



**NAVAL
POSTGRADUATE
SCHOOL**

MONTEREY, CALIFORNIA

THESIS

**A HEURISTIC ALGORITHM FOR OPTIMIZED ROUTING
OF UNMANNED AERIAL SYSTEMS FOR THE
INTERDICTION OF IMPROVISED EXPLOSIVE DEVICES**

by

Michael S. Scioletti

June 2008

Thesis Advisor:

Johannes O. Royset

Second Reader:

W. Matthew Carlyle

Approved for public release; distribution is unlimited

THIS PAGE INTENTIONALLY LEFT BLANK

REPORT DOCUMENTATION PAGE			<i>Form Approved OMB No. 0704-0188</i>	
Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instruction, 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 this collection of information, including suggestions for reducing this burden, to Washington headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188) Washington DC 20503.				
1. AGENCY USE ONLY (Leave blank)	2. REPORT DATE June 2008	3. REPORT TYPE AND DATES COVERED Master's Thesis		
4. TITLE AND SUBTITLE: A Heuristic Algorithm for Optimized Routing of Unmanned Aerial Systems for the Interdiction of Improvised Explosive Devices			5. FUNDING NUMBERS	
6. AUTHOR(S) Major Michael Scott Scioletti, Usa			8. PERFORMING ORGANIZATION REPORT NUMBER	
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) Naval Postgraduate School Monterey, CA 93943-5000			10. SPONSORING/MONITORING AGENCY REPORT NUMBER	
9. SPONSORING /MONITORING AGENCY NAME(S) AND ADDRESS(ES) N/A			11. SUPPLEMENTARY NOTES The views expressed in this thesis are those of the author and do not reflect the official policy or position of the Department of Defense or the U.S. Government.	
12a. DISTRIBUTION / AVAILABILITY STATEMENT Approved for public release; distribution is unlimited			12b. DISTRIBUTION CODE A	
13. ABSTRACT (maximum 200 words) Improvised explosive devices (IEDs) are effective weapons for insurgents targeting conventional military and security forces. Real-time information gathering about likely use of such weapons is one approach to reduce the effectiveness of IEDs. Unmanned aerial system (UASs) may provide the information gathering capability commanders need to interdict IEDs. Currently, UASs are not systematically utilized in that capacity. This research develops a routing tool that uses column-generation techniques and a greedy algorithm to route UASs through suspected IED locations for the purpose of IED interdiction as it transit to and from command directed missions. In empirical studies of data sets with up to 125 IED locations and missions, the routing tool provides optimal or near-optimal solutions in all instances tested. The tool produces de-conflicted routes for up to three UASs within five minutes of computing time.				
14. SUBJECT TERMS routing, IED, orienteering problem, UAV, UAS, greedy heuristic, column generation			15. NUMBER OF PAGES 61	
			16. PRICE CODE	
17. SECURITY CLASSIFICATION OF REPORT Unclassified	18. SECURITY CLASSIFICATION OF THIS PAGE Unclassified	19. SECURITY CLASSIFICATION OF ABSTRACT Unclassified	20. LIMITATION OF ABSTRACT UU	

THIS PAGE INTENTIONALLY LEFT BLANK

Approved for public release; distribution is unlimited

**A HEURISTIC ALGORITHM FOR OPTIMIZED ROUTING OF UNMANNED
AERIAL SYSTEMS FOR THE INTERDICTION OF IMPROVISED EXPLOSIVE
DEVICES**

Michael S. Scioletti
Major, United States Army
B.S. United States Military Academy, 1998

Submitted in partial fulfillment of the
requirements for the degree of

MASTER OF SCIENCE IN OPERATIONS RESEARCH

from the

**NAVAL POSTGRADUATE SCHOOL
June 2008**

Author: Michael S. Scioletti

Approved by: Johannes O. Royset
Thesis Advisor

W. Matthew Carlyle
Second Reader

James N. Eagle
Chairman, Department of Operations Research

THIS PAGE INTENTIONALLY LEFT BLANK

ABSTRACT

Improvised explosive devices (IEDs) are effective weapons for insurgents targeting conventional military and security forces. Real-time information gathering about likely use of such weapons is one approach to reduce the effectiveness of IEDs. Unmanned aerial systems (UASs) may provide the information gathering capability commanders need to interdict IEDs. Currently, UASs are not systematically utilized in that capacity. This research develops a routing tool that uses column-generation techniques and a greedy algorithm to route UASs through suspected IED locations for the purpose of IED interdiction as it transits to and from command priority missions. In empirical studies of data sets with up to 125 IED locations and missions, the routing tool provides optimal or near-optimal solutions in all instances tested. The tool produces de-conflicted routes for up to three UASs within five minutes of computing time.

THIS PAGE INTENTIONALLY LEFT BLANK

TABLE OF CONTENTS

I.	INTRODUCTION.....	1
	A. BACKGROUND	1
	B. RESEARCH GOAL	5
	C. LIMITATIONS AND ASSUMPTIONS	5
	D. STRUCTURE OF THESIS AND CHAPTER OUTLINE	6
II.	LITERATURE REVIEW	7
	A. IED PREDICTION MODELS.....	7
	B. THE VEHICLE ROUTING PROBLEM (VRP)	8
	1. VRP Heuristic Algorithms	8
	2. VRP Exact Algorithms	9
	C. THE ORIENTEERING PROBLEM (OP)	9
	1. TOP Heuristic Algorithms	10
	2. OP Exact Solution Techniques	11
	D. MILITARY ROUTING APPLICATIONS	11
	E. LITERATURE REVIEW CONCLUSIONS	14
III.	MODEL AND ALGORITHM DEVELOPMENT.....	15
	A. MODEL DEVELOPMENT	15
	1. RIP Formulation	15
	2. Integer Program Model Discussion.....	18
	3. RIP Limitations.....	20
	B. HEURISTIC ALGORITHM	21
	1. VBA Pre-Processing.....	21
	2. Feasible Route (Column) Generation.....	23
	3. RIP-M for Route Selection.....	26
IV.	COMPUTATIONAL STUDY.....	29
	A. TEST DATA	29
	B. RESULTS	31
	C. SENSITIVITY ANALYSIS.....	36
V.	CONCLUSIONS AND FUTURE WORK	39
	A. CONCLUSIONS	39
	B. FUTURE WORK	39
	1. Route Generation	39
	2. Dynamic Updates	40
	3. UAS Routing Tactics.....	40
	LIST OF REFERENCES	41
	INITIAL DISTRIBUTION LIST	45

THIS PAGE INTENTIONALLY LEFT BLANK

LIST OF FIGURES

Figure 1	IED Casualties in Iraq (icasualties.org, 2008).	3
Figure 2	Example of graph of missions and UAS Route {1-9-3-7-10-1}.....	15
Figure 3	Graphical display of Mission and IED Nodes ($I=125$).....	22
Figure 4	Example of Route Domination Principle.....	24
Figure 5	Mission Sequence Example.....	24
Figure 6	Time Windows of Mission and IED Nodes for test data set ($I=75$).	31
Figure 7	Graph of Optimal Values of RIP, Heuristic, and RLP.....	33
Figure 8	Graph of Optimal Values of RIP, Heuristic, and RLP.....	34
Figure 9	Graph of Optimal Values of RIP, Heuristic, and RLP.....	35
Figure 10	Example Graphical Output from Routing Tool (Test Data Set $I=125$).	36

THIS PAGE INTENTIONALLY LEFT BLANK

LIST OF TABLES

Table 1	Types of UAS (FM 3x04-15, 2006).....	2
Table 2	Parameter settings for test data. All times in minutes.....	30
Table 3	Optimal Values of RIP, Heuristic, and RLP.....	32
Table 4	Optimal Values of RIP, Heuristic, and RLP.....	33
Table 5	Optimal Values of RIP, Heuristic, and RLP.....	34
Table 6	Results from Setting Sensitivity Analysis.....	37
Table 7	Time Results from Setting Sensitivity Analysis.....	37

THIS PAGE INTENTIONALLY LEFT BLANK

EXECUTIVE SUMMARY

Improvised explosive devices (IEDs) are effective weapons for insurgents targeting conventional military and security forces. Real-time information gathering about the likely use of such weapons is one approach to reduce the effectiveness of IEDs. Unmanned aerial system (UASs) may provide that information gathering capability commanders need to interdict IEDs.

Current Brigade Combat Team (BCT) routing procedures for UASs do not maximize the utilization of the UAS. This thesis develops a routing tool for UASs that changes UAS routing protocols to achieve higher utilization of the UASs during their flight time. The routing tool directs UASs to IED hotspots for interdiction as they transit between other missions. Given this tool, BCTs will no longer have to rely upon UAS operators to conduct arbitrary searches if the UAS arrives early to a mission, instead UAS operators will search specific IED hotspots between missions.

We model the UAS routing problem for IED interdiction as an integer program, which we call RIP, and present a heuristic algorithm to solve the problem. The routing tool receives inputs from the user which include locations of IED hotspots and missions. Each IED hotspot and each mission corresponds to a node in a complete graph. Each node has a reward assigned by the commander as well as operational parameters such as service time or time on target, early service start time, and late service start time. The routing tool uses a depth first search inside a column generation procedure that produces feasible routes for one UAS. These feasible routes are outputted to a master IP, which we call RIP-M solved by CPLEX or the Microsoft Excel Solver. RIP-M deconflicts multiple routes with the objective to maximize the total collection of rewards of the UASs.

In empirical studies on data sets with up to 125 nodes, the routing tool provides optimal or near-optimal solutions in all instances tested. The tool produces de-conflicted routes for up to three UASs within five minutes of computing time. In comparison, standard IP solvers such as CPLEX take more than four hours to optimally solve small instances and exceed memory limitations on larger instances.

THIS PAGE INTENTIONALLY LEFT BLANK

I. INTRODUCTION

A. BACKGROUND

Unmanned aircraft systems (UASs) have many features that have proved to be successful in support of combat operations, and they have been labeled as transformational technologies that could change how wars are fought and won (UAS Roadmap, 2005). In recent combat operations in Iraq and Afghanistan, the United States Military has achieved tactical successes from its use of a variety of UASs and their sensors, communications, and armaments payloads (FM 3x04-15). Therefore, Congress has endorsed and funded a UAS introduction and implementation plan which is designed to put cutting edge UAS technology into the hands of more commanders over the next 25 years (UAS Roadmap, 2005). The changes in technology coupled with the increased number of systems available for tactical use will definitely increase the opportunities for more diversity in UAS employment. We define a UAS in accordance with Army FM 3x90-6:

A powered, aerial vehicle that does not carry a human operator; uses aerodynamic forces to provide vehicle lift; can fly autonomously or is remotely operated; can be expendable or recoverable; and can carry a lethal or non-lethal payload.

UASs can be categorized into three main classes: man-portable, tactical, and theater as displayed in Table 1, which are defined by FM 3x04-15 as such:

- Man-portable UAS (MUAS). These UASs are small, self-contained, and portable. They usually operate below the coordinating altitude and support small ground combat teams/elements in the field. Generally, they are controlled (flown) by a single individual who also views the sensor images on a small laptop-type computer.
- Tactical UAS (TUAS). Tactical UASs are larger systems that support maneuver commanders at various tactical levels of command and can also support the small combat teams when so employed.
- Theater UAS (THUAS). Theater UASs are generally deployed to support theater-wide requirements. They permit varied support to combat team and subordinate tactical command levels depending on the type of UAS.

