



Calhoun: The NPS Institutional Archive
DSpace Repository

NPS Scholarship

Publications

2009

Optimized Routing of Unmanned Aerial Systems for the Interdiction of Improvised Explosive Devices

Royset, J.O.; Reber, D.N.

J.O. Royset and D.N. Reber, 2009, "Optimized Routing of Unmanned Aerial Systems for the Interdiction of Improvised Explosive Devices," *Military Operations Research*, Vol. 14, No. 4, pp. 5-19.

<https://hdl.handle.net/10945/38228>

This publication is a work of the U.S. Government as defined in Title 17, United States Code, Section 101. Copyright protection is not available for this work in the United States.

Downloaded from NPS Archive: Calhoun



Calhoun is the Naval Postgraduate School's public access digital repository for research materials and institutional publications created by the NPS community. Calhoun is named for Professor of Mathematics Guy K. Calhoun, NPS's first appointed -- and published -- scholarly author.

Dudley Knox Library / Naval Postgraduate School
411 Dyer Road / 1 University Circle
Monterey, California USA 93943

<http://www.nps.edu/library>

Optimized Routing of Unmanned Aerial Systems for the Interdiction of Improvised Explosive Devices

Johannes O. Royset, Assistant Professor
Operations Research Department
Naval Postgraduate School
joroyset@nps.edu

Daniel N. Reber, Major, USMC
Mission Area Analysis Branch, Operations Analysis Division
Marine Corps Combat Development Command
daniel.n.reber@usmc.mil

ABSTRACT

The paper describes an optimization-based tool for selecting routes that will best employ unmanned aerial systems (UASs) for the purpose of detecting improvised explosive devices (IEDs) or related activity. The routing tool uses preprocessing procedures, an integer linear program, and an IED prediction model to direct UASs to sectors of the area of operations with high IED activity, while accounting for factors such as winds, aircraft de-confliction, and blue force activity. Initial evaluation of the routing tool through field experiments with actual UASs suggests that the tool produces realistic routes, which can be flown in the allocated amount of time, even under windy conditions.

Application Areas: Modeling, simulation, and wargaming; Computing advances in military OR.

OR method Area: Integer programming.

INTRODUCTION

Improvised explosive devices (IEDs) can be effective weapons for insurgents targeting conventional military and security forces, see, e.g., the conflicts in Iraq and Afghanistan (ICCC 2008). While the U.S. military has taken several steps towards reducing the effectiveness of IEDs, including up-armoring vehicles and fielding jammers to protect against radio-controlled

IEDs, a systematic and optimized employment of unmanned aerial systems (UASs) has not been implemented. In fact, most IED detections in Iraq have been made visually by soldiers and marines on the ground (Chisholm 2005). Based on the second author's experience while serving with U.S. Marine Unmanned Aerial Vehicle Squadron 2 and deployed to Al Taqaddum Airbase in Iraq during 2004 and again in 2005, UASs not tasked with other missions are currently arbitrarily sweeping roads for any unusual activity or other signs which might lead to detection of IEDs. This situation has been recognized by General Ronald Keys, former commander of the Air Force's Air Combat Command, who states that airborne assets spend much time flying up and down streets that no coalition forces will be on for another 12 hours and also compares the area of operations (AO) in Iraq to a junkyard, with too many false positive detections of IEDs (Fabey 2007).

In this paper, we study UASs searching for insurgents placing IEDs as well as IEDs that are already in place. We refer to both the detection of an insurgent placing an IED and the detection of an emplaced IED as an *IED interdiction*. We develop an integer-programming-based routing tool, referred to as the IED Search Optimization Model (ISOM) that maximizes the expected number of IED interdictions by directing UASs to sectors of the AO with high IED activity. The UASs are coordinated with blue force operations so that search in a sector takes place before, and not after, blue force activity in the sector. ISOM determines time of take-off for the UASs, enforces airspace de-confliction, accounts for wind, and limits the flight time for each UAS to less than its endurance.

While we recognize that a surveillance strategy based on UASs directed by ISOM is just one component of an array of tools, technologies, and tactics that can reduce the effectiveness of IEDs, we conjecture that ISOM may have three contributions. First, compared to manual

planning, ISOM generates routes with higher expected number of IED interdictions. Second, in the same way as police patrols deter crime in a neighborhood, ISOM will help deter IED-related activities by developing effective UAS patrol routes. Third, ISOM will automate the UAS route selection process, which today is carried out manually. While we are currently unable to completely verify these contributions, preliminary field experiments of ISOM (see the last section) using actual UASs show that the tool generates operationally feasible routes with high expected numbers of IED interdictions.

ISOM uses output from IED prediction models to quantify the expected number of IED interdictions associated with searching various sectors of the AO and solves an integer linear program to determine optimized routes for a fleet of heterogeneous UASs. We consider one Air Tasking Order cycle, i.e., one sortie per UAS involved, and envision ISOM being used once or twice a day to determine the next sortie for each available UAS. In this paper, we do not address dynamic re-tasking of UASs, low-level control of UASs, and factors affecting UAS availability such as maintenance and other logistical concerns. We accept that vehicle-borne IEDs and associated activities are not likely to be detected by UAS-mounted sensors and, hence, do not consider such IEDs.

This paper continues in the next section with background on IED prediction models, relevant sensors, and route optimization models. The subsequent sections describe the combat situation and defines the problem, present the routing tool ISOM, and report the results of numerical as well as field-testing of ISOM. The paper ends with concluding remarks.

BACKGROUND

IED prediction models aim to forecast the number of IEDs in different sectors of the AO in the near future. Such models (see, e.g., Lantz 2006 and Riese 2006) typically discretize the AO into

a grid of square cells and consider only cells within a given distance of known roads. They then calculate a “threat level” for each cell, which may represent the expected number of emplaced IEDs in that cell the next day, based on various model inputs such as historical IED events, historical blue force activity, infrastructure characteristics, and geography. The output is a map graphically depicting the threat level throughout the AO on the next day.

While standard UAS-mounted electro-optical and infrared sensors may catch insurgents in the act of placing IEDs, those already emplaced are much harder to detect. Currently, significant resources are directed towards developing airborne sensors for detection of IEDs already in place. These efforts primarily utilize change detection, multispectral imaging, and hyperspectral imaging (Pike and Aftergood 2006). While initial field tests of these sensors have revealed an unacceptable high false-positive detection rate (Funkhouser 2006, Trimble 2006, Owen et al. 2005), industry has continued to pursue this effort. In addition to technological advances, we believe that focusing the search in sectors of the AO with predicted high IED activity will reduce the number of false-positive detections and make aerial search for emplaced IEDs worthwhile.

To the authors’ knowledge, the only published study on allocation of UASs for IED search is Kress et al. (2008). That study defines a road segment to be “determined” if there is strong evidence of either the presence or absence of an IED in the segment. The study presents an integer linear program that maximizes the expected number of road segments “determined” with a finite number of “looks” by UASs. Kress et al. (2008) does not consider physical restrictions on UASs such as flight-path continuity and travel times.

The search for a single stationary target in a set of cells given a known prior probability distribution of the target location is a well-solved classical search problem (see, e.g., Washburn

2002). However, since the number of IEDs in the AO is unknown, the classical results are not applicable in our context.

