ROUTING MILITARY AIRCRAFT WITH A CONSTRAINED SHORTEST-PATH ALGORITHM

W. MATTHEW CARLYLE
Operations Research Department, Naval Postgraduate School
Monterey, California, USA, mcarlyle@nps.edu

JOHANNES O. ROYSET
Operations Research Department, Naval Postgraduate School
Monterey, California, USA, joroyset@nps.edu

R. KEVIN WOOD
Operations Research Department, Naval Postgraduate School
Monterey, California, USA, kwood@nps.edu

17 April 2007

We formulate and solve aircraft-routing problems that arise when planning missions for military aircraft that are subject to ground-based threats such as surface-to-air missiles. We use a constrained-shortest path (CSP) model that discretizes the relevant airspace into a grid of vertices representing potential waypoints, and connects vertices with directed edges to represent potential flight segments. The model is flexible: It can route any type of manned or unmanned aircraft; it can incorporate any number of threats; and it can incorporate, in the objective function or as side constraints, numerous mission-specific metrics such as risk, fuel consumption, and flight time. We apply a new algorithm for solving the CSP problem and present computational results for the routing of a high-altitude F/A-18 strike group, and the routing of a medium-altitude unmanned aerial vehicle. The objectives minimize risk from ground-based threats while constraints limit fuel consumption and/or flight time. Run times to achieve a near-optimal solution range from fractions of a second to 80 seconds on a personal computer. We also demonstrate that our methods easily extend to handle turn-radius constraints and round-trip routing.

Applications Area: Computing Advances in Military OR
OR Method: Network Methods

1 Introduction

This paper describes the application of a new constrained shortest-path (CSP) algorithm for identifying an optimal or near-optimal route for military aircraft such as strike aircraft, unmanned aerial vehicles (UAVs), and cruise missiles. Mission planning for such aircraft typically seeks to identify a route from origin to destination that balances the risk imposed by some combination of enemy threats, flight time, fuel consumption, strike effectiveness, and possibly other factors. We intend for our algorithm to form the core of an automated route optimizer, or "autorouter," in a mission-planning system.

The difficulty of determining an appropriate route and managing the many details of a mission has prompted the development of a number of air-mission-planning systems. These comprise various hardware and software components for organizing, calculating, and displaying mission-related information. For example, SAIC Mission Planning System (2007) and FalconView (2007) extract relevant information from databases, display manually prepared routes on a computer screen together with geographical information, and analyze the given routes for factors such as threats and fuel consumption. An inter-service mission-planning system is also being developed by the U.S. Department of Defense and several defense contractors, with operational testing under way (JMPS 2007).

Manually planned routes have obvious disadvantages, and fast autorouters will eventually become standard components of mission-planning systems. In fact, some autorouters for military aircraft do exist, including CLOAR (2007), OPUS (2007), and JRAPS (see Tharp 2003). However, as discussed below, these have a number of modeling and computing shortcomings.

Two model types have been proposed for autorouting: (*i*) Continuous models, classically based on the calculus of variations, and (*ii*) discrete models that represent airspace as a network. See Vian and More (1989), Novy (2001), and Zabarankin et al. (2006) for examples of the former case, and see Lewis (1988), Leary (1995), Lee (1995), Grignon et al. (2002), Kim and Hespanha (2003), and Zabarankin et al. (2006) for examples of the latter.

A typical continuous model seeks to identify an optimal route, defined via one or more continuously varying curves, by solving of a system of nonlinear equations; see Hebert (2001) for a detailed review. A series of papers (Zabarankin et al. 2002, Murphey et al. 2003a, Zabarankin et al. 2006) show how to model and solve a system of equations analytically for the case of a single threat and a single constraint on route length. However, formulating and solving such systems, analytically or numerically, is difficult given an arbitrary number of threats, and given multiple constraints on factors such as fuel consumption and flight time. (See also Inanc et al. 2004.)

Tsitsiklis (1995) and Polymenakos et al. (1998) describe reasonably efficient algorithms for finding "continuous shortest paths," but these algorithms do not easily extend to side-constrained problems or to problems with direction-dependent travel costs. Pfeiffer et al. (2005) develop a continuous, bi-objective model for minimizing risk and travel time along a path, subject to convex, polygonal threat regions. Based on an assumed sequence of n threats along any path, these authors identify an optimal path by solving a convex nonlinear program. However, the optimal sequence of threats will not normally be known, and finding a globally optimal path may therefore require the solution of $n!$ nonlinear programs.

A different type of continuous model describes routing as an obstacle-avoidance problem (Bortoff 2000, Helgason et al. 2001). But, an aircraft trying to reach a target cannot always avoid all threats, so these models could apply only in special cases.

Even if the shortcomings described above could be overcome, any continuous routing model that produces routes having smooth curves probably produces routes that are unflyable by a human pilot or a human UAV controller, or by a cruise missile using a "bang-bang" flight-control mechanism. In general then, we conclude that continuous routing models are unsuitable for use in autorouters.

A discrete routing model represents airspace using a network: Edges representing flight segments connect vertices in a three-dimensional grid embedded in airspace, although a two-dimensional grid will suffice for some situations. An edge's length represents the risk incurred by traversing the corresponding flight segment, or it represents a weighted sum of risk and other factors such as fuel consumption and travel time over that segment. Lewis (1988) appears to be the first to consider three-dimensional aircraft routing in discretized airspace. His discretization defines a network model that seeks a "minimum-cost" path with respect to a non-additive, composite measure of detection probability and fuel consumption; his nonlinear objective function necessitates a heuristic solution. But, a discrete model like Lewis's having a linear objective function will solve quickly using a standard, unconstrained shortest-path algorithm: The algorithm's output would be a route that is optimal for the composite measure being minimized (within the approximation entailed by the discretization), and any reasonable number of metrics can be combined in the objective function with modest computational effort. Unfortunately, this approach cannot be guaranteed to produce a route that minimizes one factor while satisfying a constraint on another.

Clearly, we would like to be able to place firm constraints on a mission with respect to fuel consumption, and/or elapsed time, and/or total risk, etc. Minimizing an additive risk measure, and constraining additive measures of the other factors in the airspace network, produces a constrained shortest-path problem (CSPP) (e.g., Lee 1995, Zabarankin et al. 2002, Murphey et al. 2003a, Zabarankin et al. 2006). CSPPs are NP-complete (Garey and Johnson 1979, p. 214), but numerous algorithms for solving them have been proposed and tested; for example, see Joksch (1966), Handler and Zang (1980), Aneja et al. (1983), Desrochers and Soumis (1988), Beasley and Christofides (1989), Lorenz and Raz (2001), Juttner et al. (2001), Van Mieghem et al. (2001), Korkmaz and Krunz (2001), Dumitrescu and Boland (2003), and Carlyle et al. (2007). (Kuipers et al. 2004 provides a general review of the topic.) Successful applications of these algorithms have appeared in such areas as transportation (Nachtigal 1995), commercial aircrew scheduling (Vance et al. 1997, Day and Ryan 1997), and signal routing in communication networks (Chen and Nahrstedt 1998, Kuipers et al. 2004).

The computational cost of solving the large-scale CSPPs that arise in aircraft routing has apparently restricted the use of such formulations in existing aircraft autorouters. Consequently, these autorouters have relied on computationally tractable, unconstrained, shortest-path approximations (e.g., Tharp 2005), or have applied a heuristic version of A* search to solve the CSPPs approximately (OPUS 2007).