	Speed (knots)		Maximum Altitude	Maximum Endurance ¹	Maximum Range
	Cruise	Dash			
Man-portable					
Desert Hawk	50	50	1,000 AGL	1 hr	10 km ²
Dragon Eye	45	45	500 AGL	1 hr	10 km ²
Pointer	45	45	985 AGL	1 hr	5 km ²
Raven	34	60	1,000 AGL	1 hr	10 km ²
Silver Fox	50	50	1,000 AGL	10 hrs	32 km ²
Tactical					
Hunter	70	100	15,000 MSL	8-9 hrs	125 km ^{2,8,4}
IGNAT	70	120	25,000 MSL	30 hrs	500 nm ²
Pioneer	65	110	15,000 MSL	5 hrs	185 km ²
Scan Eagle	70	70	19,000 MSL	15 hrs	110 km ²
Shadow	70	105	15,000 MSL	5 hrs	50 km ²
Fire Scout	80	125	20,000 MSL	6 hrs	280 km ²
ERMP	70	150	29,000 MSL	36 hrs	1200 km ²
Theater					
Global Hawk	310	340	60,000 MSL	28 hrs	5,400 nm ³
Predator	70	117	26,000 MSL	20 hrs	500 nm ³
Predator B	90	220	45,000 MSL	16 hrs	500 nm ³
¹ Endurance equals total time from takeoff to landing. ² Range equals range from GCS. ³ Range equals total mechanical range of UA. ⁴ With retransmission, range equals 200 km.					

Table 1 Types of UAS (FM 3x04-15, 2006).

UAS operations continue to support battlefield commanders in new and emerging ways and are currently integrated into nearly every facet of military operations such as: air interdiction, artillery fire support, close air support, battlefield damage assessment, communication relay, intelligence, laser operations, personnel recovery, surveillance and reconnaissance, target acquisition, and psychological operations (FM 3x04-15, 2006). This list is not exhaustive; new ways to employ UASs are constantly evolving. The focus and anticipation of such developing technologies and uses is due in large part to the ability of UASs to provide the commander with real-time situational awareness without exposing soldiers to direct conflict.

An improvised explosive device (IED) is defined as a device placed or fabricated in a makeshift manner incorporating “destructive, lethal, noxious, pyrotechnic, or incendiary chemicals and designed to destroy, incapacitate, harass, or distract” the target (JIEDDO, 2007). It may incorporate military parts, but for the most part, IEDs are normally built from ad hoc materials that do not require sophisticated military components. Currently, IEDs are a leading cause of casualties among U.S. forces in Iraq

and Afghanistan. Figure 1 illustrates the number of fatalities from IEDs during the Iraq war alone. The number of IED casualties has fluctuated month to month, but consistently remains substantial.

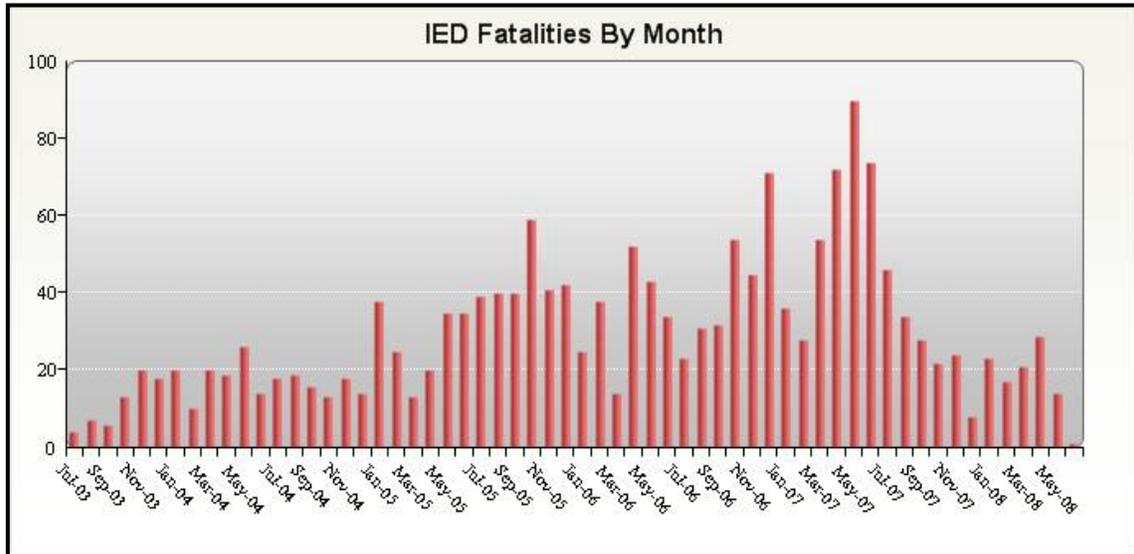


Figure 1 IED Casualties in Iraq (icasualties.org, 2008).

Both prior to and during the early stages of combat operations in Iraq and Afghanistan, the primary use of UASs was strictly for offensive operations; however, due to the sustained use of IEDs in both theaters, UASs are now being sought by commanders to assist in IED interdiction operations. Therefore, commanders must now balance UAS support for high-priority maneuver operations with their use to search for IEDs and related activities. Last year the U.S. Army deployed approximately twenty-five Shadow UASs to Baghdad alone, flying some 15,000 hours a month, while performing a variety of missions (Defense Data, 2007).

Most prominent among the Shadow's "overwatch" missions in Baghdad were:

- IED Defeat - Where the Shadow is used to monitor the emplacement of improvised explosive devices, leading to their removal or defeat.
- Counter-IED - Where the Shadow is used to deter the emplacement of IEDs or, once they are emplaced, to track/trace and defeat their networks of facilitators, communicators, and suppliers.

- Counter-IDF - Where the Shadow is used to identify the source/location of indirect fire against U.S. sites and to cue rapid counter-strikes against these sources.

These tactics using UASs have been proven in operational environments to work, and one way to further increase UAS effectiveness is to incorporate them into standard UAS routing procedures. In this thesis, we develop a routing tool that improves the utilization of UASs by directing them to “IED hotspots” as they transit between other missions.

We define an IED hotspot to be an area target that is perceived to have an increased likelihood of a buried IED or an IED cache being located there. We say that an IED is “detected” at an IED hotspot if a UAS visually identifies someone placing an IED, visually identifies an IED in the ground, or identifies a location where there is a collection of munitions which could be used as an IED cache. When a UAS reaches an IED hotspot, it inspects the area for IEDs. If there is an IED detection, the UAS confirms the detection, maintains coverage, and provides situational awareness to ground forces responding to the detection. We refer to this sequence of actions as an IED interdiction mission.

The operational environment is “a composite of the conditions, circumstances, and influences that affect the employment of military forces and bear on the decisions of the unit commander” (FM 3x90-6, 2006). The U.S. Army has recently re-organized the majority of its land combat forces into Brigade Combat Teams (BCTs) that effectively operate in the contemporary operational environment that exists today, and will exist well into the near future. Regardless of type (heavy, infantry, or Stryker), BCTs are the Army’s basic tactical maneuver units, and the smallest combined arms units that can be committed independently. BCTs are designed to conduct offensive, defensive, and stability operations. “Their core mission is to close with the enemy by means of fire and maneuver to destroy or capture enemy forces, or to repel their attacks by fire, close combat, and counterattack” (FM 3x90-6, 2006). Internal to each of these BCTs is the Shadow TUAS. The TUAS is the BCT commander's most versatile confirming sensor and responds directly to his requirements (FAS, 2000).

Current BCT routing procedures do not systematically route UASs to IED hotspots as they transit between other missions. We note, however, that BCTs do execute IED interdiction missions separately. In this thesis, we distinguish between IED interdiction missions and “other” missions such as reconnaissance of terrain and enemy forces, area surveillance, area security, target acquisition and designation, and battle damage assessment, and communications support. We refer to the other missions simply as *missions* and specifically label the examination of an IED hotspot as an *IED interdiction mission*. The importance of missions and IED hotspots are distinguished by the commander’s priority for UAS employment. Given the relatively low number of operational UASs in a BCT, it is currently unrealistic to assume that the sole mission of a UAS will be IED search. Therefore, UASs will typically be routed to IED hotspots when it is compatible with the execution of other missions. Using IED hotspots as waypoints for mission routing does not guarantee detection; rather it provides increased opportunities to leverage current technologies to further assist the commander in preventing or responding to an IED.

B. RESEARCH GOAL

At the BCT level, planners do not possess a routing tool that is implementable using software inherent to every Army computer. Therefore, in this thesis we develop a routing tool consisting of an integer program to model the UAS routing problem and a heuristic algorithm that obtains high-quality solutions to the integer program within a few minutes. The short computing time facilitates rapid re-planning of UAS routes in response to changing environments. The routing tool takes missions and IED hotspots as inputs and produces high-quality UAS routes that efficiently utilize the UAS given flight time constraints. The routing tool is implemented in Microsoft Excel and is therefore compatible with the most common computers in the Department of Defense (DoD).

C. LIMITATIONS AND ASSUMPTIONS

Our routing tool is designed for the BCT and its TUAS planning requirement of routing no more than four TUASs simultaneously. It does not attempt to develop routes

for larger, more strategic, types of UASs like the Predator or Hunter, or for smaller types like the Raven. It is based on a deterministic model that assumes given data will remain constant for the duration of a UAS route. If this is not the case, the routing tool must be re-initiated with the updated data.

UAS operations require line of sight (LOS) communications between the UAS and its ground control station (GCS) throughout the duration of the route. Troy (2008) reported that throughout a 15-month deployment to Iraq, the GCS operated from a static location and effectively maintained LOS coverage with all UASs for missions of length up to 75 km. Therefore, the routing tool assumes a fixed GCS location.

The routing tool assumes that UASs have a dedicated airspace to fly. This prevents UAS operators from having to deconflict their route with other low flying aircraft including helicopters to prevent collision. This routing tool, however, provides deconfliction between UASs in the dedicated airspace.

Further, the routing tool does not track actual movement of the UAS. It only provides a waypoint-to-waypoint flight plan. If the UAS changes course due to unexpected events, or if other conditions change, the routing tool must be re-initiated with the new conditions. The routing tool ignores the chance of mechanical failure or hostile actions that may incapacitate a UAS.

D. STRUCTURE OF THESIS AND CHAPTER OUTLINE

This thesis is organized into five chapters including the Introduction. Chapter II provides a review of literature on IED defeat efforts and UAS employment as well as a brief background on the field of Vehicle Routing Problems (VRP) and Orienteering Problems (OP). In Chapter III, we present a model of the UAS routing problem formulated as an Integer Program. This integer program is solved by a heuristic algorithm based on column generation techniques and implemented in Visual Basic for Applications (VBA). In Chapter IV, we compare the heuristic algorithm solution to the integer program solution. Chapter V summarizes the research, presents the main findings and insights, and discusses the potential future work.

II. LITERATURE REVIEW

A. IED PREDICTION MODELS

For the most part, commanders do not know where and when the next IED attack will occur; therefore, mathematical models known as IED prediction models were designed to help commanders identify locations where an IED is likely to occur given specific data inputs. This likelihood is normally represented on a map similar to elevation data by color bands that represent area likelihoods as opposed to individual point probabilities. These models are not used to pinpoint activity, but identify hotspots or locations which demonstrate an increased likelihood of an IED as compared to the entire operating area. Empirical tests show that these models in concept have the potential to provide useful results to a commander. At least two different models have been developed to predict the placement of IEDs specifically in Iraq: Riese (2006) developed a tool known as *Threat Mapper* and Ahner (2008) developed a mathematical model based on unit activity as recorded by blue force tracker (BFT).