While the literature on IED search by UASs is scarce, there are many studies on optimal routing of military aircraft. Zabaranin et al. (2006) and Royset et al. (2008) consider minimum-risk routing of military aircraft and formulate that problem as a constrained shortest-path problem. O'Rourke et al. (2001) and Harder et al. (2004) develop a dynamic routing tool for UASs, which uses a tabu search heuristic to determine routes. The tool specifically addresses routing of the Predator UAS. Shetty et al. (2006) consider routing and target assignment for combat UASs. They formulate the problem as a mixed-integer linear program, which is solved by decomposing the problem into routing and assignment subproblems. Kress and Royset (2008) allocate small UASs and their ground control stations for special operations. They formulate the problem as a two-stage stochastic optimization model, where allocating ground control stations represents the first-stage decision and routing of UASs represents the second stage.

Moser (1990) formulates an aerial reconnaissance problem as a team orienteering problem with time windows. The formulation generalizes the classical orienteering problem (see, e.g., Feillet et al. 2005 for a review) by considering a team of players and time window constraints that restrict when nodes can be visited. The resulting optimization model is computationally expensive to solve optimally. In fact, the orienteering problem is NP-hard (Golden et al. 1987) and the generalized problem is at least as difficult to solve.

Chao et al. (1996), Tang et al. (2005), and Archetti et al. (2007) develop and compare heuristic algorithms for solving the team orienteering problem (without time windows). Boussier et al. (2005) construct an exact algorithm for the same problem using branch-and-price and specialized branching rules. Kantor and Rosenwein (1992) consider the (single-player)

orienteering problem with time windows and develop two heuristic algorithms. Time window constraints also arise in the context of vehicle routing problems, see, e.g., Tan et al. (2001) and Toth and Vigo (2002).

In this paper, we are motivated by the team orienteering problem with time windows in Moser (1990). However, the nature of IED search by UASs necessitates a new optimization model which determines the take-off time for the UASs, accounts for airspace de-confliction, and distinguishes between UAS search and transit between waypoints. Moreover, we utilize the typical close proximity between likely IED locations to heuristically reduce the model size and make it tractable by standard integer-programming solvers.

COMBAT SITUATION AND PROBLEM DEFINITION

We consider an AO that may stretch for several tens of kilometers. An example is the 30-kilometer by 40-kilometer area that encompasses the cities of Fallujah and Ramadi in western Iraq. In this area, a UAS squadron is tasked with searching roads for IEDs in support of three to four regimental combat teams in addition to one brigade combat team. There is a small number, U say, of UASs (e.g., one to five) available for IED search during a portion of the next 24-hour air tasking order (ATO) cycle. That portion of time defines the planning horizon. Each UAS is equipped with a sensor that can detect emplaced IEDs and/or a sensor that can detect insurgents in the act of placing an IED. Each UAS is launched from a base and must return to the same base by the end of the planning horizon and no later than what its flight endurance permits.

We assume that an IED prediction model is available for the AO and that the prediction model has discretized the AO into I squared cells, each containing a portion of the road network and surrounding areas within the AO. We assume that I is much larger than U . Typically, cell size in IED prediction models is between 50 and 800 meters square (Riese 2006, Lantz 2006).

The prediction model assigns each cell i with a numerical value τ_i that represents the “threat level” in that cell. Typically, τ_i is the prediction model’s estimate of the number of emplaced IEDs in that cell. Hence, we refer to τ_i as the expected number of IEDs in cell i .

The UASs will launch from their bases, fly over some of the cells and attempt to interdict IEDs, and then return to their bases. Each UAS has a known search effectiveness that represents the probability of interdicting an IED in a cell, given that there is an interdiction opportunity in the cell. The goal is to determine take-off times and routes for the UASs so that the expected number of IED interdictions are maximized, while constraints on the flight duration and airspace de-confliction are satisfied. We refer to this problem as the UAS routing problem.

MODEL FORMULATION AND PREPROCESSING PROCEDURES

The routing tool ISOM consists of an integer linear program that models the UAS routing problem, a set of preprocessing procedures that restrict the program, and a standard integer programming solver that finds an optimal or near-optimal solution of the restricted program. In this section, we formulate the UAS routing problem as an integer linear program and discuss associated assumptions. Since the resulting program is a generalization of the orienteering problem, it is impractical to solve the integer linear program directly by means of standard integer programming solvers. Hence, we derive a set of preprocessing procedures that restricts the program, reduces its size, and allows solution of practical instances in reasonable computing time.

Model Formulation

We discretize the planning horizon into T time steps and denote the endurance (in time steps) of UAS u ($u \in \{1, 2, \dots, U\}$) by $e_u(\leq T)$. In practice, the UASs may not be available for IED search

during the whole 24-hour ATO cycle. Hence, the planning horizon is often much shorter than 24 hours.

We assume that the sweep-widths of the UASs' sensors are at least as large as the side of the cells so it is only necessary to fly over the cell once to cover the whole cell. If that is not the case, we subdivide the cells into smaller cells and divide the expected number of IEDs in the cell accordingly. We assume that the UASs will fly in straight lines from the center of a cell to the center of another cell and will not loiter at one cell.

We observe that existing IED prediction models only provide a single number, τ_i , representing the expected number of IEDs in cell i during the next day and do neither predict the time of placement of IEDs nor quantify the probability distribution of the number of IEDs in the cell. Hence, we adopt the following simple model of how the expected number of IEDs in cell i evolves during the planning horizon. Let $\tau_{i,0} \leq \tau_i$ be the expected number of IEDs present in the cell prior to the first time step. IED prediction models do not currently provide $\tau_{i,0}$, but that quantity can be estimated as a fraction of τ_i using the length and time of the planning horizon and intelligence assessments about the timing of IED placement. Moreover, let $\tau_{i,t} = \tau_{i,0} + (t/T)(\tau_i - \tau_{i,0})$ be the expected number of IEDs in cell i at time step $t \in \{1, 2, \dots, T\}$. Consequently, $(1/T)(\tau_i - \tau_{i,0})$ is the rate with which more IEDs are placed in the cell. We also assume that no IEDs are removed except as a result of interdiction by UAS. Clearly, this simple model may not completely describe the true number of IEDs in a cell during the planning horizon. However, in view of the data available to planners, the model appears reasonable.

To simplify the integer program and reduce computing times, we restrict the UASs to routes that search no cell more than once. This restriction is not significant in situations where few

IEDs are placed during the planning horizon, i.e., $\tau_{i,0}$ is close to τ_i , and the false-negative detection rate is relatively small. In such situations, it is better to search new cells instead of revisiting previously searched cells. Even in other situations, the large number of cells in the AO and the typical alternative objective of maximum search coverage justify the restriction to search no cell more than once.

IED prediction models do not provide information about the dependence between the numbers of IEDs in two different cells. Hence, it is unclear how information about IED interdictions (or no interdiction) in one cell influences our belief about the expected number of IEDs in the other cells. Consequently, we assume as in Kress et al. (2008) that the numbers of IEDs in different cells are statistically independent and we do not update our belief about the expected number of IEDs in other cells after searching a cell. Moreover, we do not consider false-positive detections as such detections in a cell will not – due to independence – change our belief about the expected number of IEDs in other cells. We simply assume that each UAS will execute its preplanned route regardless of interdictions during the planning horizon. It is beyond the scope of the paper to account for the possibility of future recourse actions while planning the route.