Recently, however, Zabarankin et al. (2006) have applied the label-setting CSP algorithm of Dumitrescu and Boland (2003) to solve certain large-scale CSPPs for aircraft routing, and they report the best computational results on this topic to date. Nonetheless, much room remains for improvement. Zabarankin et al. first describe a radar-threat model (see Marcum 1947), which leads to an analytically tractable, continuous routing model as shown in an earlier paper (Murphey et al. 2003a). (See Leary 1995 for the application of a similar radar-threat model to an unconstrained, discrete aircraft-routing problem.) The model assumes a single radar threat, a single side constraint to limit route length, and an ellipsoid-shaped aircraft. The authors then build an analogous discrete model, and verify that it finds routes that correspond closely to the optimal routes provided by the continuous model. (They also present computational investigations of discrete models with two and three radars.) But, in doing this, the authors make no attempt to model (*i*) terrain avoidance, (*ii*) terrain-masking of threat radars, (*iii*) variable flight speed to improve threat avoidance, or (*iv*) more than a single side constraint. Furthermore, they (*v*) handle turn-radius constraints only heuristically (see Murphey et al. 2003b), (*vi*) test in a hypothetical airspace having an unrealistic geometry (the aircraft's maximum altitude is similar to the horizontal distance the aircraft must travel), and (*vii*) report long solution times (up to 1.5 hours for the heuristic, and up to three hours for the optimal algorithm).

It may be possible to overcome some of the omissions and difficulties, noted above, within the paradigm of Zabarankin et al. (2006). However, variable flight speed, extra side constraints, and turn constraints could substantially increase both storage requirements and the computational workload for a label-setting CSP algorithm. For example, avoiding the use of a heuristic to handle turn constraints requires an expanded network model (Caldwell 1961), or an algorithm having more complex vertex labels and a less stringent test for label dominance compared to the heuristic. Section 5.2 discusses this topic further.

This paper addresses the modeling omissions described above, and overcomes computational difficulties by applying a fast and versatile CSP algorithm. The basic algorithm, developed by Carlyle et al. (2007), combines Lagrangian relaxation and enumeration of near-shortest paths: Problems with more than one hundred thousand vertices and edges, and with up to ten side constraints, usually solve to optimality in a few minutes on a personal computer. Carlyle et al. show that this algorithm can be an order of magnitude faster than the label-setting algorithm of Dumitrescu and Boland (2003) used by Zabarankin et al. (2006). Additionally, as we will see in Section 5.2, our algorithm more easily extends to problems with turn constraints than does the label-setting algorithm.

Using the algorithm in Carlyle et al. (2005), we can describe and solve an aircraft-routing model that minimizes the risk of destruction from ground-based threats such as surface-to-air missiles (SAMs), while (*i*) placing firm limits on fuel consumption, or fuel consumption and flight time, and (*ii*) restricting turning radii, if desired. (A modified algorithm, as opposed to a modified network, ensures constraint satisfaction here.) Our modeling and computational tests cover the routing of both manned and unmanned aircraft.

The remainder of the paper is outlined as follows. The next section describes the CSP formulation for a generic aircraft-routing problem; Section 3 outlines the algorithm we use for solving CSPPs; Section 4 presents an aggressive network-reduction scheme to eliminate edges that can be proven not to lie on any optimal path; Section 5 presents two case studies; and Section 6 presents a summary and conclusions.

2 Constrained Shortest-Path Model for Aircraft Routing

We model the airspace in the area of operations (AO) by a directed network $G = (V, E)$ in which vertices $v \in V$ represent potential waypoints in three-dimensional space (two-dimensional in some cases), and directed edges $e = (u, v) \in E$ represent potential flight segments between distinct vertices $u, v \in V$. An aircraft, or group of aircraft, will fly from waypoint to waypoint along, and in the direction of, the specified edges. A suitable mesh and edge distribution must be selected based on aircraft characteristics such as maximum turn radius and/or climb rate, and on features of the operational environment such as threats and terrain features. We discuss this topic in detail for each application in section 5. (With a sufficiently fine mesh of vertices and sufficient density of edges, a discrete model can also identify a route that differs only negligibly from the optimal route produced by a continuous model; see Kim and Hespanha 2003. However, such a route could involve so many course corrections as to be unflyable.)

The aircraft will fly from some origin vertex s (e.g., a point designated for entering the AO), to some destination vertex t (e.g., a weapons launch point near a target), along a directed *s-t path*. This path is an ordered set of edges, $E_P = \{(s, v_1), (v_1, v_2), \ldots, (v_{k-1}, t)\}$. A path is *simple* if no vertices are repeated. A set of nonnegative real numbers measuring, for example, risk, traversal time, or fuel consumption is associated with each edge. And, a path's total risk, or time, or fuel consumption is evaluated simply as the sum of the relevant edge values. In the CSP model, one of these path measures will be minimized, while the others are constrained by upper bounds. We refer to the optimized measure as *length*; the other measures, indexed by $i \in I$, are *weights*. Let $c_e \geq 0$ and $f_{ie} \geq 0$, $i \in I$, denote length and weights of edge e, respectively. The length of path E_P is simply $\sum_{e \in E_P} c_e$ and the path's i-th weight is $\sum_{e \in E_P} f_{ie}$. For each $i \in I$, g_i prescribes an upper limit on a path's i-th weight.

We define the aircraft-routing problem as:

> Find a simple, directed, s-t path E_P^* in G such that $\sum_{e \in E_P^*} f_{ie} \leq g_i$ for all $i \in I$, and such that $\sum_{e \in E_P^*} c_e$ is minimum over all s-t paths E_P.

In a general context, this problem is known as the (resource-)constrained shortest-path problem (CSPP).

The CSP model is certainly reasonable for a cruise missile that makes a one-way trip from origin to destination, but a human pilot also wishes to make the return trip. A valuable UAV should make the return trip as well. Generally, doctrine and common sense prescribe different ingress and egress routes to a target. In particular, airspace controllers often specify a certain airspace corridor for ingress and another for egress to avoid enemy fire as well as accidents and friendly fire (Zacherl 2006). With such separation, the CSP model can determine an optimal, round-trip flight path by simply using a directed network consisting of two sub-networks. The first sub-network represents ingress routes from s to t, while the second represents egress routes from its source at t to its sink at s', which could be a duplicate of s. Because the two sub-networks are essentially disjoint, the optimal path from s to s' solves the joint, ingress-egress problem. Section 5.5 provides a computational example to illustrate.

3 Solving the Constrained Shortest-Path Problem

Carlyle et al. (2007) develop an efficient, exact algorithm for solving certain CSPPs, and we intend to apply that algorithm for routing military aircraft. For completeness, this section presents the essence of that algorithm. The algorithm is called "Lagrangian relaxation plus (near-shortest path) enumeration," or "LRE" for short. The LRE algorithm is related to the Lagrangian-based algorithms of Handler and Zang (1980) and Beasley and Christofides (1989). Its implementation also exploits and extends the preprocessing procedures of Aneja et al. (1983), Beasley and Christofides (1989), and Dumitrescu and Boland (2003).

Let A denote the vertex-edge incidence matrix for a directed graph $G = (V, E)$: For each $e = (v, u) \in E$, $A_{ue} = 1$, $A_{ve} = -1$, and $A_{we} = 0$ for any $w \neq u, v$. Let $\mathbf{b}$ denote the $|V|$-vector such that $b_s = 1$, $b_t = -1$ and $b_v = 0$ for all $v \in V \backslash \{s, t\}$. And, define this additional notation:

$$\mathbf{g} = (g_1\, g_2 \cdots g_{|I|})^T,\ \ \mathbf{c} = (c_1\, c_2 \cdots c_{|E|}),\ \ \mathbf{f}_i = (f_{i1}\, f_{i2} \cdots f_{i|E|}),\ \text{and}\ F = \begin{bmatrix} \mathbf{f}_1 \\ \mathbf{f}_2 \\ \vdots \\ \mathbf{f}_{|I|} \end{bmatrix}.$$

Then, CSPP may be written as an integer program (Ahuja et al. 1993, p. 599),

$$\begin{aligned} \textbf{CSPIP} \quad z^* \ \equiv \ & \min_{\mathbf{x}\in\{0,1\}^{|E|}} \mathbf{cx} && (1)\\ \text{s.t. } A\mathbf{x} \ &= \ \mathbf{b} && (2)\\ F\mathbf{x} \ &\leq \ \mathbf{g}, && (3) \end{aligned}$$

where $x_e^* = 1$ if edge e is in the selected optimal path, and $x_e^* = 0$, otherwise. We refer to constraints (3) as *side constraints,* and refer to $\hat{\mathbf{x}}$ as a *path* when it satisfies all constraints of **CSPIP** except possibly the side constraints. (The potential for cycles in an optimal path $\hat{\mathbf{x}}$ can be safely ignored because $\mathbf{c} \geq \mathbf{0}$ and because of the structure of our solution algorithm.)