Threat Mapper, initially developed as a tactical tool to help soldiers with a variety of spatial forecasting and identification problems such as mortar attacks, weapons caches and safe houses, is now being leveraged in similar capacity to identify IED hotspots. These locations not only include actual IEDs on the ground, but the weapons caches that support the emplacement of IEDs. *Threat Mapper* uses “advanced pattern analysis techniques to measure the spatial similarity between an area of interest and an observed behavior” (Riese, 2006). The results of this measurement are presented to the analyst in a threat map, which provides increased situational awareness to the unit thereby facilitating better tactical planning. It does not predict success in finding an IED.

More recently, analysts from the U.S. Army TRADOC Analysis Center (TRAC) Monterey developed a mathematical model tested for use in the Iraqi battle-space that captures the interaction between U.S. Coalition Force activities as recorded by a blue force tracker (BFT) and locations of IEDs. The TRAC model aims to identify the time when U.S. forces should decrease current activity levels to prevent being hit by an IED to

using technological assets like UASs for the purposes of reconnaissance, surveillance, and target acquisition missions. TRADOC developed eight models using four separate regions in Iraq with differing terrain, forces, and demographics. Models were developed for each area, lending insight into which BFT patterns over time were more likely to result in IED events, the likelihood of repeat IED events, and the effectiveness of cache clearing operations (Ahner, personal communication, April 2, 2008). The results of this model can be used by planners to further identify IED hotspot locations. These models have shown promise as decision aids for U.S. commanders in Iraq in planning convoys, patrols, and other ground operations.

B. THE VEHICLE ROUTING PROBLEM (VRP)

The VRP is a combinatorial optimization problem seeking to service a number of customers with a fleet of vehicles. The objective function of the VRP is to minimize cost or minimize the number of vehicles required to service all of the customers exactly once. Some VRP problems model the movement or flow of goods along a route and require vehicle capacity constraints not to be exceeded. In the VRP with time windows (VRPTW), additional constraints that identify allowable delivery times increase the difficulty of the problem (Solomon, 1987). The VRP is the subject of intensive research efforts that involve both exact and heuristic algorithms; however, due to the combinatorial nature of the problem and its classification as an NP-Hard problem, the majority of research has focused on heuristics (Chiang & Russell 1997).

1. VRP Heuristic Algorithms

Several heuristics have been used in an attempt to overcome the difficulties of solving VRP and VRPTW. Solomon (1987) acknowledges that tour-building algorithms for the VRPTW are divided into sequential and parallel methods, which means that routes will be constructed either one by one or all at once. In this article, he determines that his insertion method yields the best and most adaptable results of the six tested. The objective of his insertion method is to choose a location in the current route for an un-routed customer that minimizes a measure of the extra distance and extra time required to

include it in the route (Solomon, 1987). Osman (1993) achieved good results by applying a simulated annealing heuristic to solve the VRP by moving one customer from one route to another or by exchanging two customers from two routes. Chiang and Russell (1997) implement the concept of parallel construction for their initial routes and use reactive tabu search, which attempts to avoid local minima and limit cycles by varying the tabu list size. Chiang and Russell's tabu search heuristic equaled or exceeded previously published results in 23 out of the 56 test instances (Chiang & Russell, 1997). Tan et al. (2001) developed three heuristics for solving the VRTPW: simulated annealing, tabu search, and genetic algorithms. Both simulated annealing and genetic algorithms produce acceptable solutions, but tabu search is reported to be the most effective heuristic for producing near-optimal solutions; however, the other two heuristics are both much faster.

2 VRP Exact Algorithms

Desrosiers, Soumis, and Desrochers (1984) embedded column generation techniques within a linear-programming-based branch-and-bound framework for solving vehicle routing problems with time window constraints exactly. The advantage of column generation is that it obtains an optimal solution in finite computing time. The American Red Cross (ARC) (2003) did a study to solve a blood processing VRPTW using a depth first route generation search with column generation techniques. In their analysis, the ARC used column generation techniques to store all the feasible routes developed in the early stage of their algorithm, they then solve an integer program to choose a subset of the routes that satisfies the blood collection constraint while minimizing the total traveling distance. The results of this model were not compared to any known instances, but produced an efficient solution for the blood distribution in the state of Connecticut.

C. THE ORIENTEERING PROBLEM (OP)

The OP is based on the concept of the individual outdoor sport usually played in a mountainous or heavily forested area where a person's score is based on the number of points reached within a certain time limit. If the individual fails to return to the start

point by the required end time, he is disqualified, so it is up to him to plan his route accordingly to maximize total points and return before the time constraint. The OP with multiple players is referred to as the Team OP (TOP), while the OP or TOP with time windows is referred to as OPTW or TOPTW. The TOP is a variant of the Vehicle Routing Problem in which a set of vehicle tours are constructed such that the total collected reward received from visiting a subset of customers is maximized and the length of each vehicle tour is restricted by a pre-specified time.

1. TOP Heuristic Algorithms

Early efforts to solve the TOP began with Ramesh and Brown (1991) which proposed a 4-phase heuristic for the OP that consisted of insertion, improvement, deletion, and final insertion phases. Chao, Golden, and Wasil, who developed a 2-point exchange heuristic based on the notion of record-to-record improvement, were also successful in solving instances of the TOP (Chao, Golden, & Wasil 1996) achieving favorable results compared to 67 test instances previously published and 40 new tests problems. Tang and Miller-Hooks (2005) developed a tabu search heuristic embedded in an adaptive memory procedure that alternates between small and large neighborhood stages during the solution improvement phase, which employs both greedy and random procedures for neighborhood solution generation. Archetti and Hertz (2006) compare two generalized tabu search heuristics with a variable neighborhood search (VNS) in solving the TOP.

Kantor and Rosenwein (1992) developed two heuristic algorithms to solve instances of the OPTW which included both an insertion and tree heuristic. Their insertion heuristic constructs a path by iteratively inserting a previously un-routed customer node into a route in the best slot (Kantor & Rosenwein, 1992). Although the insertion heuristic tended to have shorter computational running times (as it considers less of the feasible solution space), the tree heuristic obtains improved values of total reward for the OPTW. The tree heuristic is based on a depth first search which begins with a partial route and adds nodes to it until either a complete route is generated or until one of the six rules developed by the authors suggest that the route be abandoned. The

algorithm maintains track of the route with the highest total reward and compares against all subsequent routes produced. Unfortunately, Kantor and Rosenwein only found optimal solutions for instances having up to 25 total nodes with their tree heuristic.

2. OP Exact Solution Techniques

Butt and Ryan (1999) developed an optimal solution procedure for the multiple tour maximum collection problem (MTMCP) using column generation. Their solution procedure works well on realistic size problems up to 100 total nodes, particularly when the total nodes visited in any tour is relatively small. Boussier, Feillet, and Gendreau (2007) developed an exact algorithm for team orienteering problems, which uses a column generation approach. The authors note that their model is a first of its kind for the TOP and that the only similar published algorithm is Gueguen (1999), who solved a selective VRPTW. In their efforts to obtain an optimal solution, the authors embed the column generation into a branch-and-price scheme.

D. MILITARY ROUTING APPLICATIONS

We now consider models specifically developed for routing military aircraft and UASs. Moser (1990) modeled Tactical Aerial Reconnaissance Vehicles routing using the Multi-Player Orienteering Problem with Time Windows, which is similar to the TOPTW. He developed both an integer program and heuristic to solve a Marine Corps aircraft routing problem that is nearly identical in concept to a UAS routing problem. He relies on his heuristic to generate solutions to his empirical study as he claims that the integer program is not solvable in any reasonable time frame. The heuristic is based on a greedy insertion method and is implementable in FORTRAN capable of inputting various targets (customers), a heterogeneous vehicle fleet (aircraft with different fuel/speed constraints), alternate drop-off locations, and commander's priority on target value to produce an acceptable solution (Moser, 1990).

Hill, Bailey, O'Rourke, and Carlton (2001) use the VRPTW as a base for their heuristic algorithm that uses a Java-encoded heuristic to dynamically route UASs using reactive tabu search. It employs three options for arranging the initial solution, which are

as follows: listed ordering, time window midpoint, and randomized ordering. The solution neighborhood is generated by 3-opt exchanges, which execute swap moves followed by an insertion move. Unlike Archetti and Hertz (2006) whose VNS algorithm does not allow infeasible solutions in the neighborhood search, Hill et al. recognize that past vehicle routing problems have feasible solution spaces that are disjoint, so their algorithm accepts infeasible solutions, which is facilitated through the use of high penalty factors. The tabu list-length is based on the number of iterations occurring between cycles. This model is also capable of receiving and adapting to emerging priority targets and locked and forbidden routes. This model was compared to known VRP solutions for total node values of 25, 50, and 100 in which it produces optimal solutions within reasonable computing time in most examples.

Harder, Hill, and Moore (2004) develop a more robust router than Hill et al. (2001) for UASs for the Air Force, specifically Predator operations, which is also encoded in Java. Harder et al.'s Java encoded routing heuristic solves instances for heterogeneous vehicles, vehicle endurance limits, time windows, and time walls for the sites requiring coverage, site priorities, and asymmetric travel distances. The architecture for this problem is based on the idea of reuse rather than reinvention or re-implementation (Harder et al., 2004). The object oriented capabilities of Java facilitate the creation of different classes and use existing classes to develop a solution technique that responds to user input. This algorithm uses a core router that processes user input, parameters, and stores the data that calls on a universal vehicle router that provides a means to solve the VRP. The basis for this prototype is a general tabu search algorithm. Through the use of a "listening" prompt, objects listen for key events to trigger specific tabu search strategies such as intensification, diversification, and strategic oscillation (Harder et al., 2004). Harder et al.'s results are comparable to results achieved by other heuristic algorithms that have produced solutions for the VRP of total nodes up to 100.

In addition, Shetty, Sudit, and Nagi (2008) recently developed a new technique to solving a Combat UAS (CUAS) routing problem that utilizes a decomposition scheme with two problems: target assignment, which is modeled as a minimum cost network flow problem, and a VRP, which routes each UAS. A tabu search heuristic coordinates

the solutions for each of the problems by solving a traveling salesman problem (TSP) for routing feasibility. This heuristic requires data inputted with the following information: priorities of targets, service level required to destroy each target, service level required to avoid collateral damage, heterogeneous fleet, fuel constraint conditions, and weaponry load constraint on each CUAS. The authors feel that their model with additional constraints and coding can be employed for time-sensitive targets, but do not provide or attempt formulations that are capable of handling these added inputs. They introduce a concept of a fractional service by means of payload to provide to some level of support to targets that cannot be fully serviced. This concept has value given UASs may not be able to remain at a mission as long as needed, but could provide alternate coverage durations to meeting possible minimum support constraints. The authors acknowledge that this model needs major testing before it is ready for implementation into military operations.

Reber and Royset (2008) developed an integer linear program referred to as IED Search Optimization Model (ISOM) that uses substantial pre-processing techniques to reduce the number of arcs in an undirected graph to solve the orienteering problem for multiple UASs on a discretized grid based on output from an IED prediction model. Tactical level operators can use ISOM to determine routes that will best employ their UAS for the purpose of detecting IEDs or IED related activities. ISOM receives output from an existing IED prediction model and uses it to establish relative values for searching various portions within a sector of operation. ISOM accounts for factors such as winds, sensor sweep-width, and aircraft de-confliction (Royset & Reber, 2008). Royset and Reber's integer program is an exact algorithm that only considers counter-IED operations; however, this model provides a framework for added research on incorporating the concepts of routing UASs for the purpose of IED interdiction. Further, pre-processing efforts of this model are useful in minimizing the scope of the problem which assists operators by reducing computation time. This model, however, assumes that the sole mission of the UAS is IED detection and does not consider the support of maneuver operations in planning the route nor does it consider time windows of execution. It also requires industrial strength commercial solvers like CPLEX to solve.