We adopt the following detection model. Let $\alpha_{i,t,u} \in [0,1]$ denote the probability of detecting an emplaced IED in cell i at time step t by UAS u given that the IED is in the cell. Assuming independence between the probabilities of detecting different IEDs in the same cell, we find that the number of IEDs detected in cell i by UAS u at time step t , given that there are n IEDs in the cell, has a binomial distribution with parameters n and $\alpha_{i,t,u}$. Since the expected number of IEDs in cell i at time step t is $\tau_{i,t}$, we find that the expected number of interdiction of emplaced IEDs in cell i at time step t is $\alpha_{i,t,u}\tau_{i,t}$ given that the cell is searched by UAS u . Furthermore, let

$\beta_{i,t,u} \in [0,1]$ denote the probability of detecting an insurgent in the act of placing an IED in cell i at time step t by UAS u given that the act takes place. Using an identical binomial detection model to the one for emplaced IEDs, we find that the expected number of interdiction of insurgents placing IEDs in cell i at time step t is $\beta_{i,t,u}(\tau_{i,t+1} - \tau_{i,t})$ given that the cell is searched by UAS u . Here, $\tau_{i,t+1} - \tau_{i,t}$ is the expected number of IEDs placed in cell i between time steps t and $t+1$, with $\tau_{i,T+1}$ set equal to $\tau_{i,T}$. Hence, the total expected number of IED interdictions in cell i at time step t by UAS u is $p_{i,t,u} = \alpha_{i,t,u}\tau_{i,t} + \beta_{i,t,u}(\tau_{i,t+1} - \tau_{i,t})$. The integer linear program aims to route the UASs such that the total expected number of IED interdictions, over all UASs and all time steps, is maximized.

Even though search of a cell can take place at most once in our model, we allow any number of transits through a cell. Hence, the model formulation below distinguishes between visiting a cell with the purpose of searching or transiting. We let $a \in \{0,1\}$ denote these ‘‘actions,’’ where 0 and 1 represent search and transit, respectively.

In the set of cells, we include special depot cells d_u , $u \in \{1,2,\dots,U\}$, that represent the bases of the UASs. UAS u must depart d_u and return to it no later than e_u time steps after its departure and no later than time step T . The depot cells have no opportunity for IED interdiction, i.e., $\tau_{d_u,t} = 0$ for all time steps t and UAS u . A cell can be the depot node of more than one UAS. We let $c_{i,j,u,a}$ be the travel time between cell i and cell j for UAS u when it carries out action a at cell i (measured in time steps). In practice, the travel times between different locations depend on the wind conditions. Wind conditions are of particular importance for small UASs with maximum air speed often less than 40 knots. We adopt a simple model where the travel time $c_{i,j,u,a}$ is a known deterministic number for all i, j, u , and a , which is computed based on the distance

between cells i and j and the expected ground speed, accounting for expected winds, for UAS u during the flight segment between cells i and j . (We refer to Nachmani (2007) for a more detailed wind model.) Hence, the travel times are typically asymmetric, i.e., $c_{i,j,u,a} \neq c_{j,i,u,a}$. We discuss the calculation of $c_{i,j,u,a}$ in further detail in the last section.

In some cases, particularly with small UASs, multiple UASs are assigned to search in a given sector within a limited altitude block. For these situations it is indispensable that we ensure separation of UASs throughout the planning horizon to avoid accidents. In order to address this situation, we define G groups. Each group consists of a set of UASs that need to be mutually de-conflicted. We let U_g denote the set of UASs in group $g \in \{1, 2, \dots, G\}$. Moreover, we let $A(i,j)$ denote the pairs of cells that are “in conflict” with the pair of cells (i,j) . Specifically, if (i,j) is flown by a UAS in a group, it will, unless time separated, become too close to another UAS in the same group flying between any pair of cells in $A(i,j)$. Adequate time separation is ensured using the set $D(t)$, which contains all time steps that are too close to t for sufficient time separation. Finally, we let $F(i)$ denote the “forward star” of cell i , i.e., the set of cells to which a UAS may fly directly from cell i . In principle, $F(i)$ could simply be all cells except i , but as we will see below that will result in a large integer linear program. In a later section, we discuss how to construct a suitable $F(i)$ which balances the size of the integer linear program with flight-path flexibility for the UASs. Similarly, we let $R(i)$ denote the “reverse star” of cell i , i.e., the set of cells from which a UAS may fly directly to cell i . Using this notation, the formulation of the integer linear program P is as follows.

Indices

i,j,k,l Cell, $i,j,k,l \in \{1, 2, \dots, I\}$.

u UAS, $u \in \{1, 2, \dots, U\}$.

t, s Time step, $t, s \in \{1, 2, \dots, T\}$.

a Action in cell ($a = 0, 1$ represent “search” and “transit,” respectively).

g Group of UASs to be mutually de-conflicted, $g \in \{1, 2, \dots, G\}$.

Sets

$F(i)$ Forward star of cell i .

$R(i)$ Reverse star of cell i .

U_g Set of UASs u in group g .

$A(i, j)$ Set of pairs of cells which are “in conflict” with pair (i, j) .

$D(t)$ Set of time steps too close to t for adequate separation of UASs.

Data

$P_{i,t,u}$ Expected number of IED interdictions in cell i at time step t by UAS u given search by UAS u .

d_u Cell which serves as depot of UAS u .

e_u Flight endurance in time steps for UAS u .

$c_{i,j,u,a}$ A positive travel time in time steps from i to j for UAS u performing action a at i .

Binary Variables

$X_{i,j,t,u,a}$ 1 if UAS u starts moving from cell i to cell j at time step t , and performs action a at cell i , 0 otherwise.

Mathematical Formulation

$$P: \max \sum_{i,j,t,u} p_{i,t,u} X_{i,j,t,u,0} \quad \text{(Objective function)} \quad (1)$$

subject to

$$\sum_{j \in F(d_u), t, a} X_{d_u, j, t, u, a} = 1 \quad \forall u \quad \text{(Start at depot)} \quad (2)$$

$$\sum_{\substack{j \in R(i), a \\ s: s-t-c_{j,j,u,a} \geq 1}} X_{j,i,s,u,a} = \sum_{j \in F(i), a} X_{i,j,t,u,a} \quad \forall t, u, i \neq d_u \quad \text{(Continuity of route)} \quad (3)$$

$$\sum_{j \in F(i), t, u} X_{i,j,t,u,0} \leq 1 \quad \forall i \quad \text{(Prevent duplicate search)} \quad (4)$$

$$\sum_{\substack{i \in R(d_u), a \\ t: t+c_{i,d_u,u,a} \leq T}} X_{i,d_u,t,u,a} = 1 \quad \forall u \quad \text{(Return to depot)} \quad (5)$$

$$\sum_t \left(\sum_{\substack{j \in F(d_u), a \\ s: s \leq t}} X_{d_u, j, s, u, a} - \sum_{\substack{i \in R(d_u), a \\ s: s+c_{i,d_u,u,a} \leq t-1}} X_{i, d_u, s, u, a} \right) \leq e_u \quad \forall u \quad \text{(Endurance)} \quad (6)$$