Using the standard theory of Lagrangian relaxation (e.g., Ahuja et al. 1993, pp. 598-648), we know that for any appropriately dimensioned row vector $\boldsymbol{\lambda} \geq \mathbf{0}$,

$$\begin{aligned} z^* \ \geq \ \underline{z}(\boldsymbol{\lambda}) \ \equiv \ & \min_{\mathbf{x}\in\{0,1\}^{|E|}} \mathbf{cx} \ + \ \boldsymbol{\lambda}(F\mathbf{x} - \mathbf{g}) && (4)\\ \text{s.t. } A\mathbf{x} \ &= \ \mathbf{b}. && (5) \end{aligned}$$

Rewriting the objective function, we can optimize the Lagrangian lower bound $\underline{z}(\boldsymbol{\lambda})$ through

$$\begin{aligned} \textbf{CSPLR} \quad \underline{z}^* \ \equiv \ & \max_{\boldsymbol{\lambda}\geq\mathbf{0}} \underline{z}(\boldsymbol{\lambda}) && (6)\\ = \ & \max_{\boldsymbol{\lambda}\geq\mathbf{0}} \ \min_{\mathbf{x}\in\{0,1\}^{|E|}} (\mathbf{c} + \boldsymbol{\lambda}F)\mathbf{x} - \boldsymbol{\lambda}\mathbf{g} && (7)\\ \text{s.t. } A\mathbf{x} \ &= \ \mathbf{b}. && (8) \end{aligned}$$

For any fixed $\boldsymbol{\lambda} \geq 0$, computing the lower bound $\underline{z}(\boldsymbol{\lambda})$ simply requires the solution of a shortest-path problem with Lagrangian-modified edge lengths. An integer optimal solution exists for the linear-programming relaxation of the inner minimization of **CSPLR**, so we know that $\underline{z}^*$ equals the optimal objective value of the linear-programming relaxation of **CSPIP** (e.g., Fisher 1981). Unfortunately, it is easy to construct examples in which this bound greatly underestimates z^* (see Lee 1995 and section 5 in the current paper), so the success of the LRE approach can depend on the ability to quickly close a large duality gap.

The outer maximization over $\boldsymbol{\lambda}$ can be solved by numerous methods; for instance, see Fox and Landi (1971), Beasley and Christofides (1989), DeWolfe et al. (1993), Wolsey (1998, pp. 172-173). We use repeated bisection search in the coordinate directions because we expect to have only a few side constraints, and because this technique seems to work well for such cases (DeWolfe et al. 1993, Carlyle et al. 2007).

The LRE algorithm also requires an upper bound, $\bar{z} \geq z^*$. Any path $\hat{\mathbf{x}}$ that satisfies the side constraints (3) yields such a bound, $\bar{z} = \mathbf{c}\hat{\mathbf{x}}$. Such paths often appear as a byproduct of optimizing $\underline{z}(\boldsymbol{\lambda})$, especially if the problem possesses only a few side constraints. But, a special "phase-I algorithm" can also be used to identify a feasible path, if necessary (Carlyle et al. 2007). In the worst case, $\bar{z} = (|V| - 1) \max_{e \in E} c_e$ is always valid for a feasible problem.

Now, given $\bar{z}$, and an arbitrary Lagrangian vector $\boldsymbol{\lambda} \geq \mathbf{0}$, the following theorem and corollary show that we may view the problem of solving **CSPIP** as one of simple path enumeration. Carlyle et al. (2007) prove this theorem explicitly, but it may be found implicitly in Handler and Zang (1980).

Theorem 1 *All optimal solutions* $\mathbf{x}^*$ *to* **CSPIP** *are contained in the set* $\hat{X}(\boldsymbol{\lambda}, \bar{z})$, *where* $\bar{z}$ *denotes an upper bound on* z^* *for* **CSPIP**, *and* $\hat{X}(\boldsymbol{\lambda}, \bar{z})$ *denotes the set of feasible paths* $\hat{\mathbf{x}}$ *to* **CSPLR** *that satisfy* $\mathbf{c}\hat{\mathbf{x}} + \boldsymbol{\lambda}(F\hat{\mathbf{x}} - \mathbf{g}) \leq \bar{z}$.

Proof: Since $F\mathbf{x}^* \leq \mathbf{g}$ and $\boldsymbol{\lambda} \geq \mathbf{0}$, the result follows from the facts that (i) $\mathbf{c}\mathbf{x}^* + \boldsymbol{\lambda}(F\mathbf{x}^* - \mathbf{g}) \leq z^*$, and (ii) $z^* \leq \bar{z}$. ∎

Corollary 1 *If* **CSPIP** *is feasible, an optimal solution* $\mathbf{x}^*$ *can be identified by (a) establishing an upper bound* $\bar{z} \geq z^*$, *(b) enumerating* $\hat{\mathbf{x}} \in \hat{X}(\boldsymbol{\lambda}, \bar{z})$, *and (c) selecting*

$$\mathbf{x}^* \in \operatorname*{argmin}_{\hat{\mathbf{x}} \in \hat{X}(\boldsymbol{\lambda}, \bar{z})} \{\mathbf{c}\hat{\mathbf{x}} \mid F\hat{\mathbf{x}} \leq \mathbf{g}\}. \tag{9}$$

∎

Theorem 1 and Corollary 1 are valid for any $\boldsymbol{\lambda} \geq \mathbf{0}$, but it is easy to devise examples that show how an optimal or near-optimal $\boldsymbol{\lambda}$ for **CSPLR** can exponentially reduce the size of $\hat{X}(\boldsymbol{\lambda}, \bar{z})$, which reduces the computational workload correspondingly. Thus, we do attempt to maximize $\underline{z}(\boldsymbol{\lambda})$ but, for simplicity, use heuristic stopping rules for the maximization process. We have verified on medium-sized problems, through direct solution of linear programs, that these rules typically maximize the Lagrangian bound to within 1% of the optimal value.

The theorem and corollary also imply that we may need to enumerate each path $\hat{\mathbf{x}}$ satisfying $(\mathbf{c} + \boldsymbol{\lambda} F)\hat{\mathbf{x}} - \boldsymbol{\lambda}\mathbf{g} \leq \bar{z}$. Let $\mathbf{x}^*_{\boldsymbol{\lambda}}$ solve the shortest-path problem given the edge-length vector $\mathbf{c} + \boldsymbol{\lambda} F$ so that $\underline{z}(\boldsymbol{\lambda}) = (\mathbf{c} + \boldsymbol{\lambda} F)\hat{\mathbf{x}}^*_{\boldsymbol{\lambda}} - \boldsymbol{\lambda}\mathbf{g}$. Then, we can solve CSPP by enumerating all paths $\hat{\mathbf{x}}$ such that $\underline{z}(\boldsymbol{\lambda}) \leq (\mathbf{c} + \boldsymbol{\lambda} F)\hat{\mathbf{x}} - \boldsymbol{\lambda}\mathbf{g} \leq \bar{z}$. In turn, this means that, given edge-length vector $\mathbf{c} + \boldsymbol{\lambda} F$, and adding the Lagrangian constant term $-\boldsymbol{\lambda}\mathbf{g}$ to the length of any path, we wish to find all δ-optimal (near-shortest) paths for $\delta \equiv \bar{z} - \underline{z}(\boldsymbol{\lambda})$. Of course, $\bar{z}$ may change as the algorithm identifies new feasible solutions, so δ may change; and, if δ ever goes to 0, the algorithm can halt. The full LRE algorithm can now be outlined.

LRE Algorithm for CSPP (Outline)

Input: $G = (V, E)$, s, t, $\mathbf{c}$, $\mathbf{g}$, and F defining a CSPP.