E. LITERATURE REVIEW CONCLUSIONS

There has been a tremendous amount of research on the VRPTW and relatively little on the TOPTW; however, it is more realistic to model our UAS routing problem as a TOPTW. A BCT has four TUASs and at least eight battalions under its control; it is more likely that the number of missions identified by planning is greater than the actual number of missions the available number of UASs can accomplish within their respective endurance constraints. Further, UAS missions are most likely tied to operations that occur during specified time intervals or time windows that are linked to ground unit timelines; therefore, it is essential the routing tool be able to handle these types of constraints as well. Existing UAS routing tools target the routing of strategic UASs that support a Corps or Joint Task Force and do not directly address the need for a routing tool for planners and operators in a BCT.

III. MODEL AND ALGORITHM DEVELOPMENT

A. MODEL DEVELOPMENT

In this section, we formulate the problem of routing UASs for IED interdiction as an integer program (IP) similar to the TOPTW. We refer to this integer program as the routing integer program (RIP).

1. RIP Formulation

We construct a complete graph where each node represents an IED interdiction mission or another mission and each directed arc represents transit times between nodes. Without loss of generality, we assume that all UASs start and end their routes at node 1. Figure 2 is a graphical example of a graph with ten nodes. We note that only arcs along a specific route are drawn. Since the graph is complete all nodes are connected to all other nodes, even if not illustrated. As indicated in Figure 2, there are various data associated with nodes and arcs. We will discuss that in detail after we present the formulation.

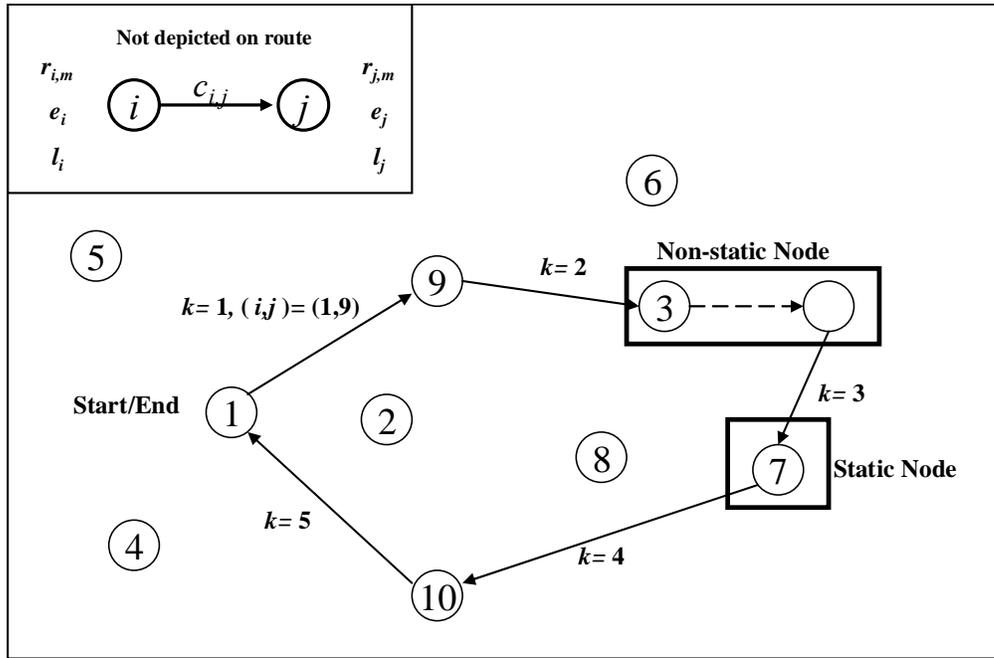


Figure 2 Example of graph of missions and UAS Route $\{1-9-3-7-10-1\}$.

A description of RIP follows:

Indices

i, j	Nodes, $i, j \in \{1, 2, \dots, I+1\}$.
m	UAS, $m \in \{1, 2, \dots, M\}$.
k	Position of an arc in the sequence of the route, $k \in \{1, 2, \dots, K\}$ where $K < I$.

Data

$r_{j,m}$	Reward for UAS m performing mission at node j .
s_j	Service time at node j .
e_j	Early time window for start service at node j .
l_j	Late time window of start service at node j .
$c_{i,j}$	Travel time from node i to node j given current wind conditions.
T_{max}	Maximum individual flight time for each UAS m .
T_{min}	Minimum individual flight time for each UAS m .
W_{max}	Maximum wait time allowed at a node.

Binary Variables

$X_{i,j,k,m}$	1 if arc (i, j) is in position k of the route for UAS m , 0 otherwise.
---------------	--

Non-negative Variables

$T_{k,m}$	Time when UAS m completes actions associated with position.
$W_{k,m}$	Wait time of UAS m at position k .

Mathematical Formulation of RIP

$$\max \sum_{i,j,k,m} r_{j,m} X_{i,j,k,m} \quad (1)$$

s.t.

$$\sum_j X_{1,j,1,m} = 1 \quad \forall m \quad (2)$$

$$\sum_{j,k,m} X_{i,j,k,m} \leq 1 \quad \forall i \in \{2, \dots, I\} \quad (3)$$

$$\sum_k X_{1,I+1,k,m} = 1 \quad \forall m \quad (4)$$

$$\sum_{i,j} X_{i,j,k,m} = 1 \quad \forall k, \forall m \quad (5)$$

$$\sum_{j,k} X_{1,j,k,m} = 2 \quad \forall m \quad (6)$$

$$\sum_i X_{i,I+1,K,m} = 1 \quad \forall m \quad (7)$$

$$\sum_i X_{i,j,k-1,m} - \sum_i X_{j,i,k,m} = 0 \quad \forall j, \forall k > 1, \forall m \quad (8)$$

$$T_{K,m} \leq T_{max} \quad \forall m \quad (9)$$

$$T_{K,m} \geq T_{min} \quad \forall m \quad (10)$$

$$\sum_{j:j \neq 1} (c_{1,j} + s_j) X_{1,j,1,m} + W_{1,m} = T_{1,m} \quad \forall m \quad (11)$$

$$\sum_{i,j} (c_{i,j} + s_j) X_{i,j,k,m} + T_{k-1,m} + W_{k,m} = T_{k,m} \quad \forall k > 1, \forall m \quad (12)$$

$$T_{k-1,m} + c_{i,j} X_{i,j,k,m} \geq e_j X_{i,j,k,m} - W_{k,m} \quad \forall i, \forall j, \forall k > 1, \forall m \quad (13)$$

$$c_{1,j} X_{1,j,1,m} \geq e_j X_{1,j,1,m} - W_{1,m} \quad \forall j, \forall m \quad (14)$$

$$T_{k-1,m} + c_{i,j} X_{i,j,k,m} \leq T_{max}(1 - X_{i,j,k,m}) + l_j X_{i,j,k,m} \quad \forall i, \forall j, \forall k > 1, \forall m \quad (15)$$

$$W_{k,m} \leq W_{max} \quad \forall m, \forall k \quad (16)$$

Equation (1) defines the objective function which represents the total reward collected during a given time limit T_{max} . Constraint (2) ensures that each UAS m begins its route from node 1 and that the arc $(1,j)$ is in the first position of the route. Constraint (3) ensures all nodes except node 1 and the dummy node are visited at most once. Constraint (4) ensures that the UAS returns to node 1 before it moves to the dummy node, which signifies that the route is complete. Constraint (5) ensures that there is one arc (i,j) per position k . Constraint (6) ensures that there are two outgoing arcs from the start node. This is a modeling constraint that ensures the last arc on the route is from node 1 to the dummy node, which facilitates a complete route that starts and ends at node 1. Only the first arc departing node 1 is actually on the route. Constraint (7) makes the last arc end at the dummy node. Constraint (8) is a balance of flow constraint, which ensures that if arc (i,j) is in the k th position of the route, then the arc in the $k+1$ position must be arc (j, ii) so a UAS cannot remain at the same node for consecutive positions. Constraint (9) ensures that each UAS returns to its node 1 no later than the time maximum and constraint (10) ensures that the route duration is greater than a time minimum. Constraints (11) and (12) are constraints for the time of the route at position k for each UAS m . Constraints (13-15) deal with the time windows of execution. Note that we both permit UAS to return home early with no penalty as long as the route duration is greater than T_{min} and allow for increased waiting times for each position k , see constraint (16).

2. Integer Program Model Discussion

Mission and IED nodes are represented by the indices i and j and there are two types of nodes. The first type of node called a “mission node,” represents a mission and the second type, called an “IED node” represents an IED interdiction mission at an IED hotspot. The actions of the UAS at a mission or IED node classify the node into one of two categories: static or non-static nodes as depicted graphically in Figure 2 and defined as follows:

- Non-static nodes represent the situation that a UAS starts a mission of either type at one location and is required to end it at a different location. Non-static nodes are represented by a point on a map for the start node with a one-way pointer to a dummy node respective of the end location.
- Static nodes are represented by static area locations that are recognized as a single point on a map.

In accordance with FM 3x90-6 (2006), each node should have a number and the following pieces of information:

- Primary objectives (including priority and collection emphasis)
- Time on target
- Latest time information of value (LTIOV) and earliest time information of value (ETIOV), which are time windows on when information is needed to make decisions

Each node has an assigned nonnegative, relative value assigned by the commander, which is based on the mission and the type of UAS used for the mission. We consider these assigned values as rewards $r_{j,m}$ which are obtained when UAS m completes the service respective to a mission or IED node j . Typically, mission node rewards are larger than IED node rewards. The UAS may fly over mission and IED nodes multiple times, but the reward at these nodes can only be collected once and only if the UAS completes the service. The objective equation is to maximize the sum of these rewards for a given flight time constraint T_{max} . Further, each mission and IED node has an individual time window of execution, which is represented by e_j for the early time of execution and by l_j for the late time of execution, and service time s_j for the service time of the activity. Specifically, these properties are defined as follows:

- Early Time: Each node has an early time of execution, which represents the earliest time a UAS can begin service at a node. If a UAS arrives to a node before this early time, then it must wait at the node until the early time before it can begin executing the task. Waiting at a node is a constraint which will be discussed in following paragraphs. Early time is designed to be consistent with the term ETIOV.
- Late Time: Each node has a late time of execution, which represents the latest time a UAS can begin executing its task at the respective node. If a UAS arrives after the Late Time, then it cannot carryout the service at the node. If the UAS arrives prior to the Late Time and after the Early Time,

it can begin service at the node immediately. Late time is reflective of the LTIOV where the late time window plus the time on target must be less than the LTIOV.

- Service Time: Specific to each node is a task the UAS must perform. This task has an associated planned execution time, for which we call the service time or as previously introduced time on target. Service times differ between mission and IED nodes. Mission nodes generally will have a much larger service time than IED nodes. UAS will remain at the node area until service time is completed.

The key element for each of these node characteristics is for the tactical planner to make a valid and meticulous planner's estimate of the service time at the node as it meets each node requirements: area of coverage, ground/air unit coordination, sensor, and weather/terrain.