$$\sum_{\substack{(k,l) \in A(i,j) \\ s \in D(t), u \in U_g, a}} X_{k,l,s,u,a} \leq 1 \quad \forall i, j, t, g \quad \text{(Airspace de-confliction)} \quad (7)$$

$$X_{i,j,t,u,a} \in \{0,1\} \quad \forall i, j, t, u, a$$

Equation (1) defines the objective function which represents the expected number of IED interdictions during the planning horizon. Constraints (2) ensure each UAS departs its depot exactly once. Constraints (3) ensure that if a UAS arrives at a cell during any given time step, it must depart that cell on the same time step. In constraints (4), we limit the number of times a cell is searched; however, we permit UASs to transit previously searched cells. Constraints (5)

ensure that each UAS returns to its depot exactly once. Note that we permit a UAS to return to its depot early with no penalty. Constraints (6) limit the flight duration for each UAS. The first term inside the parenthesis is one if UAS u has taken off at time step t , and zero otherwise. The second term is one if UAS u has returned to its base at time step t , and zero otherwise. Hence, the left-hand side counts the number of time steps a UAS is in the air. Finally, constraints (7) enforce time and space separation between UASs in the same group. In addition to the constraints (1)-(7), blue force activity may impose further restrictions as we discuss next.

Blue force presence in a cell is clearly a prerequisite for a blue force casualty in that cell. Hence, one may argue that it is best to search a cell immediately prior to blue force activity in that cell. However, large distances between cells of interest, the uncertainty in blue force activity, ground commanders desire to keep all options on the table, and the goal to protect police and civilians complicate the situation and make the strategy impractical. Still, it is undesirable to search a cell right after blue force activity. If possible, the search should have taken place before the activity. For example, consider the case where it is known in advance that a route clearance team will sweep a particular route some time before we expect a convoy to travel the route. Due to the low speed of the route clearance team, it may be more than one hour ahead of the convoy, which typically travels at much higher speed. If these operations take place according to plan, then for each cell that contains a portion of the route in question there is a window of opportunity for insurgents to place an IED and target the coming convoy. This window begins at the time the route clearance team is outside of visual range of the cell and lasts until the convoy arrives. This window of opportunity for the insurgents translates to a time window in which the cell is a candidate for search. This situation can trivially be incorporated into P by the following addition:

Additional Set

$W(i)$ Time steps t that comprise the time window for search of cell i .

Additional Constraints

$$X_{i,j,t,u,0} = 0 \quad \forall i, j \in F(i), t \notin W(i), u \quad (\text{Prevent searches outside time window}) \quad (8)$$

While we acknowledge that blue force operations do not always go as planned, we believe this type of coordination between UAS efforts and blue force activity does stand to bear fruit not only in preventing IED casualties but also by catching insurgents in the act, revealing critical intelligence for disrupting the network behind the device. Furthermore, the addition of (8) leads to a significant restriction of P , which may reduce computing time.

Preprocessing Procedures

The integer program P contains $2UTI^2$ binary variables; a potentially large number. To facilitate the solution of P , using standard integer programming solvers, we consider several restrictions to P in an attempt to reduce the size of the program without worsening the optimal value significantly. Since a UAS flies from the center of one cell to the center of another cell in a straight line, we adopt the language of graphs and refer to the center of a cell as a node and an ordered pair of nodes as a (directed) arc. The complete graph consisting of the nodes generated by the cells in the AO and all arcs connecting the nodes represents the possible flight segments for the UASs. We duplicate this graph U times so there is one graph for each UAS. Then, for each graph (UAS), the procedures described in this section remove nodes and arcs that represent flight segments deemed unlikely to be “good” for the corresponding UAS. Since each decision variable in P corresponds to a flight segment between two nodes, the removal of arcs in the graphs specifies variables in P that can be restricted to zero (hopefully) without affecting the optimal value of P significantly. Hence, we can use the following procedures to restrict P .

Variable Elimination

We first eliminate decision variables $X_{i,j,t,u,0}$ with expected number of interdictions $p_{i,t,u}$ below a specific threshold. The corresponding flight segments are most likely not on an optimal flight route so the effect on the optimal value is expected to be marginal for low thresholds.

Node Clustering

We aggregate nodes that are located in close proximity to each other; a process we refer to as node clustering. When nodes are close to each other, e.g., they represent the centers of adjacent cells, it is sensible to aggregate them and consequently remove the associated arcs from the graph. We examine two cases depending on the sweep-width of the UAS being considered. The first case deals with a sweep-width wide enough to cover nodes adjacent to one another. We refer to this as the large sweep-width case. The second case, or small sweep-width case, addresses sweep-widths not large enough to cover adjacent nodes.

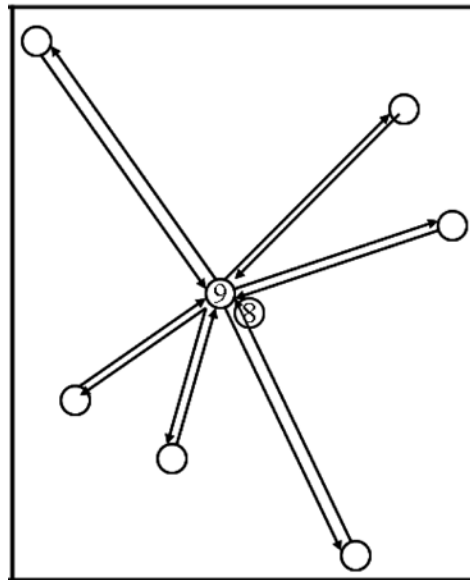


Figure 1. Node clustering for UAS with *large* sweep-width. In this example, node 8 is eliminated and its expected number of interdictions is moved to node 9.

For the large sweep-width case, we reassign the expected number of interdictions at nodes that are in close proximity according to two sets of circumstances. For the situation where two nodes are adjacent to each other and set apart from other nodes in the graph, we eliminate one of the nodes, remove all arcs going into and out of the eliminated node, and add the expected number of interdictions of the eliminated node to the adjacent node (see Figure 1). This clustering rule has the effect of creating one node of relatively large number of interdictions and fewer arcs necessary to achieve these interdictions, which simplifies the decision of whether or not to include this node in an optimal solution.

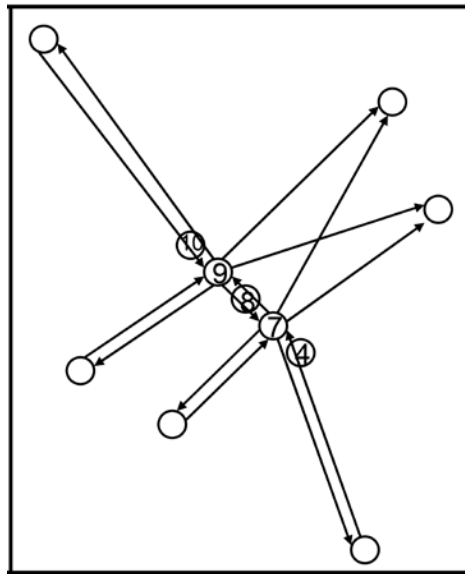


Figure 2. Node clustering for UAS with *large* sweep-width. In this example, arcs into and out of nodes 4, 8, 10 are eliminated. The expected number of interdictions at 4 is added to 7, the number at 10 is added to 9, and the number at 8 is divided between 7 and 9.