Output: An optimal path-edge incidence vector $\hat{\mathbf{x}}$.

Step 1: Find $\boldsymbol{\lambda}$ that optimizes or approximately optimizes the Lagrangian lower bound $\underline{z}(\boldsymbol{\lambda})$.

Step 2: Let $\hat{X}$ denote the set of feasible paths identified while optimizing $\underline{z}(\boldsymbol{\lambda})$. If $\hat{X} \neq \emptyset$, set upper bound $\bar{z} \leftarrow \min_{\hat{\mathbf{x}} \in \hat{X}} \mathbf{c}\hat{\mathbf{x}}$, else set $\bar{z} \leftarrow (|V| - 1)c_{\max} + \gamma$ for some $\gamma > 0$.

Step 3: Using a standard path-enumeration procedure (e.g., Byers and Waterman 1984), begin enumerating all paths $\hat{\mathbf{x}}$ such that $(\mathbf{c} + \boldsymbol{\lambda} F)\hat{\mathbf{x}} - \boldsymbol{\lambda}\mathbf{g} \leq \bar{z}$, with the following modifications:

(a) Use $\bar{z}$ and the side constraints to limit the enumeration when it can be projected that the current path cannot be extended to one whose length improves upon $\bar{z}$ or which does not violate at least one of the side constraints.

(b) Whenever the algorithm identifies a feasible path $\hat{\mathbf{x}}$ whose length is shorter than the incumbent, update the incumbent to $\hat{\mathbf{x}}$ and update the upper bound to $\bar{z} = \mathbf{c}\hat{\mathbf{x}}$.

Step 4: If no $\hat{\mathbf{x}}$ is found in Step 3, the problem is infeasible; otherwise the best solution $\hat{\mathbf{x}}$ is optimal.

End of LRE Algorithm

The path-enumeration procedure initializes itself in Step 3 by computing distances from every vertex to t using "Lagrangian edge lengths" (to be defined below), true edge lengths, and individual edge weights. Specifically, Step 3 starts by

- Computing the minimum "Lagrangian distance" $d(v)$ from each $v \in V$ to t by solving a single shortest-path problem that traverses edges backwards, starting from t, using Lagrangian edge lengths $\mathbf{c}' \equiv \mathbf{c} + \boldsymbol{\lambda} F$,

- Computing analogous minimum v-to-t distances $d_0(v)$ for all $v \in V$ with respect to edge lengths $\mathbf{c}$, and

- Computing analogous minimum v-to-t distances $d_i(v)$ for all $v \in V$ and $i \in I$ with respect to edge weights $\mathbf{f}_i$.

This initialization phase requires the solution of only $|I| + 2$ shortest-path problems.

Let $E_P(u) = \{(s, v_1), (v_1, v_2), \ldots, (v_{k-1}, u)\}$ denote a *directed s-u subpath.* In Step 3 of the algorithm, a standard path-enumeration procedure commences from s, but extends subpath $E_P(u)$ along edge $e = (u, v)$ if and only if the following conditions hold:

Conditions for extending a subpath

- $E_P(u) \cup \{e\}$ can be extended to a path whose Lagrangian length does not exceed $\bar{z}$, i.e., $L(u) + (c_e + \sum_{i \in I} \lambda_i f_{ie}) + d(v) \le \bar{z}$, where $L(u)$ denotes the Lagrangian length of $E_P(u)$ and where, by convention, we define $L(s) = -\boldsymbol{\lambda}\mathbf{g}$.

- $E_P(u) \cup \{e\}$ can be extended to a path with length strictly less than $\bar{z}$, i.e., $L_0(u) + c_e + d_0(v) < \bar{z}$, where $L_0(u)$ denotes the length of $E_P(u)$.

- For all $i \in I$, $E_P(u)$ can be extended to a path whose i-th weight does not exceed g_i, i.e., $L_i(u) + f_{ie} + d_i(v) \le g_i$ for all $i \in I$, where $L_i(u)$ denotes the i-th total weight of $E_P(u)$.

- The path does not loop back on itself.

The LRE algorithm actually defines a branch-and-bound procedure that incorporates a depth-first enumeration tree along with feasibility checks. (This may also be viewed as non-heuristic variant of A* search; for example, see Russell and Norvig 1995, pp. 92-107.)

Branching consists of extending the current subpath by one edge. Carlyle et al. (2007) show the usefulness of several enhancements to the LRE algorithm, including (*i*) the application of a network-reduction procedure at several places in the algorithm to remove edges that cannot possibly lie on an optimal path (discussed further in the next section), (*ii*) the addition of conditions based on aggregated constraints to limit path enumeration in Step 3(a) of the algorithm, and (*iii*) the use of a phase-I routine for finding initial feasible paths. We recommend these enhancements as they can improve solution times dramatically for some problems, and because they do not incur significant overhead in practice. Naturally, computational work can also be reduced by accepting an ϵ-optimal solution, i.e., by halting the algorithm as soon as $\bar{z} - \underline{z}(\boldsymbol{\lambda}) \leq \epsilon$, for some prespecified $\epsilon > 0$. Or, as in our computational tests, the algorithm can halt when a relative optimality tolerance of $r\%$ is reached: $(\bar{z} - \underline{z}(\boldsymbol{\lambda}))/\underline{z}(\boldsymbol{\lambda}) \leq r/100\%$.

4 Network Reductions

A network-reduction procedure for CSPP may be able to identify numerous vertices and edges that cannot lie on any optimal path, and remove them prior to optimization. The resulting, smaller network should require less effort to solve, simply because there are fewer vertices and edges to process. Importantly, a smaller network may also yield a tighter Lagrangian bound as well as tighter distances $d(v), d_0(v)$, and $d_i(v), i \in I$, for the path enumeration procedure. Aneja et al. (1983) apply network reductions with respect to the individual edge weights, while Beasley and Christofides (1989) and Ziegelmann (2001) also apply these with respect to edge length and Lagrangian edge length. Those authors apply network reductions only before the main algorithm begins, so these reductions are typically referred to as "preprocessing." Dumitrescu and Boland (2003) preprocess with respect to individual edge weights and edge lengths, but repeat the process multiple times. We use the following network-reduction procedure at several different points in our algorithm (see Carlyle et al. 2007). Note that the procedure generalizes the techniques described above by also using "average edge weight" $\sum_{i \in I} f_{ie}/g_i$.

Network Reduction Procedure

Input: Data for CSPP and number of scans n_s.

Step 1: Set $h \leftarrow 1$.

Step 2: For all $i \in I$, and for all $v \in V$, compute a minimum-weight s-v subpath distance $D_i(v)$ and a minimum-weight v-t subpath distance $d_i(v)$ with respect to weight vector $\mathbf{f}_i$.

Step 3: For all $v \in V$, compute a minimum-average-weight s-v subpath distance $\bar{D}(v)$ and a minimum-average weight v-t subpath distance $\bar{d}(v)$ with respect to "average weight" vector $\sum_{i \in I} \mathbf{f}_i / g_i$.

Step 4: For all $v \in V$, compute a minimum-length s-v subpath distance $D_0(v)$ and a minimum-length v-t subpath distance $d_0(v)$ with respect to length vector $\mathbf{c}$.

Step 5: For all $v \in V$, compute a minimum-Lagrangian-length s-v subpath distance $D(v)$ and a minimum-Lagrangian-length v-t subpath distance $d(v)$ with respect to weight vector $\mathbf{c} + \boldsymbol{\lambda} F$.

Step 6: Delete any edge $e = (u, v) \in E$ such that

$$D_i(u) + f_{ie} + d_i(v) > g_i \text{ for any } i \in I, \text{or} \tag{10}$$

$$\bar{D}(u) + \sum_{i \in I} f_{ie}/g_i + \bar{d}(v) > |I|, \text{or} \tag{11}$$

$$D_0(u) + c_e + d_0(v) \geq \bar{z}, \text{or} \tag{12}$$

$$D(u) - \boldsymbol{\lambda}\mathbf{g} + c_e + \sum_{i \in I} \lambda_i f_{ie} + d(v) > \bar{z}. \tag{13}$$

Step 7: If $h < n_s$ and at least one edge was deleted in Step 6, set $h \leftarrow h + 1$, and go to Step 2. Else, stop.