The distance between each node is represented by the travel time between the nodes given a constant transit speed imputed by wind conditions. Therefore, $c_{i,j}$ represents the travel time between nodes i and j . An arc represents a directional vector from one node to an adjacent node. If a UAS visits node i then node j , we represent it as arc (i,j) . The position of arc (i,j) on the route for UAS m is defined by k . We let $T_{k,m}$ be the time of UAS m at position k of the route. $T_{k,m}$ represents the cumulative time from the node 1 through position k for UAS m . For example, $T_{k-1,m} = 25$. and the time it takes to move from node i to node j during flight segment k plus the service time of node j is 20, then $T_{k,m} = 45$. We complete the formulation with a binary decision variable which represents whether or not UAS m moves from node i to node j , executes the service at node j , and arc (i,j) is in the k^{th} position of the route given by $X_{i,j,k,m}$. Specific to this problem, we include a dummy node as the value $I+1$, which facilitates building routes with nodes less than the total number of nodes. This dummy node has zero reward and service time so there is no advantage in adding it to the route.

3. RIP Limitations

Terrain and weather both can severely restrict UAS operations. Restrictive terrain (e.g., dense vegetation, urban areas) in the mission or IED hotspot area can shield targets from UAS payloads. Steep terrain requires UASs to fly closer and at higher

angles to the target, as opposed to operating from a distant position observing from lower angles. The difficulty in obtaining a clear picture is represented in an increased mission length. Although travel times between nodes account for current wind conditions, RIP does not account for the uncertainty in weather conditions. Terrain and weather factors affect the time it takes a UAS to accomplish the mission and increased complexities in either terrain or weather result in increased times of execution.

RIP does not explicitly allow a UAS to visit a node more than once, however, actual UAS flight paths may fly over mission or IED nodes without executing the service. This is a relaxation in the model to reduce the complexity which we justify by the non-static nature of UAS operations. Although the UAS sensor may be fixed at a node, the UAS must be moving at an altitude above the node to remain airborne and is not stationary. Given multiple UASs and a condensed area of operations where mission and IED nodes are closely located, UASs will fly routes that require deconfliction between vehicles at certain nodes. This model assumes the tactical planner inspects the UAS routes by time to further validate deconfliction and therefore avoid collision.

B. HEURISTIC ALGORITHM

In this subsection, we present a heuristic algorithm for RIP that inputs mission and IED nodes, that builds a forward star data structure for each arc and an adjacency list for each node, then implements a column generation application that develops feasible routes using a depth first search strategy. Key pre-processing steps of this heuristic include calculating the travel times between nodes that account for wind effects and arranging the nodes in order of decreasing reward. By arranging nodes in this manner, we develop adjacency lists for each node that lead the routing tool to a type of greedy node selection, which routes to the highest reward nodes first. The feasible routes are outputted to an integer program that deconflicts the routes of multiple UASs.

1. VBA Pre-Processing

The first step is to input the mission and IED nodes. To account for the UAS turning radius of 1-1.5 km at loiter speed (Rathinam, Sengupta, & Darbha, 2006) nodes

must be no closer than 1 km to each other. If nodes are closer than this distance, we cluster them into one singular node. A sample output from this step is depicted below in Figure 3.

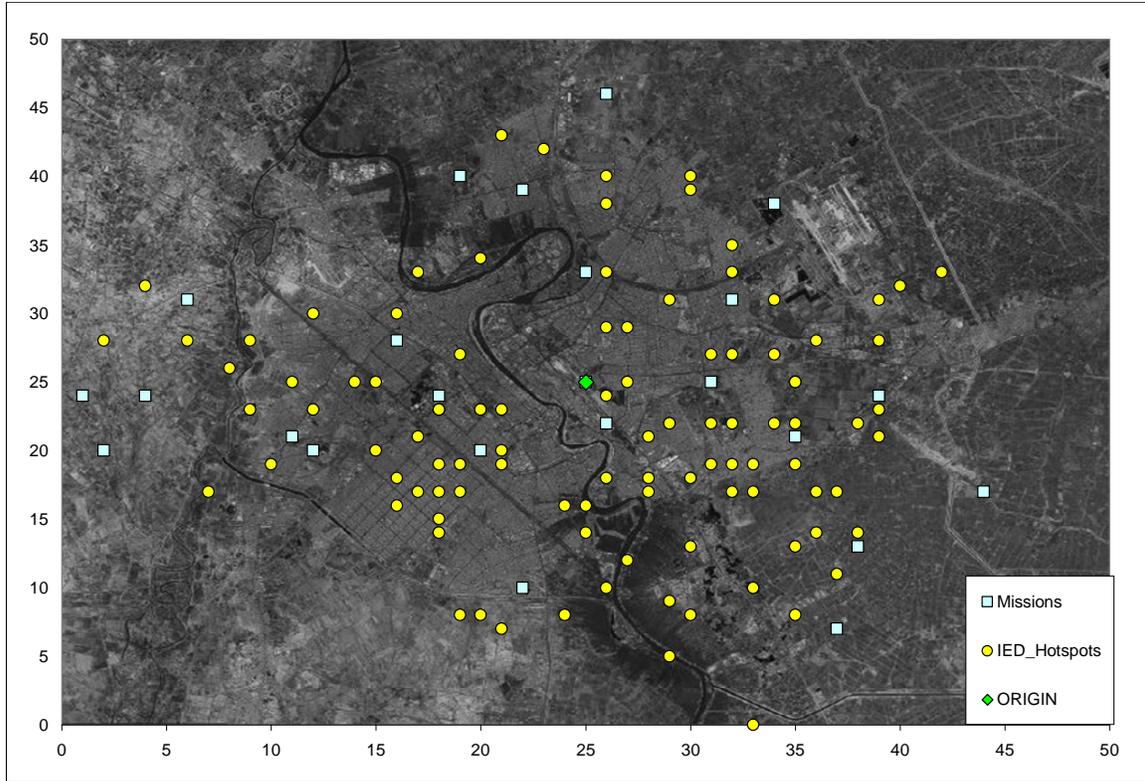


Figure 3 Graphical display of Mission and IED Nodes ($I=125$)

After nodes are inputted, the routing tool develops a time distance matrix consisting of distances c_{ij} . The distances are computed using simple vector calculation to determine the flight time between nodes based on the effects of wind, which means the time distance matrix is asymmetric. This matrix is based on the given UAS transit speed. This speed does not impact mission and IED node service times. Based on the speed and the wind, the time of travel between nodes is calculated. The tool next develops an adjacency list for each node inputted to the routing tool. A node's adjacency list is the ordered list of nodes a UAS can feasibly fly to from that node. This step involves a user interface option, which allows the user to limit the travel distance from one point to another. For example, if we set a maximum of 10km distance of travel, for the purpose

of developing feasible routes, the UAS cannot fly from node i to node j if the distance between the two nodes is greater than 10km. Options to further constrain each node's adjacency list exist. For example, the tool can reduce the adjacency list to represent the best eight missions and the best eight IED nodes for the UAS to travel based on the "Biggest Bang for the Buck," which could be determined by simply dividing node j 's reward by the time it takes to reach node j and then comparing the ratios. Options to further limit the number of nodes in an adjacency list directly impact the time it takes to solve the problem as the more nodes in an adjacency list, the greater number of feasible paths that will be generated. In either case, the adjacency list ordering supports a greedy type of search where the most valued nodes are added to the route first if feasible.

2. Feasible Route (Column) Generation

Routes are generated in this phase according to a depth first search process, which uses the forward star data structure and node adjacency lists to develop feasible routes for multiple combinations of nodes. The forward star data structure allows the route generation to occur without re-visiting previously developed feasible routes. This search process, which is similar to Kantor and Rosenwein (1992) is known to have difficulties finding optimal routes to problems with greater than 25 nodes because the depth first search process seeks to enumerate all feasible routes; therefore, this search method requires additional constraints to reduce the search space to a manageable size. In order to decrease the solution space, the following rules have been implemented, which are similar to those used by the ARC (2003):

- If a route is not feasible, then the same route with the addition of an additional node is not feasible either. If a route is infeasible it must be because it violates the time window constraints or the max flight time constraint or both.
- The total flight time of a route generated must lie between the user inputted minimum and maximum flight times. We included a minimum flight time, because we do not want to consider routes where the UAS returns early.

- Consider two different routes 1 and 2, where route 2 visits all the nodes on route 1 plus one more. If route 2 is feasible, then route 2 dominates route 1 and route 1 cannot be the optimal solution. In Figure 4 route 2 dominates route 1.

Route 1) 1-3-4-5-6
Route 2) 1-3-4-5-6-7

Figure 4 Example of Route Domination Principle.

These concepts facilitate building routes that are “complete,” i.e., the route duration is longer than T_{min} , but shorter than T_{max} , and prevents the tool from later having to consider incomplete routes that the UAS would not fly. The priority of our route building is to route the UAS through mission nodes and if time allows determine which IED nodes we can include in the route to eliminate unnecessary wait times at mission nodes. Therefore, in the next set of rules we introduce a sequence tracker, which keeps track of the mission node sequence for the route generated and compares it to the next route generated. If two sequential routes have the same mission sequence, this tracker is increased by one. If the two sequences are different, then the tracker is updated with the latest mission sequence and its counter returned to 0. For example, mission nodes are numbered 1-10 with decreasing rewards, and IED nodes numbered 11 and greater with identical rewards. The following data in Figure 5 is an example of routes produced by this routing tool:

Route 1) 1-2-3-4-19-25-26-35-9-39-40	The mission sequence is 1-2-3-4-9
Route 2) 1-2-3-4-19-25-26-35-9-39-41	The mission sequence is 1-2-3-4-9
Route 3) 1-2-3-4-19-25-26-35-9-39-42	The mission sequence is 1-2-3-4-9
Route 4) 1-2-3-4-19-25-26-35-9-39-43	The mission sequence is 1-2-3-4-9

Figure 5 Mission Sequence Example.

Each of the four routes generated in Figure 5 are different; however, each route has the same mission sequence. Also, note that because we have identical rewards for each IED hotspots that each of these routes has the same reward. Thus, there has been no improvement in total route reward over the last three iterations. Given the depth first search procedure, we will continue to generate similar paths for a number of iterations. The sequence tracker is linked to a user input parameter that dictates how many consecutive paths can be generated with the same mission sequence. This enables the user to identify the amount of feasible path exploration he wants to allow. Given the output of routes in this example, the route building process is at a point where the continued time invested in exploring feasible options may not be worth the payoff in reward obtained from the continued work. Recall that each node's adjacency list is ordered in terms of priority, so the depth first search process seeks to route through mission nodes first which means that the feasible nodes with the highest reward will most likely be routed within the first attempts of the route building process at this point in the development. The larger the value for the mission sequence iteration tracker inputted by the user, the longer the process continues to generate similar routes increasing computation time, but not guaranteeing better solutions. Nevertheless, when the sequence tracker reaches the user-defined sequence maximum, the user has another input decision to make. In figure 5, if we had set our mission sequence counter to four, the tool will now execute one of two courses of action. It will either go back to the second to last mission node and begin generating additional routes, which would mean that the mission sequence would be set to '1-2-3-4' so the route would return to '1-2-3' and proceed to the next node in '3's' adjacency list that it has yet to visit, whichever node is after '4' or the second option is for the user to allow for a random step back, which would choose one of the previous four nodes to return to with equal probability and proceed to the next node in its adjacency list.

Routes generated with similar mission sequences are compared and sorted against each other at the last step of the route generation phase. Once the feasible routes are generated, the tool then sorts the paths by the mission sequence. As we noted earlier, to account for the re-appearance of similar mission sequences, the feasible routes are sorted

by their mission sequence, then by the route total reward, and finally by the cumulative time the UAS is simply flying and not performing a mission on the route. The tool then by a similar principle used in our route development implements a domination principle heuristic, where the route with the highest reward for a specific mission sequence dominates any other route with the same mission sequence, but less total reward. Given feasibility as previously described the sequence tracker uses the following rules:

- Consider two different routes 1 and 2, where route 1 visits all the same mission nodes as route 2 but in a different order. Both routes are considered equal and only 1 of them will be recorded.
- Consider two different routes 1 and 2, where route 1 visits all the same mission nodes as route 2 but different IED hotspots. Both routes are considered equal and only 1 of them will be recorded.
- Consider a combination of the previous two rules, Routes 1 and 2 have the same mission sequence, but visit them in a different order and Route 2 visits one or more IED hotspots. Route 2 dominates route 1.