A more likely occurrence, given that the nodes are derived from cells along a road network, is the situation where several nodes are closely aligned in a row. For this circumstance, we eliminate nodes and reassign the expected number of interdictions similar to the previous case; however, we do this on the basis of position rather than expected number of interdictions. We

eliminate the nodes on each end of the row of nodes and reassign the expected number of interdictions to their respective adjacent node. Any node that falls between the new end nodes is also eliminated with its expected number of interdictions divided between the new end nodes (see Figure 2).

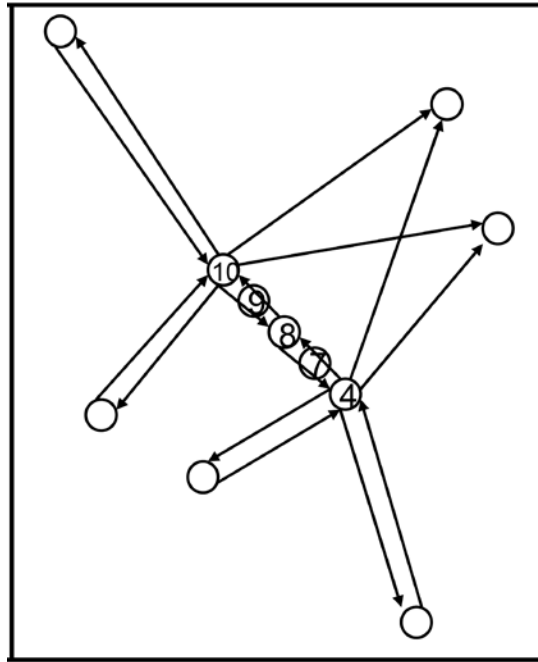


Figure 3. Node clustering for UAS with *small* sweep-width. In this example, arcs into and out of nodes 7 and 9 are eliminated. The expected number of interdictions at nodes 7 and 9 are added to node 8. Node 8 is connected to node 4 and 10 only.

For the case where the sweep-width is not large enough to sufficiently cover adjacent nodes, we consider only the situation of several nodes in a row. We reason that in most cases where a row of nodes exists, it is beneficial to enter at one end and fly along the row searching all the associated cells before altering course. Hence, we eliminate all nodes except one between the end nodes in the row, reassign the expected numbers of interdictions of the eliminated nodes to the remaining node, remove all arcs going into and out of the eliminated nodes, and eliminate all arcs going into and out of the remaining node except those connecting to the end nodes (see

Figure 3). This clustering rule has the effect of creating one high value node at the center of the row which simplifies the decision whether or not to enter the row.

Redundant Arc Removal

After node clustering, we consider rules for further elimination of arcs in the graphs. We find that there are many cases in the graphs where longer arcs strictly over-fly two or more arcs (see Figure 4). The contribution toward the objective function (1) associated with flying the longer arc is simply the expected number of interdictions at the tail node of the arc. The contribution corresponding to the equivalent shorter arcs is the sum of the expected numbers of interdictions at their tail nodes. If the longer arc take the same time or longer to fly than the equivalent shorter arcs, the longer arc can be eliminated because the shorter arcs collectively provide no smaller expected number of interdictions. We observe that these circumstances allow us to significantly reduce the network with no effect on the optimal solution.

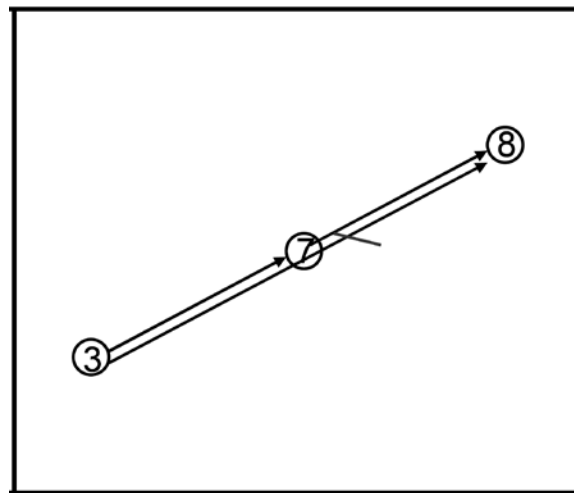


Figure 4. Redundant Arc Removal. In this example, the arc (3, 8) is eliminated since the flight segments represented by arcs (3, 7) and (7, 8) result in at least the same expected number of interdictions and consume the same travel time.

Arc Filtering

We extend the redundant arc removal rule of the previous section to also eliminate a long arc between two nodes that are only slightly offset from two shorter arcs connecting the same nodes, see Figure 5. Flight along the long arc in Figure 5 provides only interdiction opportunities at nodes 3 and 9, while flight along the short arcs may also result in interdictions at node 5. Hence, the slightly shorter distance along the long arc will tend to be insufficient to make the long arc part of an optimal solution. Specifically, we eliminate arc (i,j) when there exists a node k and arcs (i,k) and (k,j) such that $c_{i,k,u,0} + c_{k,j,u,0} \leq (1 + \delta)c_{i,j,u,0}$, where δ is a nonnegative algorithm parameter.

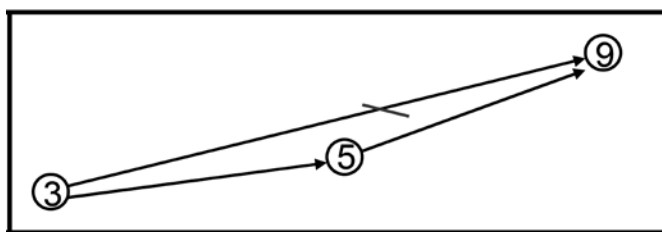


Figure 5. Arc filtering with slightly offset route. In this example, arc $(3, 9)$ is eliminated since the flight segments represented by arcs $(3, 5)$ and $(5, 9)$ result in at least the same expected number of interdictions and consume only marginally more travel time.

We rationalize that it is unlikely an optimal route will contain excessively long arcs. For this reason, we eliminate arcs with travel time greater than a specific limit η . The last section of the paper examines various arc length limits, their resulting computing times, and effect on the solutions obtained.

The procedures described in the previous sections reduce the size of the graph corresponding to each UAS. If arc (i,j) in the graph of UAS u has been removed using the procedures, we set $X_{i,j,t,u,0} = 0$ for all t in P . Hence, we rule out the option to search node i with UAS u during the flight segment corresponding to arc (i,j) .

Transit Arc Selection

We further restrict P by limiting the number of flight segments for transit. Specifically, after the procedures of the previous sections, we divide each of the resulting graphs into sectors containing multiple nodes. Within each sector, we select one node. In the sector containing the UAS's depot, we select the depot node. Then, we set $X_{i,j,t,u,l} = 0$ for all t in P unless i and j are selected nodes in different sectors with $c_{i,j,u,l} \leq \gamma$ and node j is closer to the depot of UAS u than node i . Here, γ is a positive algorithm parameter. Consequently, transit takes place on a sparser graph than search. Clearly the limitation on the number of transit possibilities in P restricts the problem further.