A similar network-reduction procedure for eliminating vertices can also be constructed (Aneja et al. 1983 and Dumitrescu and Boland 2003), but the edge-elimination procedure subsumes it, and computational time is negligible.

Dumitrescu and Boland (2003) propose scanning edges multiple times in their preprocessing procedure, which corresponds to setting $n_s > 1$ in our network-reduction procedure. Repeated scanning may reduce the network further than a single scan since the removal of edges in Step 6 can lead to longer distances in Steps 2-5. Empirically, we find little value in scanning the set of edges more than 10 times and therefore set $n_s = 10$. Aneja et al. (1983), Beasley and Christofides (1989), Ziegelmann (2001), and Dumitrescu and Boland (2003) apply network reductions prior to any calculations or after optimizing the Lagrangian lower bound. Carlyle et al. (2007) follow suit, but also experiment with "reprocessing," which repeatedly applies network reductions within the path-enumeration phase of the algorithm. The current paper adopts an aggressive network-reduction scheme that applies reductions (*i*) before Step 1 of the LRE algorithm, with $n_s = 10$, (*ii*) immediately after Step 1 (i.e., after optimizing the Lagrangian lower bound) with $n_s = 10$, and then (*iii*) during Step 3 (i.e., within the path-enumeration phase of the algorithm), every time $\bar{z}$ reduces by a multiplicative factor of α, but only with $n_s = 1$. We set $\alpha = 0.9$ in all numerical tests.

Since only a weak upper bound is available prior to Step 1, the first application of network reductions effectively utilizes only the side constraints. However, as successively tighter upper bounds are found while optimizing the Lagrangian lower bound or enumerating paths, the reductions becomes more effective and may shrink the network dramatically.

5 Applications

This section presents two case studies for military-aircraft routing, the first for an F/A-18 strike mission and the second for a UAV surveillance mission. We also demonstrate how to enforce turn-radius constraints, when needed. Computational results are obtained using the LRE algorithm as described above, but with the addition of aggregated constraints and the phase-I procedure from Carlyle et al. (2007) (not described in the current paper). We carry out computations on a desktop computer with a 3.4 GHz Intel Pentium IV processor, 3 gigabytes of RAM, the Microsoft Windows XP operating system, and with programs written and compiled using Microsoft Visual C++ Version 6.0.

5.1 Routing an F/A-18 Strike Mission

Planners wish to determine a fuel-constrained, minimum-risk route for an F/A-18 strike group from an entry point in the area of operations (AO), through enemy airspace to a specific destination such as a weapons-launch point. A strike group consists of multiple aircraft types such as fighters, radar jammers, and the primary strike aircraft, several F/A-18s in this case. The aircraft risk being shot down by enemy surface-to-air missiles (SAMs), and are subject to a limit on fuel consumption.

We formulate this routing problem as a singly constrained CSPP on a two-dimensional network consisting of a highly connected grid of vertices. Edge length c_e measures the risk (to be defined precisely below) of traveling along e, while edge e's weight $f_e = f_{1e}$ measures fuel consumption along e, with the Euclidean length of the edge used as a surrogate. The AO's limits are defined, in part, by the closest points to the destination at which the strike group might complete aerial refueling. Current doctrine specifies that F/A-18 and similar aircraft will maintain a constant, fuel-efficient altitude of about 36,000 feet, so a two-dimensional grid suffices to model the AO's airspace.

The AO covers an area of 200 nautical miles (nm) by 296 nm, laid out in a Cartesian coordinate system with the origin at the southwest corner; see Figure 1. We cover the airspace with a 26×38 rectangular grid of vertices, which implies a spacing of eight nm. This spacing corresponds to about one minute of flying time at the standard cruising speed of Mach 0.8. (See Kim and Hespanha 2003 for experiments with non-rectangular grids.) The strike group will enter the AO at the AO's western edge, at coordinates (0,104), and fly in a generally easterly direction to the destination at coordinates (296,104).

Graphically, the threat from a single SAM, with known location, can be represented as a set of concentric "threat circles," centered on the SAM's location. The central circle defines the region of highest risk around the SAM, and risk decreases, stepwise, in each annulus-shaped region further from the center. Clearly, this represents an idealized threat model, but it does reflect the current level of detail in military planning (Bindi and McCarthy 2004, Landon 2004), and more elaborate formulations are easily incorporated into the flexible CSP methodology.

Intelligence reports may not be able to locate some SAMs precisely, especially in the case of mobile SAMs. In this case, we could increase the radii of the concentric circles and decrease the corresponding risk measure to reflect the more diffuse risk. Other shapes could also be used: For instance, if a mobile SAM were spotted on a straight-line road segment some hours before a strike mission is to commence, a cigar-shaped region along the road might be appropriate.

We compute an additive risk measure c_e, for each edge e, based on the probability p_e that at least one SAM hits the strike group as the group traverses e. We compute p_e

as a as a function of the geometric length of each threat-circle intersection and associated "threat magnitudes," assuming that p_e does not depend on the subpath used to reach e. This "independence assumption" would be inappropriate if a SAM that could strike an aircraft on edge e'' could capitalize on the tracking information provided by radars associated with some edge $e' \neq e''$ that might appear earlier along the strike group's route. But, because terrain-masking cannot be exploited by a high-altitude strike group, mission planners actually expect that the enemy's long-range radar will accurately track the group. Consequently, threats to the strike group are local and independent, and p_e depends on the group's ability to jam targeting radars and to avoid any missiles that are fired at it. The independence assumption only fails here if the threat from a single SAM influences the calculation of $p_{e'}$ and $p_{e''}$ on two separate edges, e' and e'', along a path the group might traverse. But, assuming all nominal probabilities are reasonably small, it can be shown that the error induced is modest.

Given p_e for every edge e, and given the independence assumption, the probability of no SAMs hitting the strike group while traversing a path E_P is simply $\prod_{e \in E_P}(1 - p_e)$. Using a standard logarithmic transformation (Shorack 1964), we obtain the risk measure $c_e = -\log(1 - p_e)$ such that minimizing $\sum_{e \in E_P} c_e$ maximizes that product, i.e., a minimum-risk path is equivalent to a path with maximum probability of no aircraft being hit by a SAM. In the following, we report this probability of no hits, which we equate with "probability of mission success." (Of course, risk could be limited by a constraint in the CSP model, while some other objective such as time is minimized.)

Our test data define 15 SAM sites in the AO, each surrounded by two or three threat circles, with various radii. These radii depend on the technical capabilities of the corresponding SAM and its tracking radar. Figure 1 depicts the threat-circle boundaries as dotted circles inside the AO.

The simplest discretization of the AO might connect nearest-neighbor vertices, including diagonals, with edges. The resulting network would be sparse and the computational burden low, but it could lead to unrealistically jagged flight paths. On the other hand, modeling straight-line flight segments between every vertex pair would yield a dense, complete network with about 2×10^6 edges, and a high computational burden. Consequently, we explore eight different graph structures (A-H in Table 1), which are much denser than typical network models such as road networks (and much denser than the topologies employed by Zabarankin et al. 2006), but substantially sparser than a complete network. For instance, Structure A connects each vertex u to all vertices v that are between 8 nm and 12 nm away, but only those that are no further west than u. In fact, none of the networks in Table 1 includes edges with any west-bound vector component. We justify models with no short edges (see F, G, and H in Table 1) by the fact that short edges may result in routes with many edges and consequently the potential for frequent zig-zagging. Zig-zagging is undesirable from the pilot's perspective because of the associated work load. Note that some of the edges we generate, especially certain long ones, could be excluded from the network *a priori*, because they imply paths that clearly consume too much fuel. However, for simplicity in this paper, we eliminate those through network reductions, *a posteriori*.