The results of the route generation process produce a set of feasible routes to be deconflicted for the number of UASs by a master routing integer program that we refer to as RIP-M.

3. RIP-M for Route Selection

Based on the number of feasible paths generated, we now need to pick routes, one for each UAS. We use the integer program below to do so.

Indices

- i Nodes, $i \in \{1, 2, \dots, I\}$.
- y Feasible route, $y \in \{1, 2, \dots, Y\}$.

Data

- $r_{y,m}$ Reward for UAS m on route y .
- $a_{i,y}$ 1 if route y , contains node i .
- m number of routes, corresponds to number of available UASs.

Binary Variables

$X_{y,m}$ 1 if UAS m chooses route y , 0 otherwise.

Mathematical Formulation

$$\max \sum_{y,m} r_{y,m} X_{y,m} \quad (17)$$

s.t.

$$\sum_{y,m} a_{i,y} X_{y,m} \leq 1 \quad \forall i \quad (18)$$

$$\sum_y X_{y,m} \leq m \quad (19)$$

The objective function (17) of this integer linear program is to maximize the total reward by selecting up to m routes. The (18) constraint ensures that each node is visited no more than once, which prevents multiple visits to the same node. Constraint (19) ensures that up to m de-conflicted routes are chosen.

Given the unpredictability of the number of feasible routes the route generation phase produces, we expect two types of implementations of the routing tool. If an efficient IP solver like CPLEX is available, solve RIP-M as it stands. If no such solver is available and the Excel solver needs to be used, the routing tool carries out a greedy filter process to reduce the total number of routes to 200, which complies with the Excel solver binary variable capacity constraints, if there were more than 200. In either case, the heuristic algorithm provides a feasible set of routes and a lower bound of the optimal value as not all routes are considered.

THIS PAGE INTENTIONALLY LEFT BLANK

IV. COMPUTATIONAL STUDY

A. TEST DATA

We consider a situation that may resemble those in the Iraqi theater of operations, which is consistent with both the author's operational experience in Iraq and our sponsor the Joint IED Defeat Organization's (JIEDDO) input. Suppose that a BCT operates from a forward operating base (FOB), which is a static location centrally located in a 50km by 50km area of operations. This is the backdrop for separate scenarios consisting of multiple data sets of up to 125 total nodes. Given the unclassified nature of this research and attempt to model operations in the Iraqi theater, the following planning considerations were made:

- BCT FOB and UAS launch and recovery site are co-located.
- For each data set, 20% of the nodes were classified as mission nodes, and the remaining 80% classified as IED Nodes. IED and Mission nodes were spaced throughout the defined AO based on a comprehensive terrain analysis to simulate actual mission areas and potential IED hotspots. The nodes were however spaced throughout the AO to prevent clustering. This represents a more difficult routing situation for UASs as clustered mission and IED nodes typically are easier to cover.
- Mission node rewards were represented on a scale from 30 decreasing by one for each mission node in the data set. For example, if there were two missions, then the two mission nodes would have rewards 30 and 29 respectively. Alternately, IED nodes were all given the same reward of 1. Options to individually weight IED nodes according to command priority exist, but were not tested.
- Data sets of 10, 25, 50, 75, 100, and 125 were established individually and results for these data sets should not be perceived as monotonically increasing as the number of nodes increases.

Only two parameters were common throughout the entire experimentation-wind speed at 0 knots, UAS speed at 110 knots. The remainder of the parameters was varied based on the number of nodes in the data set and represented in Table 2.

Total Nodes	Flight Time Min	Flight Time Max	Wait Time Max	Service Time of Mission Node	Service Time of IED Node	Length of Time Window interval ($l_j - e_j$)
10	60	80	10	20	10	10
25	60	80	10	20	10	15
50	110	120	5	30	10	30
75	160	180	5	30	10	30
100	270	300	5	40	10	30
125	270	300	5	30 or 60	10	30

Table 2 Parameter settings for test data. All times in minutes.

It is important for the planner to have a visual picture of the time windows of execution for each node as it further validates the necessity for a routing tool for non-trivial problems. Figure 6 is an example of the 75 node data set time windows. The IED node time windows are depicted by the straight lines throughout the timeline as in this test we allowed IED nodes to be visited at any time during the route, which facilitates an increased number of feasible paths. The mission nodes for this data set are represented by the short lines respective of the time window of execution for each node. From this graph it is apparent that there is no particular order to the time windows, only they are randomly distributed throughout the time interval of operation. Further, it is not apparent from this chart, which combination of nodes will produce the optimal solution or what additional IED nodes are feasible to visit.

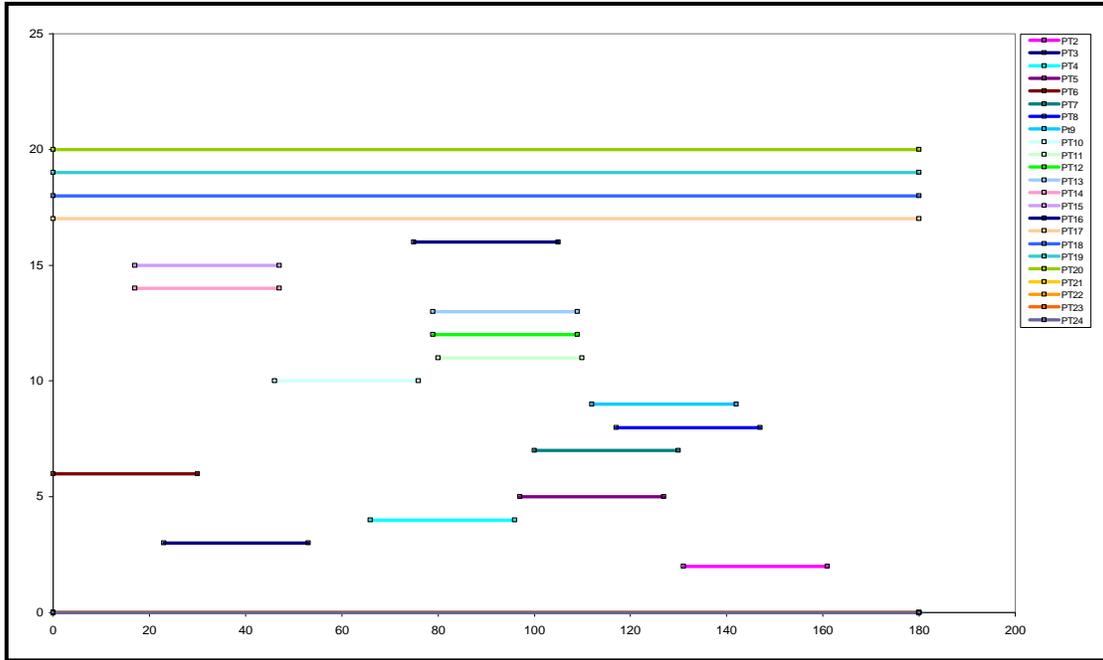


Figure 6 Time Windows of Mission and IED Nodes for test data set ($I=75$).

B. RESULTS

With the experiment scenario established, we implemented RIP in GAMS and found optimal or near-optimal solutions in most test instances using the default settings in the CPLEX version 11 solver. We also implemented our heuristic in Microsoft Excel VBA and found optimal or near-optimal solutions in all test instances. The following results were obtained by running both solutions using solvers on a Dell Precision PWS690 Intel® Xeon™ CPU 3.37GHz processor, with 3.00 GB of RAM. In some instances CPLEX failed to provide solutions for RIP when solving for total nodes greater than 75. In an effort to identify the quality of solutions of the heuristic, we also compare it with the linear programming relaxation of RIP, which we call RLP. We did this for all data sets to establish an upper bound and baseline of comparison with the heuristic when a RIP solution was not attainable. Computation times for RIP and RLP are not represented in any tables. In nearly all instances of solving RIP the heuristic was faster and in some instances we did not get an optimal solution. On the other hand, RLP is used

for establishing an upper bound to the problem not for presenting a realistic computation scenario so time is not important. The goal of the heuristic is to have computation times less than five minutes, which we consider a reasonable computation time to be of use to a tactical planner.

Table 3 represents the solutions when we solved the UAS routing problem for $m=1$ implementing RIP using a CPLEX MIP solver, our routing tool heuristic (Heuristic), and RLP using a CPLEX LP solver. In this table and subsequent tables, x denotes instances where an optimal solution could not be obtained either due to lack of memory or computation run times greater than four hours.

Total Nodes	RIP Optimal Value	Heuristic Solution Value	Heuristic Computation Time (min:sec)	RLP Upper Bound
10	33	33	0:05	60
25	85	85	0:07	97
50	88	88	0:12	104
75	141	141	0:30	183
100	x	126	0:30	191
125	x	160	3:20	203

Table 3 Optimal Values of RIP, Heuristic, and RLP.

Given the combinatorial nature of this problem, as the number of nodes increases the number of feasible solutions increases exponentially, therefore, for smaller problems of node size greater than 50, RIP achieves optimal results fairly quickly; however, as the number of nodes increases RIP computing time greatly increases and for the problems of over 100 nodes our computer ran out of memory. For all problems RLP solutions represent an upper bound to the problem and in all instances the heuristic is within 65% of RLP, which given ratio comparisons to the optimal solutions of the smaller number of node problems, presents a strong argument that we have achieved high-quality solutions for problems of total nodes of more than 100 for which we have no optimal solution available for comparison. Figure 7 represents the graph of the results of Table 3.

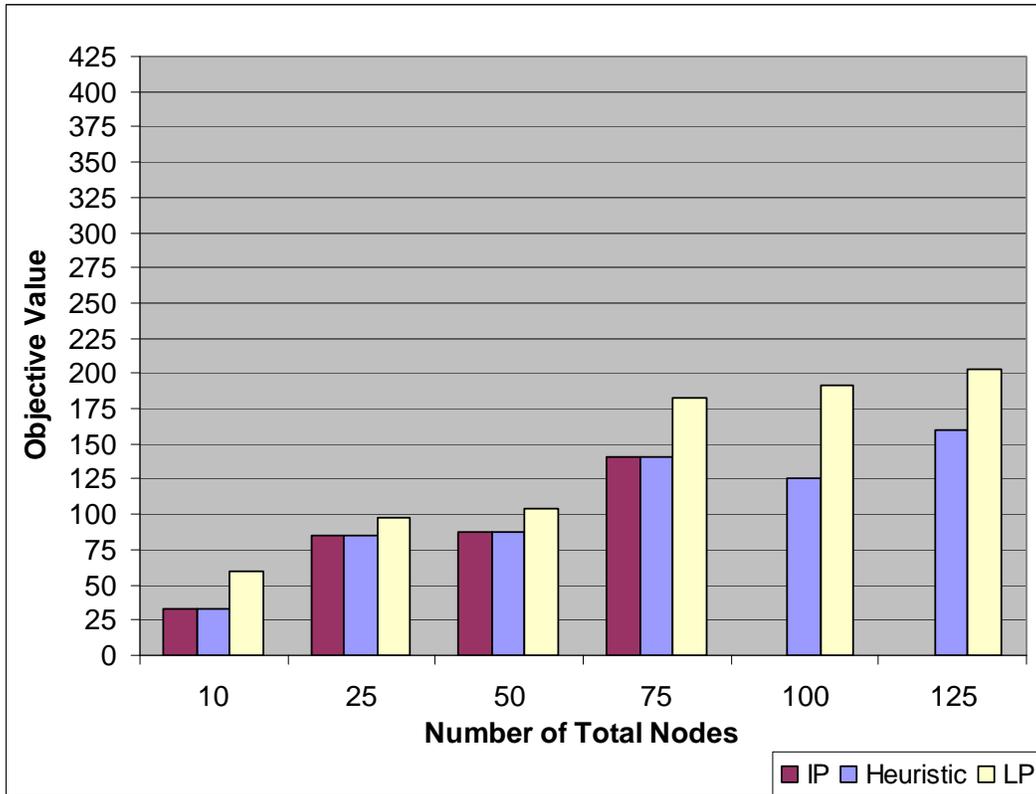


Figure 7 Graph of Optimal Values of RIP, Heuristic, and RLP.