NUMERICAL AND FIELD EXPERIMENTS

During 2007, ISOM was implemented in two, multi-day field experiments at Camp Roberts, California carried out as part of the Naval Postgraduate School-USSOCOM field experimentation program. These field experiments involved up to three UASs searching for simulated IEDs. The goal of these experiments was to test ISOM, in cooperation with UAS operators and commanders, in a real-world, operational setting.

Scenario Development

The AO for the field experiment at Camp Roberts, approximately 5-by-8 kilometer, has not been implemented in any IED prediction model. Hence, we opted to simulate the output of an IED prediction model. We discretize the AO into a grid of 200-meter by 200-meter cells and retain only cells containing roads. The cell size of 200 meter is consistent with Lantz (2006). The threat levels τ_i are randomly drawn from the uniform distribution, independently for each cell. We

selected the uniform distribution as this provides an even distribution of the threat level τ_i , a situation that tends to cause the most difficulty for the integer programming solver. In comparison, if the threat levels had been large in some cells, the solver would find it easier to bound away solutions that do not include those cells and the calculation times would be reduced. We set $\tau_{i,0} = \tau_i$ and avoid classified sensor information by setting $\alpha_{i,t,u} = 1$ for all cells i , time steps t , and UASs u . We note that $\alpha_{i,t,u} = 1$ implies an unrealistically effective sensor, but since the value is identical for all cells, time steps, and UASs the value is immaterial: All positive values give the same optimal solution.

Three types of UAS are available for the experiments: Scan Eagle, Buster, and Raven, which operate at 40, 35, and 25 knots, respectively, and are equipped with electro-optical sensors. In order to test various aspects of ISOM, it is decided that two types of experiments would be conducted concurrently. Scan Eagle, which is operated at a higher altitude and has a relatively large sweep-width, would be used to test the large sweep-width case of ISOM. Buster and Raven, which have smaller sweep-widths and operate within a narrow block of altitude require airspace de-confliction. Scan Eagle's route is optimized independently of Buster and Raven's. In order to conduct several field experiments in the time allotted, the planning horizon is set to 25 minutes and broken down into 50, 30-second time steps. In practice, the AO will be larger, the planning horizon longer, and the time resolution coarser. Even though Buster, Raven, and Scan Eagle in reality have longer endurance, we set $e_u = 50$ (the length of the planning horizon) for all UASs and fix take-off to the first time step.

While the most important aspect of the field experiments is to study routing of UASs under varying wind conditions, we also simulate IED interdictions by placing a number of large,

conspicuously colored tarps on the ground that would serve as notional IEDs. (We note that our study does not concern development or testing of sensor technology. Hence, realistic IEDs are unnecessary.) Eight scenario realizations, each having four or five IEDs, are generated by random drawings of IED locations from probability distributions corresponding to the simulated IED prediction model described above.

Computational Results

We implement the preprocessing procedures in Microsoft Excel using Visual Basic for Applications and the integer linear program P in General Algebraic Modeling System (GAMS) (GAMS, 2007). We found near-optimal solutions using the CPLEX solver (ILOG, 2007) and relative optimality tolerance of 5%. We experimented with solver options on the given test data with various travel times and found that CPLEX's default options are relatively competitive. However, branching up first appears to reduce run times by about 10% and we generally use that option. All calculations were carried out on a Dell Precision PWS690 Intel® Xeon™ CPU 3.37GHz processor, with 3.00 GB of RAM.

Clearly, the restriction of P caused by the preprocessing procedures described in the previous sections may worsen the routes compared to the optimal routes. In the scenario described in the previous section, P is unsolvable by standard integer programming solvers even for one UAS. Hence, we are unable to quantify the effect of all the preprocessing procedures. However, to illustrate the effect of arc filtering on computing time and the optimal value of P , we consider the following three base cases for one UAS.

The base cases are defined by applying all the procedures except arc filtering to the scenario for different travel times generated by a wind direction of 360° at 0 knots, a direction of 275° at 15 knots, and a direction of 315° at 12 knots. For each base, we apply the variable elimination

procedure and eliminate all decision variables corresponding to $p_{i,t,u} \leq 0.7$. Then, we apply the node clustering procedure. Collectively, these procedures result in a set of 77 nodes including one depot node. After application of the redundant arc removal procedure and the transit arc selection procedure with 1-by-1 kilometer sectors and $\gamma = 3$ time steps, there are for a wind direction of 360° at 0 knots 1155 binary variables related to search ($X_{i,j,t,u,0}$) and 51 related to transit ($X_{i,j,t,u,1}$) remaining in the model per time step. For wind directions of 275° at 15 knots and 315° at 12 knots there are a total of 1151 and 1180 binary variables remaining per time step, respectively.

Wind Direction/speed (degrees/knots)	Arc filtering time limit η (time steps)	Arc filtering deviation limit δ	Number of variables	Computing time (sec.)	Near-optimal value
360/0	∞	0	60,300	13,717	2.763
	8	0	58,400	14,416	2.742
	4	0	39,300	4,729	2.754
	∞	0.05	46,700	9,967	2.763
	8	0.05	44,600	3,393	2.763
	4	0.05	30,300	2,225	2.726
275/15	∞	0	57,550	2,928	2.497
	8	0	53,500	2,440	2.497
	4	0	36,400	2,265	2.493
	∞	0.05	45,950	1,579	2.497
	8	0.05	42,100	1,799	2.493
	4	0.05	30,150	1,243	2.478
315/12	∞	0	59,000	1,103	2.430
	8	0	56,200	701	2.430
	4	0	38,100	1,303	2.390
	∞	0.05	47,100	546	2.408
	8	0.05	44,200	677	2.408
	4	0.05	30,800	310	2.390

Table 1. Effect of arc filtering procedure for different wind conditions, arc filtering time limits η , and deviation limits δ .

We then solve the base cases with various levels of arc filtering and obtain the results in Table 1. Table 1 illustrates the effect of arc filtering for different wind conditions, arc filtering time limits η , and deviation limits δ . Column 4 shows that arc filtering reduces the number of binary variables in the base cases (60300, 57550, and 59000 in the three base cases) by up to 50%. As a result, we see significant reductions in computing times for the CPLEX solver (Column 5). With the most aggressive arc filtering, all problem instances are solved to near-optimality within 37 minutes of solver run time. We note that situations with wind and, hence, asymmetric travel times, solve significantly faster. This reduction in computing time is due to the fact that asymmetry makes it easier for the CPLEX solver to bound away solutions as they tend to be more different than in the case of symmetric travel time where many solutions tend to be quite similar. However, if the asymmetry is marginal, as in the case of light wind, the reduction in computing time would be less significant. As Column 6 demonstrates, the restriction of P caused by arc filtering gives only a small deterioration in solution quality. In each of the three base cases, all near-optimal values obtained with arc filtering are within 2% of the ones obtained without arc filtering.

