
R101	1749.36	20	115	28	1708.24	20	230	133	-2.35%	100.00%
R102	1637.60	18	119	38	1637.89	18	276	60	0.02%	131.93%
R103	1366.11	15	117	34	1325.13	15	305	238	-3.00%	160.68%
R104	1125.57	11	114	29	1125.87	11	249	48	0.03%	118.42%
R105	1505.58	15	109	26	1505.58	15	182	37	0.00%	66.97%
R106	1427.10	13	106	24	1336.98	13	260	255	-6.32%	145.28%
R107	1179.58	12	110	25	1159.98	12	210	118	-1.66%	90.91%
R108	1082.39	11	122	63	1021.92	10	337	229	-5.59%	176.23%
R109	1298.83	13	117	32	1298.83	13	197	61	0.00%	68.38%
R110	1261.33	12	123	36	1216.80	12	329	223	-3.53%	167.48%
R111	1251.27	12	113	30	1211.01	12	613	289	-3.22%	442.48%
R112	1059.87	11	103	28	1057.86	11	490	144	-0.19%	375.73%
R1 Average	**1328.72**		**114**	**33**	**1300.51**		**307**	**153**	**-2.12%**	**168.86%**
R201	1500.49	4	78	44	1411.39	4	242	79	-5.94%	210.26%
R202	1245.92	4	87	37	1208.29	4	279	220	-3.02%	220.69%
R203	1105.50	3	99	56	1086.25	3	408	129	-1.74%	312.12%
R204	877.20	3	123	76	857.13	3	315	150	-2.29%	156.10%
R205	1264.69	3	87	50	1202.58	3	218	210	-4.91%	150.57%
R206	1036.86	3	92	45	1014.42	3	290	68	-2.16%	215.22%
R207	994.28	3	105	57	940.19	3	219	210	-5.44%	108.57%
R208	782.40	3	138	88	775.72	3	286	105	-0.85%	107.25%
R209	1055.44	3	95	47	1040.22	3	291	92	-1.44%	206.32%
R210	1209.17	3	99	43	1112.49	3	312	249	-8.00%	215.15%
R211	930.30	3	125	67	873.81	3	260	253	-6.07%	108.00%
R2 Average	**1091.11**		**103**	**55**	**1047.50**		**284**	**160**	**-4.00%**	**176.60%**
RC101	1716.48	16	117	49	1715.67	16	140	47	-0.05%	19.66%
RC102	1562.99	14	105	25	1560.50	14	168	63	-0.16%	60.00%
RC103	1427.47	12	123	39	1390.95	12	237	135	-2.56%	92.68%
RC104	1427.47	12	123	39	1390.95	12	232	133	-2.56%	88.62%
RC105	1655.17	15	139	106	1703.65	15	204	83	2.93%	46.76%
RC106	1450.37	13	106	23	1443.56	13	131	38	-0.47%	23.58%
RC107	1383.18	12	118	29	1348.14	12	170	46	-2.53%	44.07%
RC108	1169.87	11	120	54	1169.13	11	144	59	-0.06%	20.00%
RC1 Average	**1474.12**		**119**	**46**	**1465.32**		**178**	**76**	**-0.60%**	**49.95%**
RC201	1655.49	4	71	39	1619.74	4	141	45	-2.16%	98.59%
RC202	1550.54	4	85	36	1396.56	4	313	307	-9.93%	268.24%
RC203	1148.98	4	113	76	1246.30	4	387	120	8.47%	242.48%
RC204	988.74	3	110	64	946.95	3	410	399	-4.23%	272.73%
RC205	1680.31	4	82	32	1551.50	4	301	237	-7.67%	267.07%
RC206	1339.68	4	90	48	1314.68	4	252	243	-1.87%	180.00%
RC207	1196.16	4	89	40	1085.64	4	307	299	-9.24%	244.94%
RC208	950.14	3	120	91	934.35	3	297	127	-1.66%	147.50%
RC2 Average	**1313.75**		**95**	**53**	**1261.96**		**301**	**222**	**-3.94%**	**216.84%**

Table 8: Comparing Adaptive TS and RTS-I (Single Starting Solution)

	Adaptive Tabu Search (Harder)			RTS-I (Single Starting Solution)			% Difference	
Problems	Distance	Vehicles	Time	Distance	Vehicles	Time (sec)	Distance	Time
C101	852.00	10	69	828.94	10	73	-2.71%	5.80%
C102	960.00	10	18	911.73	10	56	-5.03%	211.11%
C103	923.00	10	188	946.31	10	82	2.53%	-56.38%
C104	913.00	10	263	1034.77	10	56	13.34%	-78.71%
C105	860.00	10	80	828.94	10	55	-3.61%	-31.25%
C106	877.00	10	141	828.94	10	56	-5.48%	-60.28%
C107	894.00	10	113	828.94	10	83	-7.28%	-26.55%
C108	853.00	10	151	828.94	10	62	-2.82%	-58.94%
C109	854.00	10	240	870.91	10	58	1.98%	-75.83%
C1 Average	**887.33**		**140**	**878.71**		**65**	**-0.97%**	**-54.00%**
C201	591.00	3	83	591.56	3	48	0.09%	-42.17%
C202	676.00	3	179	591.56	3	61	-12.49%	-65.92%
C203	683.00	3	204	712.46	3	79	4.31%	-61.27%
C204	656.00	3	259	680.78	3	69	3.78%	-73.36%
C205	588.00	3	141	588.88	3	55	0.15%	-60.99%
C206	633.00	3	172	588.49	3	55	-7.03%	-68.02%
C207	601.00	3	159	599.49	3	56	-0.25%	-64.78%
C208	629.00	3	163	588.32	3	85	-6.47%	-47.85%
C2 Average	**632.13**		**170**	**617.69**		**64**	**-2.28%**	**-62.65%**
R101	1805.00	20	207	1749.36	20	118	-3.08%	-43.00%
R102	1661.00	19	251	1637.60	18	80	-1.41%	-68.13%
R103	1587.00	14	272	1366.11	15	109	-13.92%	-59.93%
R104	1156.00	11	243	1125.57	11	75	-2.63%	-69.14%
R105	1517.00	14	228	1505.58	15	80	-0.75%	-64.91%
R106	1344.00	13	213	1427.10	13	76	6.18%	-64.32%
R107	1247.00	12	228	1179.58	12	112	-5.41%	-50.88%
R108	1112.00	10	245	1135.53	11	76	2.12%	-68.98%
R109	1334.00	13	251	1298.83	13	76	-2.64%	-69.72%
R110	1248.00	12	248	1261.33	12	79	1.07%	-68.15%
R111	1242.00	11	223	1251.27	12	75	0.75%	-66.37%
R112	1148.00	10	232	1059.87	11	74	-7.68%	-68.10%
R1 Average	**1366.75**		**237**	**1333.15**		**86**	**-2.46%**	**-63.75%**
R201	1544.00	4	197	1559.44	4	69	1.00%	-64.97%
R202	1378.00	4	254	1245.92	4	62	-9.58%	-75.59%
R203	1210.00	3	268	1105.50	3	91	-8.64%	-66.04%
R204	946.00	3	372	877.20	3	105	-7.27%	-71.77%
R205	1208.00	3	234	1421.91	3	64	17.71%	-72.65%
R206	1094.00	3	279	1036.86	3	70	-5.22%	-74.91%
R207	1078.00	3	326	994.28	3	107	-7.77%	-67.18%
R208	989.00	2	407	782.40	3	116	-20.89%	-71.50%
R209	1157.00	3	293	1055.44	3	71	-8.78%	-75.77%
R210	1232.00	3	258	1209.17	3	75	-1.85%	-70.93%
R211	980.00	3	364	930.30	3	100	-5.07%	-72.53%
R2 Average	**1165.09**		**296**	**1110.77**		**85**	**-4.66%**	**-71.40%**
RC101	1802.00	16	223	1831.99	16	110	1.66%	-50.67%
RC102	1698.00	14	269	1562.99	14	74	-7.95%	-72.49%
RC103	1502.00	13	322	1427.47	12	102	-4.96%	-68.32%
RC104	1502.00	13	327	1427.47	12	76	-4.96%	-76.76%
RC105	1706.00	16	285	1764.79	15	87	3.45%	-69.47%
RC106	1478.00	13	355	1450.37	13	74	-1.87%	-79.15%
RC107	1434.00	12	286	1383.18	12	110	-3.54%	-61.54%
RC108	1228.00	11	261	1213.45	11	80	-1.18%	-69.35%
RC1 Average	**1543.75**		**291**	**1507.71**		**89**	**-2.33%**	**-69.37%**
RC201	1810.00	4	176	1742.40	4	68	-3.73%	-61.36%
RC202	1542.00	4	312	1550.54	4	59	0.55%	-81.09%
RC203	1484.00	3	258	1319.78	4	91	-11.07%	-64.73%
RC204	1113.00	3	336	988.74	3	89	-11.16%	-73.51%
RC205	1758.00	4	228	1680.31	4	58	-4.42%	-74.56%
RC206	1421.00	4	214	1432.17	4	60	0.79%	-71.96%
RC207	1362.00	4	239	1196.16	4	84	-12.18%	-64.85%
RC208	1099.00	3	356	1004.26	3	90	-8.62%	-74.72%
RC2 Average	**1448.63**		**265**	**1364.29**		**75**	**-5.82%**	**-71.73%**