Graph Structure	$\lvert V\rvert$	$\lvert E\rvert$	Edge lengths (nm) min	Edge lengths (nm) max	Prob. of mission success for various fuel limits g (fuel measured in nm) 300	310	320	330	340	350
A	988	4,712	8	12	0.1892	0.4738	0.5337	0.7953	0.8940	0.9222
B	988	11,048	8	18	0.3517	0.5793	0.8949	0.9240	0.9268	0.9287
C	988	22,222	8	30	0.3517	0.6225	0.9094	0.9259	0.9277	0.9305
D	988	123,166	8	80	0.5310	0.7254	0.9185	0.9268	0.9287	0.9305
E	988	228,042	8	120	0.5310	0.7261	0.9185	0.9268	0.9287	0.9305
F	988	223,330	16	120	0.5247	0.7261	0.9185	0.9268	0.9287	0.9305
G	988	195,110	40	120	0.5092	0.7045	0.9066	0.9235	0.9235	0.9235
H	988	118,454	16	80	0.5247	0.7254	0.9185	0.9268	0.9287	0.9305

Table 1: Statistics for strike-group routing given various fuel constraints. Each vertex u is connected with edges (u, v) where v lies between "min edge" and "max edge" nautical miles distant from u, but is no further west than u. Using a 1% relative optimality tolerance, the last six columns specify the probabilities of success for the near-optimal routes given various fuel limits g. These fuel limits correspond to the Euclidean distance traveled, measured in nautical miles. Figure 1 illustrates some of these routes.

The last six columns of Table 1 show that different network densities do affect the calculated probability of mission success. Naturally, a denser graph allows more flexibility and a route with higher probability of success (lower risk) is possible. It appears that graph structures F and H allow reasonable flexibility in flight planning with, as we shall see below, small computational effort. Note also that the tighter fuel limits dramatically reduce the probabilities of mission success, to levels at which the missions might not be executed.

For various fuel limits, Table 2 reports the solution time ("Run time"); the relative "initial gap" which provides a measure of the quality of the initial solution found ("Ini. gap"); and the relative duality gap ("Dual gap"). We define the relative initial gap as

$100\%(\mathbf{c}\hat{\mathbf{x}} - \underline{z}^*)/\underline{z}^*$, where $\hat{\mathbf{x}}$ denotes the best feasible solution found while optimizing $\underline{z}(\boldsymbol{\lambda})$, and we define the relative duality gap as $100\%(z^* - \underline{z}^*)/\underline{z}^*$. Note that the minimum fuel consumption for the group is 296 and the optimality tolerance is 1%. Table 2 shows that a problem from this class can, in fact, exhibit large initial gaps and large duality gaps. And, note that a problem with a small duality gap but a large initial gap may require a significant amount of enumeration, presumably because of a weak upper bound.

The edge weights, which represent fuel consumption, vary significantly among the graph structures B-H. This motivates us to examine the path-enumeration procedure within the LRE algorithm and its potential sensitivity to edge-processing order, i.e., to the order in which the enumeration mechanism scans the edges directed out of any vertex. In fact, we find that efficiency tends to improve when the algorithm processes edges in order of decreasing weight, rather than in some arbitrary order. The improvement seems to derives from the greater likelihood of finding good feasible solutions quickly. Roughly speaking, when the algorithm uses this rule, it searches for s-t paths with the fewest edges first, and thus spends less effort per path investigated in its early phases. If a path with only a few edges is just as likely to be a good feasible path as a path with many edges—and we may have no reason to believe otherwise—then, on average, the algorithm using this rule will find more good paths quickly. Consequently, all tests reported use this scheme. (This ordering scheme may be viewed as a static "branching strategy" for the underlying branch-and-bound algorithm.) We typically observe only moderate sensitivity of solution times to edge-processing order, but two instances do show order-of-magnitude improvements with the reordering.

Figure 1 illustrates some of the minimum-risk paths for the "F network" (see numerical results in Table 1). The figure clearly shows how, as the fuel limit increases, the near-optimal path becomes longer and more indirect in order to improve the probability of mission success. The second and third columns of Table 3 list the probabilities of mission success and actual fuel-consumption values, respectively, for various fuel limits. (Some of these results are also reported in Table 1.) Initially, the probability of success increases substantially as the fuel limit increases from 300 to 330. This probability does not improve much with greater fuel limits, however, because the last part of the route must fly through an unavoidable threat region.

The model above assumes constant aircraft speed along the mission's route. This is a

Fuel limit		Run time and Gap							
(fuel limit)	Statistic	A	B	C	D	E	F	G	H
300	Run time (sec.)	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
	Ini. gap (%)	<1	<1	<1	<1	<1	<1	<1	<1
	Dual gap (%)	<1	<1	<1	<1	<1	<1	<1	<1
310	Run time (sec.)	0.0	0.0	0.0	0.5	0.5	0.4	0.4	0.3
	Ini. gap (%)	3	4	3	264	260	260	136	262
	Dual gap (%)	3	4	2	117	117	117	43	117
320	Run time (sec.)	0.0	0.0	0.0	0.2	0.5	0.5	0.2	0.3
	Ini. gap (%)	7	<1	32	41	41	41	8	41
	Dual gap (%)	7	<1	10	4	4	4	7	4
330	Run time (sec.)	0.0	0.0	0.0	0.1	0.1	0.1	0.1	0.1
	Ini. gap (%)	10	1	<1	<1	<1	<1	<1	<1
	Dual gap (%)	10	<1	<1	<1	<1	<1	<1	<1
340	Run time (sec.)	0.0	0.0	0.0	0.2	0.4	0.3	0.2	0.2
	Ini. gap (%)	4	2	2	2	2	2	2	2
	Dual gap (%)	1	<1	<1	<1	<1	<1	2	<1
350	Run time (sec.)	0.0	0.0	0.0	0.2	0.3	0.4	0.3	0.2
	Ini. gap (%)	3	1	<1	1	1	1	4	1
	Dual gap (%)	3	1	<1	1	1	1	4	1

Table 2: Computational results for routing an F/A-18 strike group. For various fuel limits, this table reports solution time ("Run time"), initial solution quality ("Ini. gap") and duality gap ("Dual gap"). Given a initial feasible solution $\hat{\mathbf{x}}$, initial solution quality is defined as $100\%(\mathbf{c}\hat{\mathbf{x}} - \underline{z}^*)/\underline{z}^*$; duality gap is defined as $100\%(z^* - \underline{z}^*)/\underline{z}^*$. The optimality tolerance is 1%.

	Constant speed		Variable speed	
Fuel limit	Prob. of	Fuel consumed	Prob. of	Fuel consumed
(nm)	success	(modified nm)	success	(modified nm)
300	0.5247	300.0	0.5247	300.0
310	0.7261	310.0	0.7261	310.0
330	0.9268	328.8	0.9268	328.8
340	0.9287	339.8	0.9308	339.9
360	0.9319	358.7	0.9473	359.7
370	0.9335	370.0	0.9592s	369.3
390	0.9340	384.8	0.9697	389.3
400	0.9340	397.9	0.9776	399.9
420	0.9340	416.1	0.9915	419.2
430	0.9340	416.1	0.9965	429.4
∞	0.9340	397.9	1.0000	710.6

Table 3: Minimum-risk routing for an F/A-18 strike group with constant or variable speeds. This table compares near-optimal routes in the "F network" for various fuel limits assuming constant-speed and variable-speed paths, and using a 1% optimality tolerance. Fuel is measured in "modified nm" for the variable-speed model because the use of a high-speed edge consumes twice the fuel of its standard-speed counterpart. All run times are less than 3 seconds. Note that rows with identical success probabilities but different fuel-consumption values represent cases with multiple near-optimal solutions. (See Figures 1 and 2.)

realistic assumption for missions with uniformly low risk, but a pilot may wish to traverse a high-risk region at a higher-than-normal speed: Higher speeds enable more effective evasive maneuvers against a SAM that is actually fired at an aircraft (Landon 2004). To account for variable speeds, we add a parallel edge e' for each original edge $e \in E$. The original edge e corresponds to flying at a standard cruising speed of about Mach 0.8, as used in the constant-speed examples, while the parallel edge e' corresponds to flying at a higher speed to improve threat avoidance. Since a low-threat region requires no special actions for threat avoidance, we include e' only when $p_e \geq 0.1$. For purposes of demonstration, we represent reduced risk on a high-speed edge by defining $1 - p_{e'} = \min\{1.2(1 - p_e), 1\}$, and reflect increased fuel consumption at high speed by setting $f_{e'} = 2f_e$.