We are unable to compare ISOM directly with a human planner. However, we conjecture that a human planner will find it difficult and time consuming to determine a good route, especially in the case of multiple UASs and airspace de-confliction constraints (see Applegate et al. (2006), pp. 31-40, for the related problem of humans selecting traveling salesman tours). Table 2 summarizes the performance of the routes produced by ISOM (using $\eta = 4$ and $\delta = 0.05$) for a single UAS in the scenario described above in four cases corresponding to four different randomly generated wind conditions. As seen from column 3 of Table 2, ISOM produces routes that cover between 52 and 57 percent of the 77 cells in the AO. The right half of Table 2 gives,

for each case and for each scenario realization of IED locations, the percentage of cells with IEDs searched by the recommended route. From Table 2 we find that ISOM produces routes that, on average, search 78% of the IEDs in the AO while only examining 55% of the total number of cells. Hence, ISOM directs the UASs to likely IED locations and tends to outperform a random search policy.

Case	Wind	% of cells searched	% of IED cells searched								
			Scenario realization:	1	2	3	4	5	6	7	8
1	035/14	53		80	75	50	50	100	75	100	50
2	135/12	52		80	75	75	50	100	75	75	50
3	235/10	57		80	75	75	100	100	100	100	50
4	310/12	57		100	100	75	50	100	100	100	25

Table 2. Performance of routes generated by ISOM measured in percent of cells with IEDs searched over eight scenario realizations.

Field Experiment Results

While the field experiments involved a Raven, a Buster, and a Scan Eagle UAS (see Figure 6 for a screenshot taken from the tactical operations center) we here report the results from nine, tentatively 25-minute experiments aimed at adjusting ISOM to the operational airspeed of the Buster and Raven under real-world weather conditions.

ISOM takes as input the nominal speed of the UASs during search and transit and the predicted wind speed and direction in each cell of the AO and computes the travel times $c_{i,j,u,a}$ using vector calculations. The travel times are rounded up to an integer number of time steps. ISOM is a high-level routing tool that does not directly account for the flight dynamics of the UAS. For example, P may result in highly jagged routes that cannot exactly be flown by the UASs. The node clustering procedure significantly reduces the need for frequent turns, but the routes remain somewhat jagged as seen in Figure 6. Hence, in reality the UAS selects a smooth

trajectory that closely approximates the optimized routes. This fact and the uncertainty in wind speed may make it necessary to adjust the travel times $c_{i,j,u,a}$ so that the UASs return to their respective bases no later than the allowable flight duration.

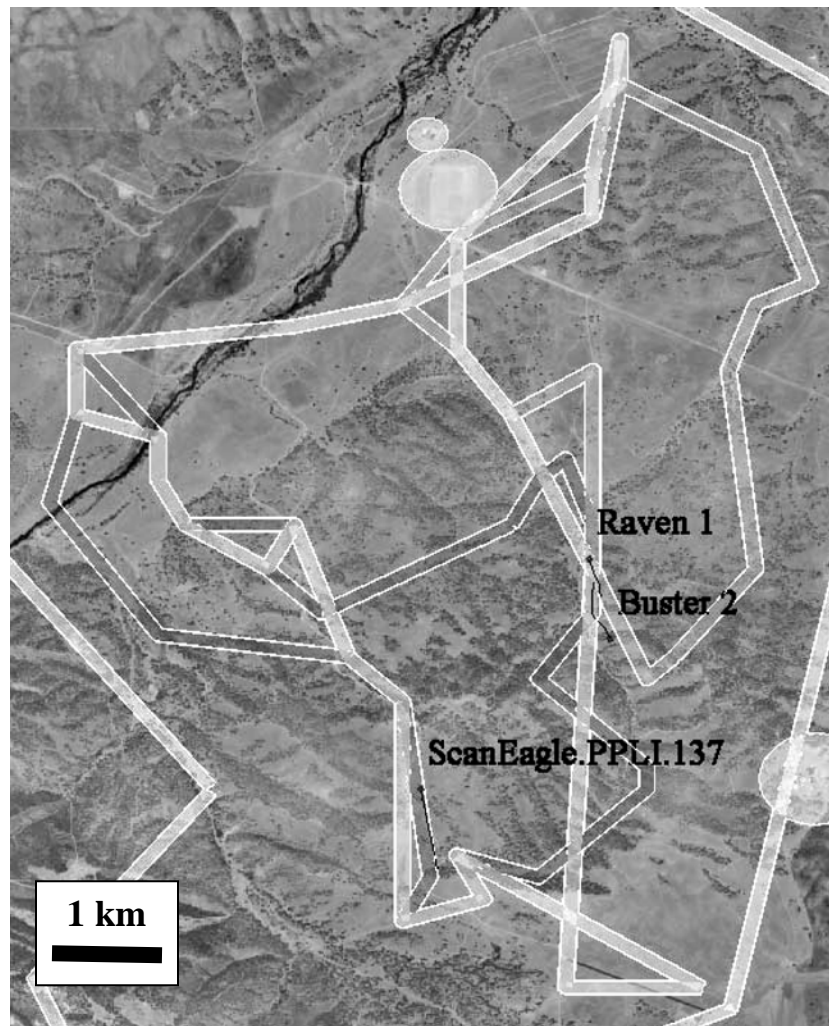


Figure 6. Optimized routes for Raven, Buster, and Scan Eagle UASs during a field experiment at Camp Roberts.

We adjust the travel times by using various speed reduction factors ranging from 70% to 100% of the nominal UAS speed. The reduced speed results in longer travel times in ISOM and consequently in shorter routes and a better chance to return to base in time. In nine field experiments with Buster and Raven, we examined the effect of different speed reduction factors

as well as the effect of ignoring wind predictions. Actual wind data were collected on site by a weather balloon. Table 3 presents the actual time required for Buster and Raven to complete routes generated by ISOM using a travel time limit of 25 minutes, i.e., $e_u = 50$ time steps of 30 seconds, and various speed reduction factors. In these cases, ISOM requires between 2 and 4 hours of run time to achieve a 5%-optimal solution.

The first six experiments used no wind correction and we see that the actual flight times tend to be unnecessarily short for a speed reduction factor of 70% and possibly too long for reduction factors of 90% and 100%. The wind corrected travel times in experiments 7-9 appear to estimate the actual travel times quite well without the need for a speed reduction. In addition, wind correction of travel times $c_{i,j,u,a}$ results in a smaller variability in actual travel times. While there is not enough data to make a conclusive argument, we believe the results suggest that no speed reduction factor is necessary when using wind corrected travel times and that the rounding of $c_{i,j,u,a}$ up to an integer time step tends to be negated by the extra time required by turns.

Experiment #	Speed Reduction Factor	Wind Corrected	UAS	Actual Time
1	70%	NO	Buster	13 min
			Raven	14 min
2	90%	NO	Buster	23 min
			Raven	24 min
3	90%	NO	Buster	24 min
			Raven	28 min
4	90%	NO	Buster	20 min
			Raven	<i>(no data)</i>
5	100%	NO	Buster	24 min
			Raven	<i>(no data)</i>
6	100%	NO	Buster	29 min
			Raven	<i>(no data)</i>
7	85%	YES	Buster	22 min
			Raven	19 min
8	90%	YES	Buster	20 min
			Raven	21 min
9	100%	YES	Buster	24 min
			Raven	23 min

Table 3. Actual time required to complete routes generated by ISOM using a 25 minute travel time limit and various speed reductions and wind corrections.