Table 9: Comparing Adaptive TS and RTS-II (Single Starting Solutions)

	Adaptive Tabu Search (Harder)			RTS-II (Single Starting Solution)			% Difference	
Problems	**Distance**	**Vehicles**	**Time**	**Distance**	**Vehicles**	**Time (sec)**	**Distance**	**Time**
C101	852.00	10	69	828.94	10	23	-2.71%	-66.67%
C102	960.00	10	18	838.73	10	40	-12.63%	122.22%
C103	923.00	10	188	868.03	10	57	-5.96%	-69.68%
C104	913.00	10	263	1034.77	10	39	13.34%	-85.17%
C105	860.00	10	80	828.94	10	24	-3.61%	-70.00%
C106	877.00	10	141	828.94	10	23	-5.48%	-83.69%
C107	894.00	10	113	828.94	10	24	-7.28%	-78.76%
C108	853.00	10	151	828.94	10	23	-2.82%	-84.77%
C109	854.00	10	240	870.91	10	37	1.98%	-84.58%
C1 Average	**887.33**		**140**	**861.90**		**32**	**-2.87%**	**-77.04%**
C201	591.00	3	83	591.56	3	32	0.09%	-61.45%
C202	676.00	3	179	591.56	3	41	-12.49%	-77.09%
C203	683.00	3	204	707.96	3	59	3.65%	-71.08%
C204	656.00	3	259	679.08	3	67	3.52%	-74.13%
C205	588.00	3	141	588.88	3	39	0.15%	-72.34%
C206	633.00	3	172	588.49	3	42	-7.03%	-75.58%
C207	601.00	3	159	588.29	3	43	-2.12%	-72.96%
C208	629.00	3	163	588.32	3	51	-6.47%	-68.71%
C2 Average	**632.13**		**170**	**615.52**		**47**	**-2.63%**	**-72.50%**
R101	1805.00	20	207	1728.24	20	66	-4.25%	-68.12%
R102	1661.00	19	251	1627.27	18	80	-2.03%	-68.13%
R103	1587.00	14	272	1361.18	15	75	-14.23%	-72.43%
R104	1156.00	11	243	1125.87	11	65	-2.61%	-73.25%
R105	1517.00	14	228	1500.92	15	57	-1.06%	-75.00%
R106	1344.00	13	213	1396.03	13	68	3.87%	-68.08%
R107	1247.00	12	228	1178.44	12	54	-5.50%	-76.32%
R108	1112.00	10	245	1045.54	11	108	-5.98%	-55.92%
R109	1334.00	13	251	1298.83	13	69	-2.64%	-72.51%
R110	1248.00	12	248	1243.66	12	69	-0.35%	-72.18%
R111	1242.00	11	223	1198.26	12	186	-3.52%	-16.59%
R112	1148.00	10	232	1057.86	11	165	-7.85%	-28.88%
R1 Average	**1366.75**		**237**	**1313.51**		**89**	**-3.90%**	**-62.62%**
R201	1544.00	4	197	1399.01	4	111	-9.39%	-43.65%
R202	1378.00	4	254	1300.40	4	120	-5.63%	-52.76%
R203	1210.00	3	268	1086.25	3	150	-10.23%	-44.03%
R204	946.00	3	372	855.55	3	148	-9.56%	-60.22%
R205	1208.00	3	234	1229.54	3	85	1.78%	-63.68%
R206	1094.00	3	279	1014.42	3	85	-7.27%	-69.53%
R207	1078.00	3	326	983.90	3	119	-8.73%	-63.50%
R208	989.00	2	407	775.72	3	127	-21.57%	-68.80%
R209	1157.00	3	293	1040.22	3	103	-10.09%	-64.85%
R210	1232.00	3	258	1152.45	3	99	-6.46%	-61.63%
R211	980.00	3	364	919.85	3	166	-6.14%	-54.40%
R2 Average	**1165.09**		**296**	**1068.85**		**119**	**-8.26%**	**-59.62%**
RC101	1802.00	16	223	1830.15	16	41	1.56%	-81.61%
RC102	1698.00	14	269	1545.89	14	64	-8.96%	-76.21%
RC103	1502.00	13	322	1408.01	12	73	-6.26%	-77.33%
RC104	1502.00	13	327	1408.01	12	72	-6.26%	-77.98%
RC105	1706.00	16	285	1705.93	15	56	0.00%	-80.35%
RC106	1478.00	13	355	1443.56	13	47	-2.33%	-86.76%
RC107	1434.00	12	286	1348.14	12	59	-5.99%	-79.37%
RC108	1228.00	11	261	1211.75	11	55	-1.32%	-78.93%
RC1 Average	**1543.75**		**291**	**1487.68**		**58**	**-3.63%**	**-79.94%**
RC201	1810.00	4	176	1619.74	4	62	-10.51%	-64.77%
RC202	1542.00	4	312	1510.68	4	105	-2.03%	-66.35%
RC203	1484.00	3	258	1246.30	4	137	-16.02%	-46.90%
RC204	1113.00	3	336	977.33	3	130	-12.19%	-61.31%
RC205	1758.00	4	228	1565.42	4	135	-10.95%	-40.79%
RC206	1421.00	4	214	1375.99	4	111	-3.17%	-48.13%
RC207	1362.00	4	239	1143.49	4	101	-16.04%	-57.74%
RC208	1099.00	3	356	963.45	3	94	-12.33%	-73.60%
RC2 Average	**1448.63**		**265**	**1300.30**		**109**	**-10.24%**	**-58.71%**