When we solved both the heuristic and RIP for UAS $m=2$, we achieved the results in Table 4. RIP failed to provide solutions for problems with number of nodes greater than 25. Figure 8 represents the graph of the results of Table 4.

Total Nodes	RIP Optimal Value	Heuristic Solution Value	Heuristic Computation Time	RLP Upper Bound
10	63	63	0:05	65
25	143	142	0:21	146
50	x	143	0:49	157
75	x	230	2:40	338
100	x	232	2:00	333
125	x	260	3:20	319

Table 4 Optimal Values of RIP, Heuristic, and RLP.

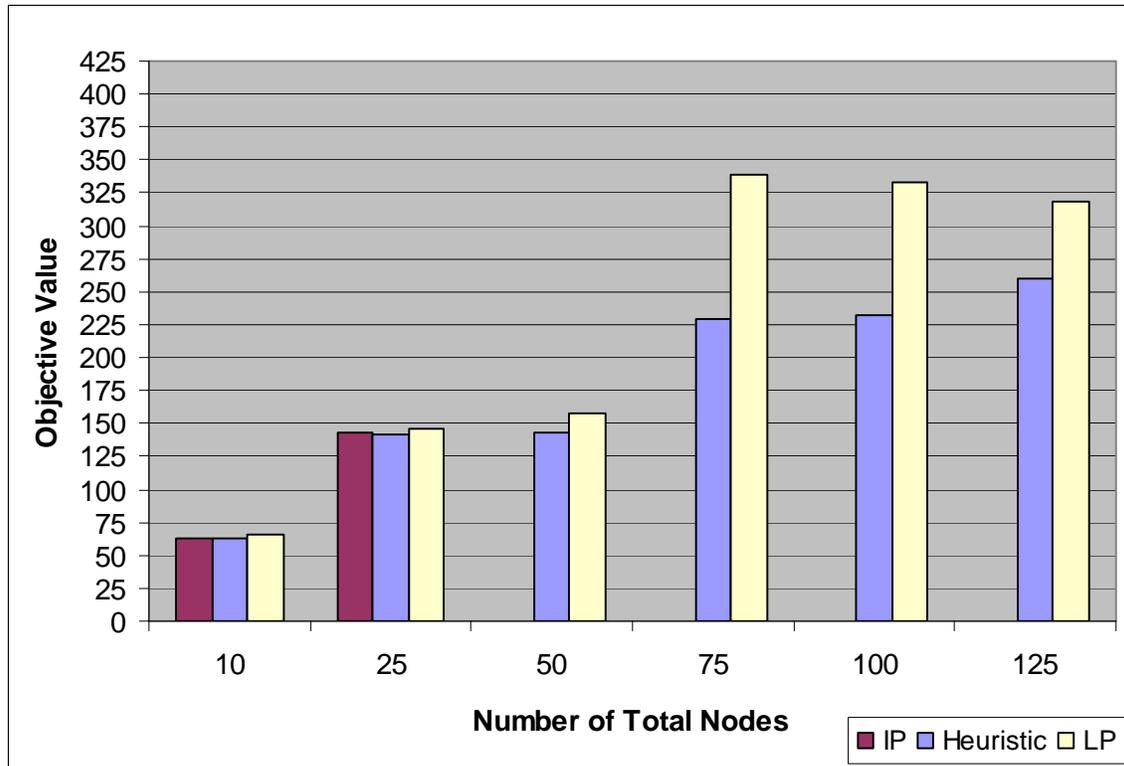


Figure 8 Graph of Optimal Values of RIP, Heuristic, and RLP

When we solved both the heuristic and RIP for UAS $m=3$, we achieved the results in Table 5. RIP failed to provide optimal solutions for problems with more than 25 nodes. Figure 9 represents the graph of the results of Table 5.

Total Nodes	RIP Optimal Value	Heuristic Solution Value	Heuristic Computation Time	RLP Upper Bound
25	149	148	0:21	154
50	x	177	0:49	202
75	x	295	2:40	356
100	x	325	2:00	412
125	x	323	3:20	x

Table 5 Optimal Values of RIP, Heuristic, and RLP.

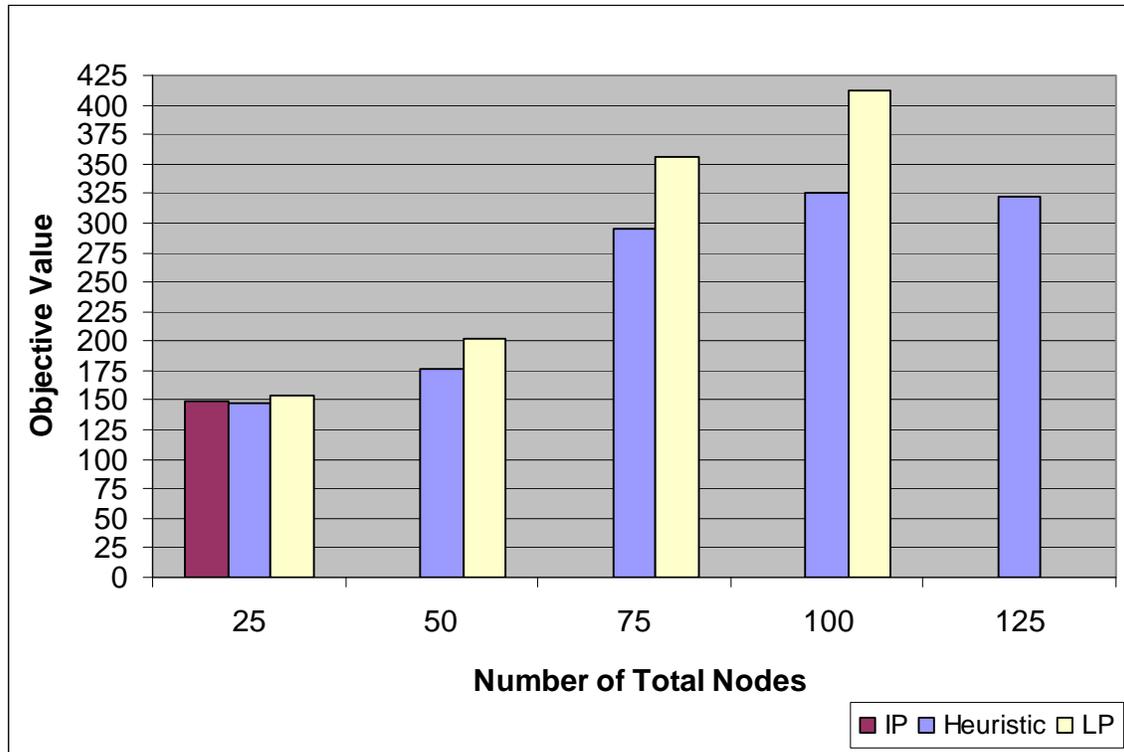


Figure 9 Graph of Optimal Values of RIP, Heuristic, and RLP.

For all data sets and values of m , the heuristic provided routes quickly and for the smaller problems it achieved optimal solutions. We recognize that by weighting IED nodes equally, there are multiple optimal solutions, but given the problem description, efficient routing procedures are desired and minor reward diversification between IED nodes will not significantly change the results. An example of the output graphically on an area of operations is depicted in Figure 10. In this figure, note the nodes with a black dot in the center of the circle vice nodes without black dots in the center of the circle. The nodes with a black dot in the center are on a route corresponding to the legend of the graph. UAS routes fly from node to node via the shortest path route; therefore, it is possible for a route to fly over a node in which it does not conduct the mission and therefore does not collect the reward.

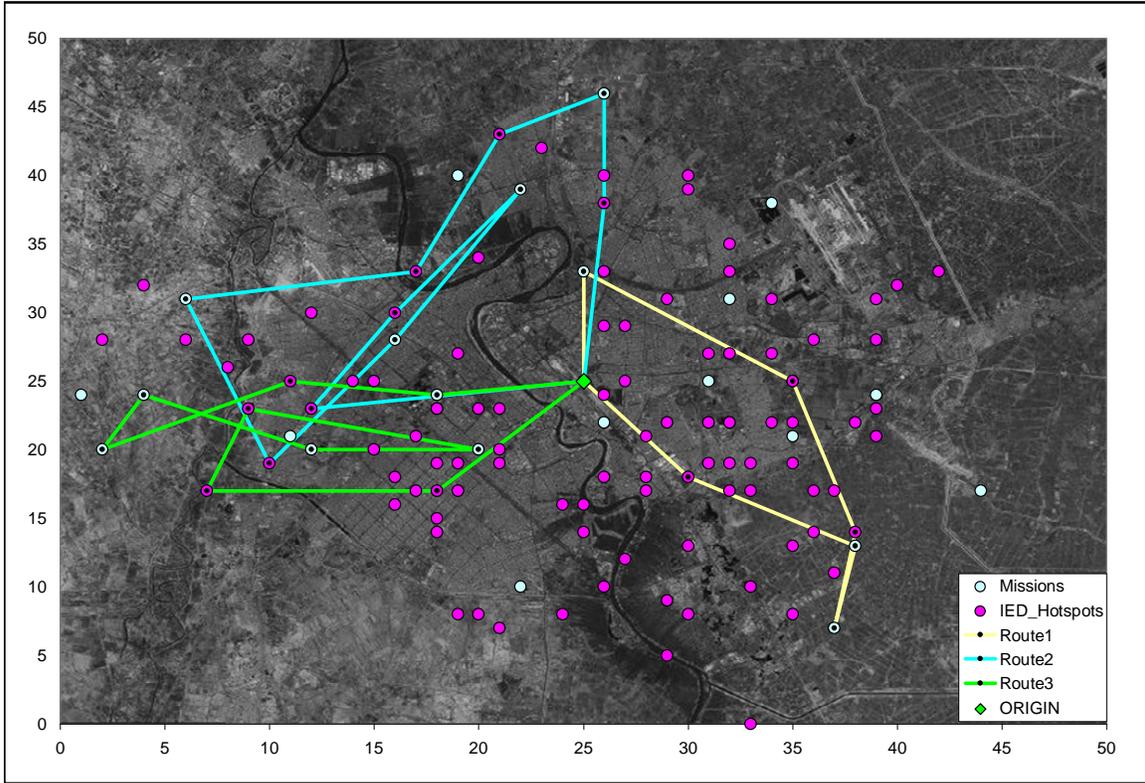


Figure 10 Example Graphical Output from Routing Tool (Test Data Set $I=125$).