CONCLUSIONS

We have developed an optimization-based routing tool called ISOM for Unmanned Aerial Systems (UASs) tasked with interdiction of Improvised Explosive Devices (IED). ISOM uses integer linear programming and preprocessing procedures to leverage recent developments in IED prediction as well as emerging UAS and sensor technology. ISOM receives output from an IED prediction model and uses it as a means to establish relative values for searching various sectors within an area of operations. The output of the tool is a set of routes which focus UAS search efforts in the most likely areas of IED occurrence and result in the maximum expected number of IED interdictions within the flight duration allowed by the UAS endurance.

With recent increases in the number of UASs in theater, it is indispensable that tactical level operators use an optimization-based approach for selecting routes that best employ their UASs.

Provided that prediction model output data are available, ISOM could be an integral part of the flight planning process for any area of operations.

Currently, ISOM provides a route for a single UAS in moderate computing time (6-37 minutes for the instances tested), but multiple UASs may result in substantially longer calculation times. (These times are often reduced by a factor of 10 when blue forces restrict the IED search to specific time windows.) While current computational requirement of ISOM may be acceptable for nightly planning of the next day's routes, the development of a fast, specialized heuristic algorithm may allow dynamic re-planning of routes during missions as conditions on the ground change. Such a heuristic algorithm may also offer operators the opportunity to examine multiple situations during mission planning. The development of specialized heuristic algorithms as well as further development of more efficient and effective preprocessing procedures for integer linear programs of the kind developed in the paper are topics for further research. We also believe that future efforts should include an examination of ISOM's robustness and effectiveness in a wider variety of simulated scenarios.

ACKNOWLEDGEMENTS

The authors are grateful to Dr. David Netzer, Naval Postgraduate School, Mr. Jack Keane, Johns Hopkins University Applied Physics Laboratory, USSOCOM, and the Office of Naval Research for supporting this research project. The authors also greatly appreciate the invaluable input received from Major M. Scioletti and Professors R. Dell, G. Bradley, and M. Kress of the Naval Postgraduate School.

REFERENCES

- Archetti, C., Hertz, A., and Speranza, M.G. 2007. Meta-Heuristics for the Team Orienteering Problem, *J. Heuristics*, Vol. 13, No. 1, 49-76.
- Applegate, D.L., Bixby, R.E., Chvatal, V., and Cook, W.J. 2006. *The Traveling Salesman Problem*, Princeton University Press.

- Boussier, S., Feillet, D., and Gendreau, M. 2007. An Exact Algorithm for Team Orienteering Problems, *4OR Quarterly Journal of the Belgian, French, and Italian Operations Research Societies*, Vol. 5, No. 3, 211-230.
- Chisholm, P. 2005. Clearing the Roads, *Special Operations Technology Online*, Vol. 3, www.special-operations-technology.com/article.cfm?DocID=1129, last accessed 21 November, 2006.
- Chao, I., Golden, B., and Wasil, E. 1996. The Team Orienteering Problem, *European Journal of Operational Research*, Vol. 88, No. 3, 464-474.
- Fabey, M. 2007. UAVs, Other Aircraft Being Misused in IED Fight, ACC Chief Says, *Aerospace Daily and Defense Report*, Vol. 222, No. 58, 1.
- Feillet, D., Dejax, P., and Gendreau, M. 2005. Travelling Salesman with Profits, *Transportation Science*, Vol. 39, No. 2, 188-205.
- Funkhouser, D.A. 2006. Unmanned Aerial Systems Division Head, Marine Corps Warfighting Lab. Private communication December 11.
- GAMS Homepage. 2007. www.gams.com, last accessed 26 August 2007.
- Golden, B., Levy, L., and Vohra, R. 1987. The Orienteering Problem, *Naval Research Logistics*, Vol. 34, No. 3, 307-318.
- Harder, R.W., Hill, R.R., and Moore, J.T. 2004. A Java Universal Vehicle Router for Routing Unmanned Aerial Vehicles, *International Transactions in Operational Research*, Vol. 11, 259-275.
- ICCC. 2008. Iraq Coalition Casualty Count Web site. <http://icasualties.org/oif/IED.aspx>, last accessed 8 February 2008.
- ILOG Optimization. 2007. www.ilog.com/products/optimization/archive.cfm, last accessed 2 September 2007.
- Kantor, M.G., and Rosenwein, M.B. 1992. The Orienteering Problem with Time Windows, *Operations Research*, Vol. 43, No. 6, 629-635.
- Kress, M., Szechtman, R., and Jones, J.S. 2008. Efficient Employment of Non-Reactive Sensors, *Military Operations Research*, Vol. 13, No. 4, 45-57.
- Kress, M., and Royset, J.O. 2008. Decision Aid for Deployment and Employment of UAVs in Special Operations Missions, *Military Operations Research*, Vol 13, No. 1, 23-33.
- Lantz, R.W. 2006. A Data Mining Approach to Forecasting IED Placement in Space and Time. Master's thesis, Naval Postgraduate School, Monterey, CA.

- Moser, H.D. 1990. Scheduling and Routing Tactical Aerial Reconnaissance Vehicles. Master's thesis, Naval Postgraduate School, Monterey, CA.
- Nachmani, G. 2007. Minimum-Energy Flight Paths for UAVs using Mesoscale Wind Forecasts and Approximate Dynamic Programming. Master's thesis, Naval Postgraduate School, Monterey, CA.
- Owen, P., Martin, R., and Carriger, T. 2005. Shadowing IEDs. *Unmanned Systems*, Vol. 23, No. 5, 13-16.
- O' Rourke, K.P., Bailey, T.G., Hill, R.R., and Carlton, W.B. 2001. Dynamic Routing of Unmanned Aerial Vehicles Using Reactive Tabu Search, *Military Operations Research*, Vol. 6, No. 1, 5-30.
- Pike, J., and Aftergood, S. 2006. Hyperspectral Imaging. Federation of American Scientists. www.fas.org/irp/imint/hyper.htm, last accessed 8 September 2007.
- 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.O., Carlyle, W.M., and Wood, R.K., 2008, Routing Military Aircraft with a Constrained Shortest-Path Algorithm, *Military Operations Research*, to appear.
- Shetty, V.K., Sudit, M., and Nagi, R. 2006. Priority-Based Assignment and Routing of a Fleet of Unmanned Combat Aerial Vehicles, *Computers and Operations Research*, Vol. 35, 1813-1828.
- Tan, K.C., Lee, L.H., Zhu, Q.L., and Ou, K. 2001. Heuristic Methods for Vehicle Routing Problems with Time Windows, *Artificial Intelligence Engineering*, Vol. 15, 281-293.
- Tang, H., Miller-Hooks, E. 2005. A Tabu Search Heuristic for the Team Orienteering Problem, *Computers and Operations Research*, Vol. 32, 1379-1407.
- Toth, P., and Vigo, D. (eds.) 2002. *The Vehicle Routing Problem*. SIAM Monographs on Discrete Mathematics and Applications, SIAM.
- Trimble, S. 2006. U.S. Eyes Hyperspectral Technology for UAVs. *Jane's Defense Weekly*, 6 September, p. 31.
- Washburn, A.R. 2002. *Search and Detection*. 4th edition, INFORMS.
- Zabarankin, M., Uryasev, S., and Murphey, R. 2006. Aircraft Routing under the Risk of Detection, *Naval Research Logistics*, Vol. 53, No. 8, 728-747.