Table 10: Comparing Adaptive TS and RTS-I (Multiple Starting Solutions)

	Adaptive Tabu Search (Harder)			RTS-I (Multiple Starting Solutions)			% Difference	
Problems	**Distance**	**Vehicles**	**Time**	**Distance**	**Vehicles**	**Time (sec)**	**Distance**	**Time**
C101	852.00	10	69	828.94	10	72	-2.71%	4.35%
C102	960.00	10	18	911.73	10	73	-5.03%	305.56%
C103	923.00	10	188	946.31	10	75	2.53%	-60.11%
C104	913.00	10	263	870.17	10	98	-4.69%	-62.74%
C105	860.00	10	80	828.94	10	65	-3.61%	-18.75%
C106	877.00	10	141	828.94	10	69	-5.48%	-51.06%
C107	894.00	10	113	828.94	10	69	-7.28%	-38.94%
C108	853.00	10	151	828.94	10	80	-2.82%	-47.02%
C109	854.00	10	240	828.94	10	84	-2.93%	-65.00%
C1 Average	**887.33**		**140**	**855.76**		**76**	**-3.56%**	**-45.76%**
C201	591.00	3	83	591.56	3	64	0.09%	-22.89%
C202	676.00	3	179	591.56	3	69	-12.49%	-61.45%
C203	683.00	3	204	712.46	3	79	4.31%	-61.27%
C204	656.00	3	259	680.78	3	87	3.78%	-66.41%
C205	588.00	3	141	588.88	3	64	0.15%	-54.61%
C206	633.00	3	172	588.49	3	67	-7.03%	-61.05%
C207	601.00	3	159	588.29	3	72	-2.12%	-54.72%
C208	629.00	3	163	588.32	3	81	-6.47%	-50.31%
C2 Average	**632.13**		**170**	**616.29**		**73**	**-2.50%**	**-57.13%**
R101	1805.00	20	207	1749.36	20	115	-3.08%	-44.44%
R102	1661.00	19	251	1637.60	18	119	-1.41%	-52.59%
R103	1587.00	14	272	1366.11	15	117	-13.92%	-56.99%
R104	1156.00	11	243	1125.57	11	114	-2.63%	-53.09%
R105	1517.00	14	228	1505.58	15	109	-0.75%	-52.19%
R106	1344.00	13	213	1427.10	13	106	6.18%	-50.23%
R107	1247.00	12	228	1179.58	12	110	-5.41%	-51.75%
R108	1112.00	10	245	1082.39	11	122	-2.66%	-50.20%
R109	1334.00	13	251	1298.83	13	117	-2.64%	-53.39%
R110	1248.00	12	248	1261.33	12	123	1.07%	-50.40%
R111	1242.00	11	223	1251.27	12	113	0.75%	-49.33%
R112	1148.00	10	232	1059.87	11	103	-7.68%	-55.60%
R1 Average	**1366.75**		**237**	**1328.72**		**114**	**-2.78%**	**-51.85%**
R201	1544.00	4	197	1500.49	4	78	-2.82%	-60.41%
R202	1378.00	4	254	1245.92	4	87	-9.58%	-65.75%
R203	1210.00	3	268	1105.50	3	99	-8.64%	-63.06%
R204	946.00	3	372	877.20	3	123	-7.27%	-66.94%
R205	1208.00	3	234	1264.69	3	87	4.69%	-62.82%
R206	1094.00	3	279	1036.86	3	92	-5.22%	-67.03%
R207	1078.00	3	326	994.28	3	105	-7.77%	-67.79%
R208	989.00	2	407	782.40	3	138	-20.89%	-66.09%
R209	1157.00	3	293	1055.44	3	95	-8.78%	-67.58%
R210	1232.00	3	258	1209.17	3	99	-1.85%	-61.63%
R211	980.00	3	364	930.30	3	125	-5.07%	-65.66%
R2 Average	**1165.09**		**296**	**1091.11**		**103**	**-6.35%**	**-65.31%**
RC101	1802.00	16	223	1716.48	16	117	-4.75%	-47.53%
RC102	1698.00	14	269	1562.99	14	105	-7.95%	-60.97%
RC103	1502.00	13	322	1427.47	12	123	-4.96%	-61.80%
RC104	1502.00	13	327	1427.47	12	123	-4.96%	-62.39%
RC105	1706.00	16	285	1655.17	15	139	-2.98%	-51.23%
RC106	1478.00	13	355	1450.37	13	106	-1.87%	-70.14%
RC107	1434.00	12	286	1383.18	12	118	-3.54%	-58.74%
RC108	1228.00	11	261	1169.87	11	120	-4.73%	-54.02%
RC1 Average	**1543.75**		**291**	**1474.12**		**119**	**-4.51%**	**-59.15%**
RC201	1810.00	4	176	1655.49	4	71	-8.54%	-59.66%
RC202	1542.00	4	312	1550.54	4	85	0.55%	-72.76%
RC203	1484.00	3	258	1148.98	4	113	-22.58%	-56.20%
RC204	1113.00	3	336	988.74	3	110	-11.16%	-67.26%
RC205	1758.00	4	228	1680.31	4	82	-4.42%	-64.04%
RC206	1421.00	4	214	1339.68	4	90	-5.72%	-57.94%
RC207	1362.00	4	239	1196.16	4	89	-12.18%	-62.76%
RC208	1099.00	3	356	950.14	3	120	-13.54%	-66.29%
RC2 Average	**1448.63**		**265**	**1313.75**		**95**	**-9.31%**	**-64.13%**