C. SENSITIVITY ANALYSIS

We now examine the sensitivity of the routing tool for changes in the parameters. As previously introduced, this routing tool implements a mission sequence tracker that limits the feasible path exploration by stopping it at a user-defined iteration number as well as a randomization option that supports a uniformly distributed jump back to a mission node in the mission sequence vice the previous mission node. Based on multiple test runs, these two factors contributed not only to the solution quality, but the computational run time as well. In an effort to help the user identify the settings which produce the optimal results, the following tests were executed and depicted in Table 6, where \mathbf{x} denotes the instance not obtaining a solution due to computational run times greater than 20 minutes:

- Sequence Tracker Settings (ST) were set at 50, 5,000, and 10,000
- Random Sequencing (Random Mode) was either turned on or off for each Sequence Tracker Setting

	ST=50		ST=5000		ST=10000	
Random Mode	Yes	No	Yes	No	Yes	No
Total Nodes						
10	33	33	33	33	33	33
25	85	85	85	85	85	85
50	88	88	88	88	88	88
75	141	141	141	x	141	x
100	125	125	125	x	125	x
125	146	160	160	x	160	x

Table 6 Results from Setting Sensitivity Analysis.

From this study, we note that the random sequencing is essential for larger problems in keeping the number of feasible paths to a manageable size and reducing computation time to less than 20 minutes. Specifically run times for problems of nodes greater than 50 were all comparable and all less than four minutes; however, run times for the larger node problems displayed tremendous ranges in computation time as evident by results in Table 7.

	ST=50		ST=5000		ST=10000	
Random	Yes	No	Yes	No	Yes	No
Total Nodes						
75	0:03	0:12	2:40	>20:00	4:13	>20:00
100	0:30	2:00	5:00	>20:00	13:00	>20:00
125	0:35	3:20	7:45	>20:00	19:30	>20:00

Table 7 Time Results from Setting Sensitivity Analysis.

It's clear from this table that the heuristic takes too long to solve for ST values greater than 5000 although that time is somewhat reduced by implementing the random sequence tracker. For large problems of this nature, we conclude that the running the heuristic on the lowest settings i.e. ST equals 50 and random sequencing, is the most efficient setting. It may not produce the optimal solution as evident by the 146 value vs.

160 optimal value displayed in Table 6; however, this value is still within 9% of the optimal value and the solution was produced in just over 30 seconds. No doubt, the random sequencing option plus the reduced tracking sequencing number produce the fastest results, but are slightly bettered by the solution quality obtained when the random sequencing option is turned off for low values of the sequencing tracker. Our recommendation would be to run the tool at the lowest settings first as it produces a very quick solution. Then run the tool for the next setting with the random off and compare. We conclude that based on our tests, there is no improvement in solution quality by increasing the sequence tracker greater than 50.

This routing tool takes advantage of knowing that 20% of the targets are mission nodes. This naturally reduces the solution space because this routing tool recognizes that 80% of the nodes are not quite as important as the mission nodes. The assumption that mission nodes are more important is valid, and if an IED node is as important then it simply becomes a mission node, which is a fairly simple conversion.

V. CONCLUSIONS AND FUTURE WORK

A. CONCLUSIONS

This thesis develops a routing tool for Unmanned Aerial Systems (UAS) with the task of interdicting Improvised Explosive Devices (IED). It is a planning tool for the tactical planner that is implementable on any Army computer and produces UAS routes designed for a BCT to maximize the utilization of a UAS according to the commander's priority for UAS missions. This routing tool does not rely exclusively on IED prediction models that require precise detail or a specific formatted input; rather it can also handle input in the form of a tactical planner's assessment and still attain high-quality solutions. Tactical level planners can use this routing tool to better prepare the reconnaissance and surveillance plan, course of action development, or even to re-route UASs from off-routed locations due to changes in initial conditions or to targets of opportunity. This routing tool does not trivialize the time or resources needed to execute an IED interdiction or push an IED interdiction mission over the commander's priority for other missions; rather it provides minimal input parameters and constraints to model difficult decisions in priority routing and seeks to route to IED hotspots when compatible to the route.

B. FUTURE WORK

1. Route Generation

Developing routes that are robust and maximize the utilization of the UAS flight time can be done in many different ways. Our routing tool builds routes one at a time and then attempts to deconflict these routes via an integer program. This works for a smaller number of UASs ($m < 3$). An alternative route building algorithm may consider building routes in parallel rather than sequentially, which may facilitate deconfliction of routes for a greater number of UASs ($m > 4$). Potential improvements to our routing tool that would facilitate this are directly related to the sequence tracker. Sequence tracking

may involve the implementation of an alternate data storage structure that better supports the construction of a greater number of routes simultaneously. Nevertheless, scenarios where BCTs and units have to fly more than four UASs exist.

2. Dynamic Updates

Targets of opportunity and changes from initial UAS routes will occur. Determining the validity and necessity of these changes to curtail the current UAS route to respond to these situations is the job of the commander. This routing tool does not take these changes as immediate inputs and produce courses of action for the commander, rather if the commander wants to implement a course of action to respond to these changes our routing tool will facilitate re-routing the UAS. A routing tool that can handle dynamic updates and provide courses of actions through alternate route recommendations to the commander is important and we consider it the next step to this routing tool. No doubt dynamic events occur frequently, responding to these events timely is essential. The way we envision the dynamic update occurring is that given the new target or new condition, the routing tool nearly instantly re-routes the UAS from the current point through the new target then back to the end node maximizing utilization.

3. UAS Routing Tactics

The concept of routing UASs through IED nodes for the purpose of interdiction to maximize UAS utilization is one that needs to be accepted and adopted by the commander as it will require changes to the standard operating procedures of the unit specifically UAS coordination with ground units. Units must be aware of the flight path of the UAS and be prepared to respond to potential IED incidents. Reaction time becomes paramount as the UASs we consider cannot hold ground or close with and destroy the enemy. Any delay in response may facilitate the IED being initiated. Current tactics suggest that using UASs for interdiction is worthy, but tactics evolve constantly as the threat adapts to them. It should be expected that any planning tool will require additional testing in that AO to fine tune parameters and settings so that routes are produced that meet the commander's intent, nevertheless, a routing tool that can further increase the flexibility of ours in modeling UAS routing will be of definite use to the commander.

LIST OF REFERENCES

- American Red Cross. (2003). Vehicle routing with time windows and time-dependent rewards: A problem for the American Red Cross. *Manufacturing & Service Operations Management*, 5(1).
- Archetti, C., Hertz, A., & Speranza, M. G. (2007). Meta-heuristics for the team orienteering problem. *Journal of Heuristics*, 13, 49-76.
- Boussier, S., Feillet, D., & Gendreau, M. (2007). An exact algorithm for team orienteering problems. *4OR Quarterly Journal of the Belgian, French, and Italian Operations Research Societies*, 5(3), 211-230.
- Butt, S. E., & Ryan, D. M. (1999). An optimal solution procedure for the multiple tour maximum collection problem using column generation. *Computers & Operations Research*, 26, 427-441.
- Casualty Count. (May 14, 2008). *IED fatalities OIF*. Retrieved May 14, 2008, from www.icasualty.org.
- Chao, I., Golden, B. L., & Wasil, E. A. (1996). A fast and effective heuristic for the orienteering problem. *European Journal of Operational Research*, 88, 475-488.
- Chiang, W., & Russell, R. A. (1997). A reactive tabu search metaheuristic for the vehicle routing problem with time windows. *INFORMS Journal on Computing*, 9(4), 417-428.
- Defense Data. (November 6, 2007). *Shadow 200 TUAS proves mettle in heavy Iraq use*. Retrieved May 14, 2008, from <http://defence-data.com/current/page38539.htm>.
- Desrosiers, J., Desrochers, M., & Soumis, F. (1984). Routing with time windows by column generation. *Networks*, 14, 545-565.
- FAS Intelligence Resource Program. (May 4, 2000). *Tactical unmanned aerial vehicle (TUAS) close-range TUAS*. Retrieved May 14, 2008, from <http://www.fas.org/irp/program/collect/cr-TUAS.htm>.
- Harder, R. W., Hill, R. R., & Moore, J. T. (2004). A java universal vehicle router for routing unmanned aerial vehicles. *International Transactions in Operational Research*, 11, 259-275.
- Headquarters Department of the Army. (August 2006). FM 3x04-15: Multi-service techniques for the tactical employment of unmanned aerial systems (UAS).
- Headquarters Department of the Army. (August 2006). FM 3x90-6: The BCT Combat Team (BCT).

- Holland, JH. (1975). *Adaptation in natural and artificial systems*. Ann Arbor, MI. University of Michigan Press.
- JIEDDO. (December 6, 2007). *JIEDDO Homepage*. Retrieved December 6, 2007, from www.jieddo.dod.mil.
- Kantor, M. G., & Rosenwein, M. B. (1992). The orienteering problem with time windows. *Journal of Operations Research*, 43(6), 629-635.
- Moser, H. D. (1990). *Scheduling and routing tactical aerial reconnaissance vehicles*. (Masters, Naval Postgraduate School).
- Office of the Secretary of Defense. (July 2005). *Unmanned aircraft systems roadmap 2005-2030*. Retrieved May 14, 2008, from <http://www.acq.osd.mil/usd/Roadmap%20Final2.pdf>.
- O'Rourke, K. P., Bailey, T. G., Hill, R. R., & Carlton, W. B. (2001). Dynamic routing of Unmanned aerial vehicles using reactive tabu search. *Military Operations Research*, 6(1), 5-30.
- Osman, IH. (1993). Simulated annealing and Tabu search algorithms for the vehicle routing problem. *Annual Operations Research*, 41, 777-786.
- Ramesh, R., & Brown, K. M. (1991). An efficient four-phase heuristic for the generalized orienteering problem. *Computers & Operations Research*, 18(2), 151-165.
- Rathinam, S., Sengupta, R., & Darbha S. (2006). A resource allocation algorithm for multi-vehicle systems with non-holonomic constraints. Retrieved May 29, 2008, from <http://path.berkeley.edu/~raja/tenureCaseMaterials/my-writing/submitted/>
- Riese, S. R. (2006). Templating an adaptive threat: Spatial forecasting in operations Enduring Freedom and Iraqi Freedom. *Engineer Magazine*, (January-March), 42-43.
- Royset, J., & Reber, D. (2008). *Optimized Routing of Unmanned Aerial Systems for the Interdiction of Improvised Explosive Devices*. Retrieved May 29, 2008, from <http://faculty.nps.edu/joroyset/roysetpa.htm>.
- Shetty, V. K., Sudit, M., & Nagi, R. (2008). Priority-based assignment and routing of a fleet of unmanned combat aerial vehicles. *Computers & Operations Research*, 35, 1813-1828.
- Solomon, M. M. (1987). Algorithms for the vehicle routing and scheduling problems with time windows. *Operations Research*, 35(2), 254-264.

- Tan, K. C., Lee, L. H., Zhu, Q. L., & Ou, K. (2001). Heuristic methods for vehicle routing problems with time windows. *Artificial Intelligence Engineering, 15*, 281-293.
- Tang, H., & Miller-Hooks, E. (2005). A tabu search heuristic for the team orienteering problem. *Computers & Operations Research, 32*, 1379-1407.
- Troy, P. (2008, January 31). Telephone interview.

THIS PAGE INTENTIONALLY LEFT BLANK

INITIAL DISTRIBUTION LIST

1. Defense Technical Information Center
Fort Belvoir, Virginia
2. Dudley Knox Library
Naval Postgraduate School
Monterey, California
3. John Van Hise
Naval Postgraduate School
Monterey, California
4. Elizabeth Mitchell
Naval Postgraduate School
5. Captain Doug Otte
Naval Postgraduate School
Monterey, California
6. Debbie Sandoval
Naval Postgraduate School
Monterey, California