Columns four and five of Table 3 show the probability of mission success and total fuel consumption, respectively, for the variable-speed solutions on the F network. The variable-speed F network contains 318,890 edges compared to 223,330 for the constant-speed network (see Table 1). Run times increase slightly for the variable-speed network, but no solution in Table 3 requires more than three seconds. For tight fuel limits, the pilot cannot increase speed and the probability of success remains unchanged. However, for fuel limits greater than

330, temporarily increasing speed becomes a viable option, and the probability of success improves.

Figure 2 depicts the minimum-risk routes with a fuel limit of 370 for a constant speed (solid line) and for variable speed (dashed line). The constant-speed solution involves a long detour to exploit a marginally safer approach to the destination—compare the near-optimal route for fuel limit 340 shown in Figure 1—while the variable-speed solution conserves fuel initially for a final, high-speed, direct approach to the destination.

5.2 Turn-Radius Constraints

Any aircraft has a limited turning radius. The 90-degree turn in Figure 1 is a reasonable approximation of reality for a highly maneuverable F/A-18 at that figure's scale of hundreds of nautical miles. However, other aircraft such as cruise missiles are less maneuverable than strike and fighter aircraft, and they may also be controlled at a much finer scale. We may therefore wish to impose "turn-radius constraints," or simply "turn constraints," on an aircraft's route that limit all turn angles to θ degrees or less, for some predefined constant $\theta > 0$ (Boroujerdi and Uhlmann 1998, Helgason et al. 2001).

Zabarankin et al. (2006) incorporate turn constraints by modifying their label-setting CSP algorithm. This enforces realistic constraints, but a detailed description of the modified algorithm in Murphey et al. (2003b) reveals that it is a heuristic, not an exact algorithm. The heuristic maintains non-dominated risk-distance labels at each vertex, and records the standard predecessor-vertex datum for each label. The predecessor data are used to ensure that no label is updated by following an edge whose traversal would require an overly sharp turn. An exact algorithm would require a three-part vertex label that includes the predecessor vertex, and would apply dominance tests only to labels having the same predecessor vertex.

The classical exact method of incorporating turn constraints in a network-routing problem (Caldwell 1961) first "expands" each vertex v by adjacent vertices v' that might precede v in a path; let $<v', v>$ denote such an expanded vertex. If the original network has edges (v', v) and (v, v''), and the turn angle involved in flying the corresponding flight segments is not too sharp, then and only then is an "expanded edge" created, $(<v', v>, <v, v''>)$. (Actually, Caldwell adds penalties that depend on the turn angle.) Thus, if we were to modify our network's topology and numerical data appropriately, and then apply the LRE algorithm to that network, we would have handled turn constraints for aircraft routing. Of course, Zabarankin et al. could also apply this method, too.

However, a modest variant of the LRE algorithm provides a simpler method of incorporating turn constraints, one that does not modify the network's topology: In Step 3 of the algorithm, under "Conditions for extending a subpath," we simply add one more condition: If edge $e' = (v_{k-1}, u)$ has just been added to the current path, then edge $e = (u, v)$ can be added to the path only if the angle between e and e' does not exceed θ. This modification is valid because $d(v)$, $d_0(v)$, and $d_i(v)$ for all i are computed while ignoring turn constraints and therefore provide valid lower bounds on turn-constrained versions of Lagrangian distances, true distances, and weights from v to t, respectively. (This algorithmic variant points out a key advantage of the LRE approach to solving a CSPP: The full history of the route under consideration is always available.)

Tighter bounds than those resulting from $d(v)$, $d_0(v)$, and $d_i(v)$, based on explicitly turn-constrained shortest paths, might be useful here, and could be computed with any standard method (e.g., Caldwell 1961, Boroujerdi and Uhlmann 1998). However, tighter bounds are unnecessary to achieve acceptable computational efficiency in our tests, so we do not pursue that topic in this paper.

For testing purposes, we simply imagine that the constant-speed F/A-18 problem on the "F network," with turn-radius constraints added, represents a high-altitude cruise-missile routing problem. As a baseline, we use the problem with a fuel limit of 370. (See Table 3, row six, columns two and three; and see the path denoted by a solid line in Figures 1 and 2.) Figure 3 depicts three different routes using turn angles that are (a) unconstrained (the baseline), (b) limited to at most 60 degrees, and (c) limited to at most 30 degrees. The corresponding probabilities of mission success are 0.9335, 0.9318 and 0.9307, with corresponding solution times of 2.3, 80.4 and 15.2 seconds. Clearly, adding turn constraints can increase solution times, but the reported times should be more than acceptable for many applications. We further note that the standard method for handling turn constraints in this model, that is, using an expanded network, could simply fail to solve. On average, each vertex in the D-H networks has between 120 and 230 incoming edges, which implies that the standard, expanded network would require more than 10^7 vertices. That many edges could present computational difficulties for current computers.

5.3 Route Planning for Unmanned Aerial Vehicles

We next apply the CSP methodology to planning a medium-altitude surveillance mission for a UAV. At the planning stage, and perhaps even during a mission, minimum-risk routes must be determined that are feasible with respect to maneuverability and fuel consumption. We imagine a UAV with capabilities similar to the current Northrop Grumman Hunter MQ-5B, but with better communications capabilities and hence longer range. Cruising speed is 120 kilometers per hour (km/hr), climb and dive rate is 200 meters per minute (m/min), and the aircraft's mission radius, which will be varied, is at least 500 km. (See Jane's 2005 for a description of the MQ-5B's predecessor, the RQ-5A, and see Northrop Grumman 2005 for the manufacturer's datasheet on the MQ-5B.)

The UAV is assigned to provide detailed battle-damage assessment by observing a target in an AO with active enemy radars and SAMs. A 400 km by 200 km mountainous area northeast of Boise, Idaho, serves as the AO; see Figure 4. The UAV will enter the AO at the area's southwest corner at an altitude of 3400 meters, and will attempt to reach the target located in the northeast corner. Target observation will occur at 2400 meters.

We use digital terrain elevation data freely available from the National Geospatial-Intelligence Agency (2004). Elevations are accurate to within ±30 meters at least 90% of the time, and are provided at points on a grid with 30 arc-second (1 km) spacing. Given the UAV's cruising speed and climb and dive rates, it is convenient then to approximate the AO with a three-dimensional grid network with vertices that have a two-kilometer horizontal spacing and 200-meter vertical spacing. (The horizontal spacing yields edge-traversal times of about one minute.) We adopt metric units here because all terrain and aircraft data are specified in such units.

We begin generating a network model of the AO by defining a grid with 201×101 equally spaced vertices in the horizontal plane. This is replicated 16 times, at 200-meter intervals, starting at 400 meters above sea level. The implied maximum altitude of 3400 meters suffices because the UAV plans a stealthy flight that takes advantage of terrain-masking of threat radar, and this is available only at lower altitudes. The nominal, three-dimensional grid has $201 \times 101 \times 16 = 324{,}816$ vertices, but vertices below the terrain need not be modeled, so the actual number becomes 187,284.

With two-kilometer spacing in the horizontal plane, vertical spacing between vertices corresponds to the climb and dive rate of the UAV, so we nominally connect each vertex to each of its nearest neighbors, including diagonal neighbors, but omit connections to vertices directly above and below. (Of course, the climb and dive rate to the diagonal vertices will be somewhat slower than the nominal 200 m/min.) The mission must follow a path that runs generally southwest to northeast, so we omit any edge that does not have a horizontal component in the north, northeast, east, or southeast direction. This results in a network with 2,011,730 edges.