Table 11: Comparing Adaptive TS and RTS-II (Multiple Starting Solutions)

	Adaptive Tabu Search (Harder)			RTS-II (Multiple Starting Solutions)			% Difference	
Problems	Distance	Vehicles	Time	Distance	Vehicles	Time (sec)	Distance	Time
C101	852.00	10	69	828.94	10	74	-2.71%	7.25%
C102	960.00	10	18	838.73	10	124	-12.63%	588.89%
C103	923.00	10	188	915.76	10	144	-0.78%	-23.40%
C104	913.00	10	263	901.46	10	216	-1.26%	-17.87%
C105	860.00	10	80	828.94	10	40	-3.61%	-50.00%
C106	877.00	10	141	828.94	10	53	-5.48%	-62.41%
C107	894.00	10	113	828.94	10	61	-7.28%	-46.02%
C108	853.00	10	151	828.94	10	65	-2.82%	-56.95%
C109	854.00	10	240	828.94	10	131	-2.93%	-45.42%
C1 Average	**887.33**		**140**	**847.73**		**101**	**-4.46%**	**-28.11%**
C201	591.00	3	83	591.56	3	55	0.09%	-33.73%
C202	676.00	3	179	591.56	3	133	-12.49%	-25.70%
C203	683.00	3	204	707.96	3	170	3.65%	-16.67%
C204	656.00	3	259	679.08	3	288	3.52%	11.20%
C205	588.00	3	141	588.88	3	73	0.15%	-48.23%
C206	633.00	3	172	588.49	3	140	-7.03%	-18.60%
C207	601.00	3	159	588.29	3	83	-2.12%	-47.80%
C208	629.00	3	163	588.32	3	81	-6.47%	-50.31%
C2 Average	**632.13**		**170**	**615.52**		**128**	**-2.63%**	**-24.78%**
R101	1805.00	20	207	1708.24	20	230	-5.36%	11.11%
R102	1661.00	19	251	1637.89	18	276	-1.39%	9.96%
R103	1587.00	14	272	1325.13	15	305	-16.50%	12.13%
R104	1156.00	11	243	1125.87	11	249	-2.61%	2.47%
R105	1517.00	14	228	1505.58	15	182	-0.75%	-20.18%
R106	1344.00	13	213	1336.98	13	260	-0.52%	22.07%
R107	1247.00	12	228	1159.98	12	210	-6.98%	-7.89%
R108	1112.00	10	245	1021.92	10	337	-8.10%	37.55%
R109	1334.00	13	251	1298.83	13	197	-2.64%	-21.51%
R110	1248.00	12	248	1216.80	12	329	-2.50%	32.66%
R111	1242.00	11	223	1211.01	12	613	-2.50%	174.89%
R112	1148.00	10	232	1057.86	11	490	-7.85%	111.21%
R1 Average	**1366.75**		**237**	**1300.51**		**307**	**-4.85%**	**29.46%**
R201	1544.00	4	197	1411.39	4	242	-8.59%	22.84%
R202	1378.00	4	254	1208.29	4	279	-12.32%	9.84%
R203	1210.00	3	268	1086.25	3	408	-10.23%	52.24%
R204	946.00	3	372	857.13	3	315	-9.39%	-15.32%
R205	1208.00	3	234	1202.58	3	218	-0.45%	-6.84%
R206	1094.00	3	279	1014.42	3	290	-7.27%	3.94%
R207	1078.00	3	326	940.19	3	219	-12.78%	-32.82%
R208	989.00	2	407	775.72	3	286	-21.57%	-29.73%
R209	1157.00	3	293	1040.22	3	291	-10.09%	-0.68%
R210	1232.00	3	258	1112.49	3	312	-9.70%	20.93%
R211	980.00	3	364	873.81	3	260	-10.84%	-28.57%
R2 Average	**1165.09**		**296**	**1047.50**		**284**	**-10.09%**	**-4.06%**
RC101	1802.00	16	223	1715.67	16	140	-4.79%	-37.22%
RC102	1698.00	14	269	1560.50	14	168	-8.10%	-37.55%
RC103	1502.00	13	322	1390.95	12	237	-7.39%	-26.40%
RC104	1502.00	13	327	1390.95	12	232	-7.39%	-29.05%
RC105	1706.00	16	285	1703.65	15	204	-0.14%	-28.42%
RC106	1478.00	13	355	1443.56	13	131	-2.33%	-63.10%
RC107	1434.00	12	286	1348.14	12	170	-5.99%	-40.56%
RC108	1228.00	11	261	1169.13	11	144	-4.79%	-44.83%
RC1 Average	**1543.75**		**291**	**1465.32**		**178**	**-5.08%**	**-38.75%**
RC201	1810.00	4	176	1619.74	4	141	-10.51%	-19.89%
RC202	1542.00	4	312	1396.56	4	313	-9.43%	0.32%
RC203	1484.00	3	258	1246.30	4	387	-16.02%	50.00%
RC204	1113.00	3	336	946.95	3	410	-14.92%	22.02%
RC205	1758.00	4	228	1551.50	4	301	-11.75%	32.02%
RC206	1421.00	4	214	1314.68	4	252	-7.48%	17.76%
RC207	1362.00	4	239	1085.64	4	307	-20.29%	28.45%
RC208	1099.00	3	356	934.35	3	297	-14.98%	-16.57%
RC2 Average	**1448.63**		**265**	**1261.96**		**301**	**-12.89%**	**13.64%**

Table 12: Comparing Adaptive TS and PTS

Problem	Adaptive Tabu Search (Harder) Distance	Vehicles	CPU (sec)	PTS (averaged over 3 runs) Distance	Vehicles	CPU (sec)	% Difference Distance
C101	852.00	10	69	878.36	10	25	3.09%
C102	960.00	10	18	1084.71	10	58	12.99%
C103	923.00	10	188	953.08	10	66	3.26%
C104	913.00	10	263	1064.28	10	88	16.57%
C105	860.00	10	80	878.36	10	40	2.14%
C106	877.00	10	141	852.95	10	48	-2.74%
C107	894.00	10	113	869.66	10	48	-2.72%
C108	853.00	10	151	857.38	10	56	0.51%
C109	854.00	10	240	953.46	10	64	11.65%
C1 Average	**887.33**	**10**	**140**	**932.47**	**10.00**	**54**	**5.09%**
C201	591.00	3	83	591.56	3	30	0.09%
C202	676.00	3	179	625.71	3	81	-7.44%
C203	683.00	3	204	722.28	3	81	5.75%
C204	656.00	3	259	695.99	3	106	6.10%
C205	588.00	3	141	601.43	3	55	2.28%
C206	633.00	3	172	610.24	3	60	-3.60%
C207	601.00	3	159	597.45	3	60	-0.59%
C208	629.00	3	163	600.88	3	70	-4.47%
C2 Average	**632.13**	**3**	**170**	**630.69**	**3**	**68**	**-0.23%**
R101	1805.00	20	207	1951.61	20	37	8.12%
R102	1661.00	19	251	1850.88	18	47	11.43%
R103	1587.00	14	272	1450.29	15	63	-8.61%
R104	1156.00	11	243	1162.83	11	80	0.59%
R105	1517.00	14	228	1665.13	15	42	9.76%
R106	1344.00	13	213	1460.07	13	55	8.64%
R107	1247.00	12	228	1215.64	12	73	-2.52%
R108	1112.00	10	245	1131.35	10	97	1.74%
R109	1334.00	13	251	1412.51	13	66	5.89%
R110	1248.00	12	248	1323.34	12	82	6.04%
R111	1242.00	11	223	1341.79	12	92	8.03%
R112	1148.00	10	232	1103.33	11	118	-3.89%
R1 Average	**1366.75**	**13.25**	**237**	**1422.40**	**13.5**	**71**	**4.07%**
R201	1544.00	4	197	1527.32	4	57	-1.08%
R202	1378.00	4	254	1382.35	4	75	0.32%
R203	1210.00	3	268	1202.74	3	99	-0.60%
R204	946.00	3	372	912.37	3	143	-3.56%
R205	1208.00	3	234	1360.92	3	72	12.66%
R206	1094.00	3	279	1058.00	3	94	-3.29%
R207	1078.00	3	326	979.01	3	114	-9.18%
R208	989.00	2	407	831.08	3	158	-15.97%
R209	1157.00	3	293	1100.82	3	93	-4.86%
R210	1232.00	3	258	1209.98	3	101	-1.79%
R211	980.00	3	364	936.91	3	138	-4.40%
R2 Average	**1165.09**	**3.090909**	**296**	**1136.50**	**3.181818**	**104**	**-2.45%**
RC101	1802.00	16	223	1903.64	16	12	5.64%
RC102	1698.00	14	269	1640.27	14	82	-3.40%
RC103	1502.00	13	322	1504.05	12	66	0.14%
RC104	1502.00	13	327	1494.73	12	68	-0.48%
RC105	1706.00	16	285	1822.94	15	56	6.85%
RC106	1478.00	13	355	1522.03	13	61	2.98%
RC107	1434.00	12	286	1409.96	12	77	-1.68%
RC108	1228.00	11	261	1217.46	11	96	-0.86%
RC1 Average	**1543.75**	**13.5**	**291**	**1564.38**	**13.125**	**64**	**1.34%**
RC201	1810.00	4	176	1636.08	4	54	-9.61%
RC202	1542.00	4	312	1540.54	4	74	-0.09%
RC203	1484.00	3	258	1345.10	4	91	-9.36%
RC204	1113.00	3	336	1000.13	3	124	-10.14%
RC205	1758.00	4	228	1738.77	4	67	-1.09%
RC206	1421.00	4	214	1445.16	4	74	1.70%
RC207	1362.00	4	239	1208.23	4	89	-11.29%
RC208	1099.00	3	356	1015.65	3	125	-7.58%
RC2 Average	**1448.63**	**3.625**	**265**	**1366.21**	**3.75**	**87**	**-5.69%**

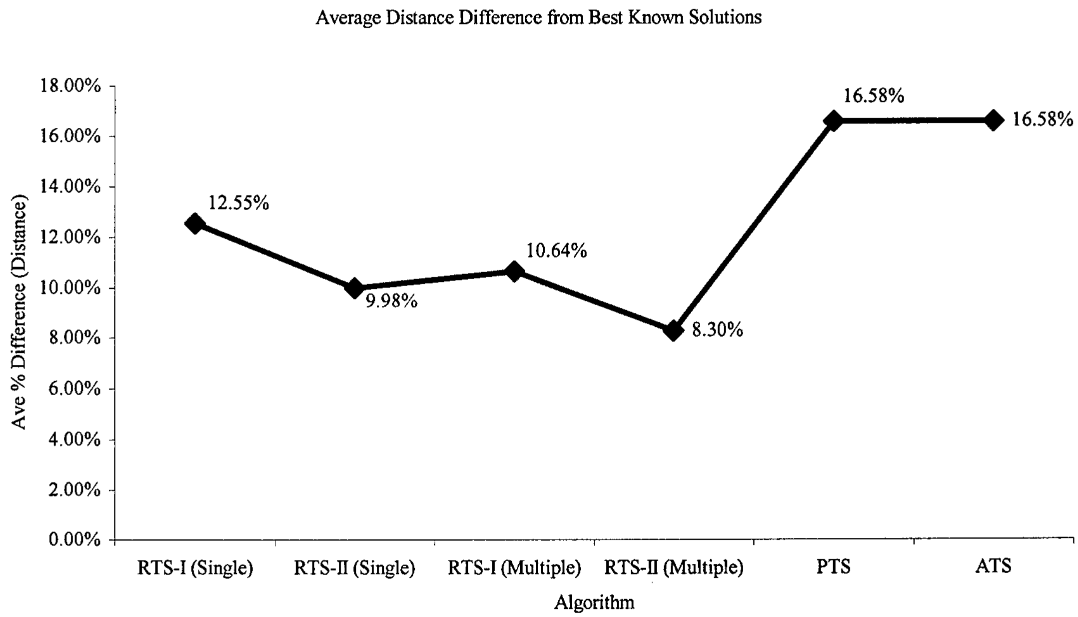

Figure 5: Average Distance Difference from Best Known Solutions

Table 13: Comparing RTS-II (Multiple Starting Solutions) with Best Known

	Best Known			RTS-II (Multiple Starting Solutions)				% Difference	
Problems	Distance	Vehicles	Source	Distance	Vehicles	Time (sec)	Best (sec)	Distance	Time
C101	827.30	10	Desrochers *et al* 1992	828.94	10	74	12	0.20%	--
C102	827.30	10	Desrochers et al 1992	838.73	10	124	70	1.38%	--
C103	826.30	10	Kohl and Madsen 1997	915.76	10	144	34	10.83%	--
C104	822.90	10	Kohl and Madsen 1997	901.46	10	216	77	9.55%	--
C105	827.30	10	Kohl and Madsen 1997	828.94	10	40	11	0.20%	--
C106	827.30	10	Desrochers et al 1992	828.94	10	53	13	0.20%	--
C107	827.30	10	Desrochers et al 1992	828.94	10	61	15	0.20%	--
C108	827.30	10	Desrochers et al 1992	828.94	10	65	16	0.20%	--
C109	827.30	10	Kohl and Madsen 1997	828.94	10	131	75	0.20%	--
C1 Average	**826.70**			**847.73**		**101**	**36**	**2.54%**	**--**
C201	591.56	3	Potvin and Bengio 1996	591.56	3	55	23	0.00%	--
C202	591.56	3	Potvin and Bengio 1996	591.56	3	133	30	0.00%	--
C203	591.17	3	Rochat and Taillard 1995	707.96	3	170	45	19.76%	--
C204	590.60	3	Potvin and Bengio 1996	679.08	3	288	49	14.98%	--
C205	588.88	3	Potvin and Bengio 1996	588.88	3	73	19	0.00%	--
C206	588.49	3	Potvin and Bengio 1996	588.49	3	140	21	0.00%	--
C207	588.29	3	Rochat and Taillard 1995	588.29	3	83	19	0.00%	--
C208	588.32	3	Rochat and Taillard 1995	588.32	3	81	27	0.00%	--
C2 Average	**589.86**			**615.52**		**128**	**29**	**4.35%**	**--**
R101	1607.70	18	Desrochers et al 1992	1708.24	20	230	133	6.25%	--
R102	1434.00	17	Desrochers et al 1992	1637.89	18	276	60	14.22%	--
R103	1207.00	13	Thangiah *et al*1994	1325.13	15	305	238	9.79%	--
R104	1007.31	9	Shaw 1997	1125.87	11	249	48	11.77%	--
R105	1377.10	14	Rochat and Taillard 1995	1505.58	15	182	37	9.33%	--
R106	1252.03	12	Rochat and Taillard 1995	1336.98	13	260	255	6.78%	--
R107	1104.66	10	Shaw 1997	1159.98	12	210	118	5.01%	--
R108	963.99	9	Shaw 1997	1021.92	10	337	229	6.01%	--
R109	1205.96	11	Shaw 1997	1298.83	13	197	61	7.70%	--
R110	1135.07	10	Shaw 1997	1216.80	12	329	223	7.20%	--
R111	1096.73	10	Shaw 1997	1211.01	12	613	289	10.42%	--
R112	953.63	10	Rochat and Taillard 1995	1057.86	11	490	144	10.93%	--
R1 Average	**1195.43**			**1300.51**		**307**	**153**	**8.79%**	**--**
R201	1254.09	4	Kilby et al 1997	1411.39	4	242	79	12.54%	--
R202	1214.28	3	Taillard *et al* 1997	1208.29	4	279	220	-0.49%	--
R203	948.74	3	Rochat and Taillard 1995	1086.25	3	408	129	14.49%	--
R204	867.33	2	Kilby et al 1997	857.13	3	315	150	-1.18%	--
R205	998.72	3	Kilby et al 1997	1202.58	3	218	210	20.41%	--
R206	833.00	3	Thangiah et al1994	1014.42	3	290	68	21.78%	--
R207	814.78	3	Rochat and Taillard 1995	940.19	3	219	210	15.39%	--
R208	738.60	2	Rochat and Taillard 1995	775.72	3	286	105	5.03%	--
R209	855.00	3	Thangiah et al1994	1040.22	3	291	92	21.66%	--
R210	963.37	3	Kilby et al 1997	1112.49	3	312	249	15.48%	--
R211	923.80	2	Taillard et al 1997	873.81	3	260	253	-5.41%	--
R2 Average	**946.52**			**1047.50**		**284**	**160**	**10.67%**	**--**
RC101	1669.00	14	Thangiah et al1994	1715.67	16	140	47	2.80%	--
RC102	1554.75	12	Taillard et al 1997	1560.50	14	168	63	0.37%	--
RC103	1110.00	11	Thangiah et al1994	1390.95	12	237	135	25.31%	--
RC104	1135.48	10	Shaw 1997	1390.95	12	232	133	22.50%	--
RC105	1643.38	13	Taillard et al 1997	1703.65	15	204	83	3.67%	--
RC106	1448.26	11	Taillard et al 1997	1443.56	13	131	38	-0.32%	--
RC107	1230.48	11	Shaw 1997	1348.14	12	170	46	9.56%	--
RC108	1139.82	10	Taillard et al 1997	1169.13	11	144	59	2.57%	--
RC1 Average	**1366.40**			**1465.32**		**178**	**76**	**7.24%**	**--**
RC201	1406.94	4	Kilby *et al* 1997	1619.74	4	141	45	15.12%	--
RC202	1162.80	4	Kilby et al 1997	1396.56	4	313	307	20.10%	--
RC203	1068.07	3	Kilby et al 1997	1246.30	4	387	120	16.69%	--
RC204	803.90	3	Kilby et al 1997	946.95	3	410	399	17.79%	--
RC205	1302.42	4	Kilby et al 1997	1551.50	4	301	237	19.12%	--
RC206	1156.26	3	Kilby et al 1997	1314.68	4	252	243	13.70%	--
RC207	1075.25	3	Kilby et al 1997	1085.64	4	307	299	0.97%	--
RC208	833.97	3	Rochat and Taillard 1995	934.35	3	297	127	12.04%	--
RC2 Average	**1101.20**			**1261.96**		**301**	**222**	**14.60%**	**--**

4.3 Conclusion

Move strategies play an important role in finding good solutions. RTS-II employs a strategy that takes advantage of problem-specific information—the average time window length—and information about the current solution—the average duration of all tours—to determine which moves to generate. RTS-II with multiple starting solutions was the best performing algorithm in terms of minimizing distance. The results for this algorithm averaged within less than 9% of the best know solutions with a running time average of less than 4 minutes. RTS-II with a single starting solution also performed well in terms of minimizing distance; this algorithm averaged within 10% of the best know solutions with a running time average of approximately a minute and a half.

Unfortunately, the probabilistic tabu search did not seriously challenge either Harder's adaptive tabu search, reactive tabu search I, or reactive tabu search II in terms of solution quality. For the vehicle routing problem, a tabu search algorithm should always select the best move.

RTS-II provides higher quality solutions than Harder's adaptive tabu search with solution times comparable to Harder's adaptive tabu search algorithm originally built into the AFIT Router software. Given Harder's algorithm was designed for speed, the RTS-II algorithm seems a logical addition in the AFIT Router.

5. Conclusion

5.1 Contributions

This research enhanced the prototype application for routing unmanned aerial vehicles, called the AFIT Router, previously delivered by AFIT to the UAV Battlelab. This newer version with the *Route Builder* addition allows the user to route aerial vehicles, specifically unmanned aerial vehicles, while considering general restricted operating zones. Ultimately, this enhancement allows for better model realism.

This research demonstrated that a reactive tabu search with a responsive move strategy, RTS-II, could produce good solutions for the TSP and VRP. We provided this additional tabu search solver within the AFIT Router software.

5.2 Recommendations for Further Research

Currently, the *Route Builder* addition only considers static restricted operating zones that are always active. Future work could focus on applying time windows to restricted operating zones or even allowing these zones to move along a path or trajectory to better model obstacles like weather. Terrain may also impact a UAV mission. Thus, the *Route Builder* addition could be improved to handle three-dimensional restricted operating zones that have height as well as defining a geographic region on the ground.

The literature on tabu search presented many advanced tabu and local search techniques that were not considered in this effort. One such technique discussed by Glover (1993) called target analysis can be used to improve TS performance for a specific problem type. Future researchers should consider this analytical technique when setting parameters values for their TS.

Also, future researchers could consider using a limited version of probabilistic tabu search where the probabilistic move selection technique is only used to induce more diversity in the search.

BIBLIOGRAPHY

Asseo, Sabi J. In-flight Replanning of Penetration Routes to Avoid Threat Zones of Circular Shapes. Northrop Grumman Corporation, Pico Rivera CA (1998).

Battiti, Roberto and Giampietro Tecchiolli. "The Reactive Tabu Search," *ORSA Journal of Computing*, 6: 126-140 (1994).

Canan, James W. "Seeing More, and Risking Less, with UAVs," *Aerospace America*, 37(10): 27-31 (October 1999).

Carlton, William B. *A Tabu Search Approach to the General Vehicle Routing Problem*. Ph.D. dissertation. University of Texas, Austin TX (1995).

Chiang, Wen-Chyuan and Robert A. Russell. "Simulated Annealing Metaheuristic for the Vehicle Routing Problem with Time Windows," *Annals of Operations Research*, 63: 3-27 (1996).

Christofides, Nicos. *Graph Theory: An Algorithmic Approach*. London: Academic Press, Inc., (1975).

Clarke, G. and J.W. Wright. "Scheduling Vehicles from a Central Depot to a Number of Delivery Points," *Operations Research*, 12: 568-581 (1964).

Coffin, Marie and Matthew Saltzman. "Statistical Analysis of Computational Tests of Algorithms and Heuristics," *INFORMS Journal of Computing*, 12(1): 24-44 (Winter 2000).

Desrochers M., J. Desrosiers, and M.M. Solomon. "A New Optimization Algorithm for the Vehicle Routing Problem with Time Windows," *Operations Research*, 40: 342-354 (1992).

Flood, Randy A. *A Java Based Human Computer Interface for a UAV Decision Support Tool Using Conformal Mapping*. MS thesis, AFIT/GCS/ENS/99M. School of Engineering, Air Force Institute of Technology (AU), Wright-Patterson AFB OH, (March 1999).

Gendreau, Michel, Alain Hertz, and Gilbert Laporte. "New Insertion and Postoptimization Procedures for the Traveling Salesman Problem," *Operations Research*, 40(6): 1086-1094 (1992).

Gendreau, Michel, Gilbert Laporte, and René Séquin. "A Tabu Search Heuristic for the Vehicle Routing Problem with Stochastic Demands and Customers," *Operations Research*, 40(3): 469-477 (May-June 1996).

Gendreau, Michel, Gilbert Laporte, Christophe Musaranganyi, and Eric D. Taillard. "A Tabu Search Heuristic for the Heterogeneous Fleet Vehicle Routing Problem," *Computers and Operations Research*, 26: 1153-1173 (1999).

Gillett, B.E. and L. Miller. "A Heuristic Algorithm for the Vehicle Dispatching Problem," *Operations Research*, 22: 340-349 (1974).

Glover, Fred. "Future Paths for Integer Programming and Links to Artificial Intelligence," *Computers and Operations Research*, 13: 533-549 (1986).

Glover, Fred. "A User's Guide to Tabu Search," *Annals of Operations Research*, 41: 3-28 (1993).

Glover, Fred and Manuel Laguna. *Tabu Search.* Boston MA: Kluwer Academic Publishers, (1997).

Hachicha, M., M.J. Hodgson, G. Laporte, and F. Semet. "Heuristics for the Multi-vehicle Covering Tour Problem," *Computers and Operations Research*, 27: 29-42 (2000).

Hadjiconstantinou, Eleni and Nicos Christofides. "A New Exact Algorithm for the Vehicle Routing Problem Based on *q*-Paths and *k*-Shortest Paths Relaxations," *Annals of Operations Research*, 61: 21-43 (1995).

Hall, Leslie. "Computational Complexity," in *Encyclopedia of Operations Research and Management Science*. Ed. Saul I. Gauss and Carl M. Harris. Norwell MA: Kluwer Academic Publishers, (1996).

Harder, Robert. *A Java Universal Vehicle Router in Support of Routing Unmanned Aerial Vehicles*. MS thesis, AFIT/GOR/ENS/00M-16. School of Engineering, Air Force Institute of Technology (AU), Wright-Patterson AFB OH, March 2000.

Helsgaun, Keld. "An Effective Implementation of the Lin-Kernighan Traveling Salesman Heuristic," *European Journal of Operational Research*, 126: 106-130 (2000).

Kilby, P., P. Prosser, and P. Shaw. "Guided Local Search for the Vehicle Routing Problem," In *Proceedings of the 2nd International Conference on Meta-heuristics*, (1997).

Kinney, Gary. *A Hybrid Jump Search and Tabu Search Metaheuristic for the Unmanned Aerial Vehicle (UAV) Routing Problem.* MS thesis, AFIT/GOA/ENS/00M-5. School of Engineering and Management, Air Force Institute of Technology (AU), Wright-Patterson AFB OH, March 2000.

Kohl, N. and O.B.G. Madsen. "An Optimization Algorithm for the Vehicle Routing Problem with Time Windows Based on Lagrangian Relaxation," *Operations Research*, 45(3): 395 (1997).

Laporte, Gilbert. "The Traveling Salesman Problem: An Overview of Exact and Approximate Algorithms," *European Journal of Operational Research*, 59: 231-247 (1992a).

Laporte, Gilbert. "The Vehicle Routing Problem: An Overview of Exact and Approximate Algorithms," *European Journal of Operational Research*, 59: 345-358 (1992b).

Laporte, Gilbert, Michel Gendreau, Jean-Yves Potvin, and Frédéric Semet. "Classical and Modern Heuristics for the Vehicle Routing Problem," *International Transactions in Operational Research*, 7: 285-300 (2000).

Lawler, E.L., J.K. Lenstra, A.H.G. Rinnooy Kan, and D.B. Shmoys. *The Traveling Salesman Problem. A Guided Tour of Combinatorial Optimization.* John Wiley & Sons Ltd., (1985).

Løkketangen, Arne and Fred Glover. "Solving Zero-One Mixed Integer Programming Problems Using Tabu Search," *European Journal of Operational Research*, 106: 624-658 (1998).

Nanry, William P. and J. Wesley Barnes. "Solving the Pickup and Delivery Problem with Time Windows Using Reactive Tabu Search," *Transportation Research Part B*, 34: 107-121 (2000).

O'Rourke, Kevin. *Dynamic Unmanned Aerial Vehicle Routing with a Java Encoded Reactive Tabu Search Metaheuristic*. MS thesis, AFIT/GOA/ENS/99M-06. School of Engineering, Air Force Institute of Technology (AU), Wright-Patterson AFB OH, March 1999.

Parker, Gary R. and Ronald L. Rardin. "An Overview of Complexity Theory in Discrete Optimization: Part II. Results and Implications," *IIE Transactions*, 14(2): 83-89 (1982).

Potvin, J.-Y. and S. Bengio. "The Vehicle Routing Problem with Time Windows—Part II: Genetic Search," *ORSA Journal on Computing*, 82(2): 165 (1996).

Renehan, Jeffery N. "Unmanned Aerial Vehicles and Weapons of Mass Destruction: A Lethal Combination," School of Advance Airpower Studies (AU), Maxwell AFB AL, August 1997.

Rochat, Yves and Éric D. Taillard. "Probabilistic Diversification and Intensification in Local Search for Vehicle Routing," *Journal of Heuristics*, 1: 147-167 (1995).

Semet, Frédéric and Éric D. Taillard. "Solving Real-Life Vehicle Routing Problems Efficiently Using Tabu Search," *Annals of Operations Research*, 41: 469-488 (1993).

Shaw, P. "A New Local Search Algorithm Providing High Quality Solutions to Vehicle Routing Problems. APES Group, Dept. of Computer Science, University of Strathclyde, Glasgow, Scotland, UK. (June 1997).

Solomon, Marius M. "Algorithms for the Vehicle Routing and Scheduling Problems with Time Window Constraints," *Operations Research*, 35(2): 254-265 (1987).

Solomon, Marius M. "VRPTW Benchmark Problems." Excerpt from unpublished article, n. pag. http://www.cba.neu.edu/~msolomon/. (15 May 2000).

Stout, Brian W. "Smart Moves: Intelligent Pathfinding." Excerpt from unpublished article, n. pag. http://www.gamasutra.com/features/19970801/pathfinding.html. (August 1997).

Taillard, E., P. Badeau, M. Gendreau, F. Guertain, and J.-Y. Potvin. "A Tabu Search Heuristic for the Vehicle Routing Problem with Soft Time Windows," *Transportation Science*, 32(2) (1997).

Thangiah, S.R., I.H. Osman, and T. Sun. *Hybrid Genetic Algorithms, Simulated Annealing and Tabu Search Methods for the Vehicle Routing Problem with Time Windows*. Technical Report UKC/OR94/4, Institute of Mathematics and Statistics, University of Kent, Canterbury, UK. (1994).

Tsubakitani, Shigeru and James R. Evans. "Optimizing Tabu List Size for the Traveling Salesman Problem," *Computers and Operations Research*, 25: 91-97 (1997).

Tsubakitani, Shigeru and James R. Evans. "An Empirical Study of a New Metaheuristic for the Traveling Salesman Problem," *European Journal of Operational Research*, 104(2): 113-128 (1998).

Wilbur, George F. Strategic Route Planning Using Informed Best-First Search. Boeing Military Airplane Company, Seattle WA. (1998).

Woodruff, David L. and Eitan Zemel. "Hashing Vectors for Tabu Search," *Annals of Operations Research*, 41: 123-137 (1993)

Xu, Jiefeng, Steve Y. Chiu, and Fred Glover. "Fine-tuning a Tabu Search Algorithm with Statistical Tests," *International Transactions in Operational Research*, 5(2): 233-244 (1998).

Zanakis, Stelios H., and James R. Evans. "Heuristic 'Optimization': Why, When, and How to Use It," *Interfaces*, 11(5): 83-89 (June 1982).

REPORT DOCUMENTATION PAGE		Form Approved OMB No. 074-0188

The public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of the collection of information, including suggestions for reducing this burden to Department of Defense, Washington Headquarters Services, Directorate for Information Operations and Reports (0704-0188), 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302. Respondents should be aware that notwithstanding any other provision of law, no person shall be subject to an penalty for failing to comply with a collection of information if it does not display a currently valid OMB control number.
PLEASE DO NOT RETURN YOUR FORM TO THE ABOVE ADDRESS.

1. REPORT DATE *(DD-MM-YYYY)*	2. REPORT TYPE	3. DATES COVERED *(From – To)*
20-03-2001	**Master's Thesis**	MAR 00 – MAR 01

4. TITLE AND SUBTITLE	
ROUTING UNMANNED AERIAL VEHICLES WHILE CONSIDERING GENERAL RESTRICTED OPERATING ZONES	5a. CONTRACT NUMBER
	5b. GRANT NUMBER
	5c. PROGRAM ELEMENT NUMBER
6. AUTHOR(S)	5d. PROJECT NUMBER
Brown, Darin T., 1Lt, USAF	5e. TASK NUMBER
	5f. WORK UNIT NUMBER

7. PERFORMING ORGANIZATION NAMES(S) AND ADDRESS(S)	8. PERFORMING ORGANIZATION REPORT NUMBER
Air Force Institute of Technology Graduate School of Engineering and Management (AFIT/EN) 2950 P Street, Building 640 WPAFB OH 45433-7765	AFIT/GOR/ENS/01M-04

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)		10. SPONSOR/MONITOR'S ACRONYM(S)
Mark O'Hair, Lt Col, USAF UAV Battlelab 1003 Nomad Way, Suite 107 Elgin AFB, FL 32542-6867	mark.ohair@elgin.af.mil Comm: (830) 882-5940 x208 DSN: 872-5940 x208	11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION/AVAILABILITY STATEMENT

APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED.

13. SUPPLEMENTARY NOTES

14. ABSTRACT

US military forces employ unmanned aerial vehicles (UAVs) to conduct intelligence-gathering missions worldwide. For a typical mission, commanders may task UAV operators to gather imagery on 100 or more sites or targets. UAV operators must quickly prepare mission plans that meet the needs of their commanders while dealing with real-world constraints such as time windows, site priorities, imagery requirements, UAVs with different capabilities (i.e. imagery equipment, speed, and range), and UAVs departing from different bases. Previous AFIT research provided the UAV Battlelab with a tool, AFIT Router, for generating high-quality routes to aid mission planning. This research enhances the AFIT Router by providing the ability to define general restricted operating zones and to build routes that consider these zones. This research also examines and compares a probabilistic tabu search heuristic and two reactive tabu search heuristics for solving vehicle routing problems.

15. SUBJECT TERMS

Unmanned Aerial Vehicles, Routing Around Obstacles, Tabu Search, Vehicle Routing Problem, Java

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
					Raymond Hill, Lt Col, USAF
a. REPORT	b. ABSTRACT	c. THIS PAGE			19b. TELEPHONE NUMBER *(Include area code)*
U	U	U	UU	59	(937) 255-6565, ext 4323 (Raymond.Hill@afit.edu)