The UAV is subject to two threat types. Four fixed SAM installations present "type-I threats." Two of these each have a radar range with a 150 km radius and 18,000 meter ceiling; they are located at coordinates (151, 149) and (301, 51), with coordinates measured in kilometers in a Cartesian-coordinate system whose origin lies at the southwest corner of the AO. Two short-range SAM installations, each with a 27.8 km range but with the same 18,000 meter ceiling, are located at coordinates (331, 164) and (365, 124). We model each SAM threat as an ellipsoid with circular horizontal cross-section centered at the SAM's location and a vertical half-axis of 18,000 meters. We assume that the airspace with line of sight to the SAM's location within the ellipsoid is subject to the same threat. Airspace within the ellipsoid, but with line of sight blocked by terrain, is not subject to the threat. Similar to the strike-group example, this represents a fairly simple threat model but, again, the flexibility of the CSP methodology makes more detailed models easy to incorporate. For instance, a threat model could easily account for an aircraft's radar cross section(s), which can vary by edge and along a single edge (Leary 1995, Zabarankin et al. 2006).

Hand-held SAMs, mobile anti-aircraft artillery, and small-arms fire constitute the type-II threat. Since specific intelligence is rarely available on low-altitude threats like these, we assume a uniformly low risk from them over the whole AO, but with that risk decreasing exponentially with distance above the terrain.

As in the F/A-18 example, we construct an additive risk measure c_e, for each edge e, based on the probability p_e of being destroyed by a type-I or type-II threat while the UAV is traversing e. Again, we compute p_e as a function of how much of edge e's length is exposed to various threats, and the magnitude of those threats. We assume no communication between potential observers so that the "independence assumption" is reasonable here. Thus, minimizing $\sum_{e \in E_P} c_e = \sum_{e \in E_P} -\log(1-p_e)$, over all s-t paths E_P, is equivalent to maximizing the probability of no hits from type-I and type-II threats over those paths. Again, we define this as the "probability of mission success."

The edge weight $f_e = f_{1e}$ represents the amount of fuel consumed while flying along edge e and relates to the geometric length of e in kilometers, denoted l_e, and whether the edge is level, ascending or descending. Specifically, we let f_e equal $l_e, 2l_e$, or $0.9l_e$ for the three cases, respectively.

Table 4 reports computational results obtained using a 5% relative optimality tolerance. (We increase the optimality tolerance here because the network is significantly larger than in the first example, and a 1% tolerance leads to orders-of-magnitude increases in computing times in a few cases.) The first column of the table specifies the fuel limit and the second column gives the probability of mission success for the best path found. The third column shows actual fuel consumption for each path, and the fourth column gives solution times.

Figure 4 shows horizontal and vertical views of the near-optimal path with a fuel limit of 485.0. Figure 5 shows similar plots for the near-optimal path with a fuel limit of 530.0. The vertical views make it evident that the near-optimal routes do use terrain-masking to avoid being tracked by radars. (Note: The vertical flight path appears jagged only because of the compressed horizontal scale.)

In Figure 5, the UAV initially stays at a high altitude of 3400 m because terrain masks the line of sight to the first SAM located at (151, 149), and because that altitude nearly eliminates type-II threats. At 200 km into the flight, however, a line of sight would be established to the first SAM, and the UAV descends in response. The UAV maintains a low altitude until it exits the SAM's threat region, approximately 100 km from the destination. The UAV avoids the second long-range SAM centered at (301, 51) by exploiting terrain-masking, and it simply circumvents the short-range SAMs.

5.4 UAV Routing: Multiple Side Constraints

We have already shown that our solution methodology can handle a fuel constraint and turn-radius constraints together. However, all examples up to this point have incorporated only a single, standard side constraint (on fuel), and we wish to demonstrate that multiple

Fuel limit (nm)	Prob. of success	Fuel consumed (nm)	Run time (sec.)
482	0.3742	481.5	1.1
485	0.7288	485.0	1.1
490	0.9818	489.6	15.2
500	0.9871	499.2	9.3
510	0.9883	509.1	5.4
520	0.9890	518.7	5.6
530	0.9893	527.9	2.2
540	0.9894	533.1	2.1
550	0.9894	533.1	2.1

Table 4: Constrained minimum-risk routing for a UAV. The optimality tolerance is 5% and solution times ("Run time") are listed in seconds. It is impossible to reach the destination with less than 481.5 units of fuel. Figures 4 and 5 illustrate two of these cases.

side constraints can be incorporated successfully.

Incorporating two or more side constraints may be important for some applications. For instance, a routing problem for a time-critical mission could require both a fuel constraint and a time-to-target constraint (FM 90-36 1997). Accordingly, we now suppose that the UAV mission described above imposes both types of constraints. (Carlyle et al. 2007 solve large models with up to ten side constraints, but we believe it unlikely that more than two side constraints will be necessary in most aircraft-routing problems.) Each edge now has two weights, one representing fuel consumption and the other flight time. We assume a constant ground speed of 120 km/hr and use the horizontal projection of the geometric length l_e km, divided by the constant ground speed, as a surrogate for time.

Table 5 reports computational results for different combinations of fuel and flight-time limits for the UAV. For each near-optimal path found, the table reports the probability of mission success. No solution time exceeds 45 seconds, and 1-12 seconds is typical. Figure 6 shows horizontal and vertical views of the best path found given fuel and time limits of 530.0 and 245.0, respectively. We note that imposing a time constraint of 245.0 reduces the probability of success only slightly, from 0.9893 to 0.9887. As seen by comparing Figures 5 and 6, a time-constrained route must be more direct, and hence it crosses several high-threat regions. However, most of the threat can be avoided through aggressive terrain-masking.

	Probability of success					
Fuel limit	Time limit (time units)					
(nm)	242	245	250	255	260	265
485.0	0.7288	0.7288	0.7288	0.7288	0.7288	0.7288
490.0	0.9547	0.9818	0.9818	0.9818	0.9818	0.9818
500.0	0.9862	0.9871	0.9871	0.9871	0.9871	0.9871
510.0	0.9870	0.9882	0.9882	0.9883	0.9883	0.9883
520.0	0.9879	0.9885	0.9886	0.9890	0.9890	0.9890
530.0	0.9881	0.9887	0.9890	0.9893	0.9893	0.9893
540.0	0.9881	0.9887	0.9890	0.9893	0.9893	0.9893

Table 5: Fuel and time-constrained minimum-risk routing for a UAV. The optimality tolerance is 5%. All run times are less than 45 seconds. It is impossible to reach the destination with less than 481.5 units of fuel or in less than 241.4 time units. Figure 6 illustrates one of these cases.

5.5 UAV Routing: Ingress and Egress

The case studies above demonstrate that our algorithm quickly finds routes *to* a target. But, the CSP methodology extends easily to find a round trip, from origin to a target and back, when the airspace is separated into two disjoint regions, one for the ingress and one for the egress. In fact, this situation requires no change in the algorithm, only modest changes in the network model. Consider the minimum-risk routing problem for the UAV with a single side constraint on fuel consumption as described in Section 5.3. The UAV will enter the AO at the area's southwest corner at an altitude of 3400 meters, observe the target from 2400 meters in the northeast corner, and then return to the southwest corner to exit the AO at 3400 meters.

The airspace controller has assigned the airspace below and above the southwest-northeast diagonal for ingress and egress, respectively. We create a network that is identical to the one used in Section 5.3, except that: The directions of arcs are reversed above the diagonal, edges across the diagonal are omitted, and the final destination vertex t is located one grid space (2 km) north of the origin vertex s. The total number of edges is 1,982,958.

To exercise this round-trip model, we repeat tests analogous to those reported in Table 4, but using the modified network and with doubled fuel limits. We do not report detailed results, but the longest run time is only 33 seconds, and Figure 7 displays the route found given a fuel limit of 1060.

6 Conclusions

This paper has examined the use of a constrained shortest-path (CSP) model and a new solution algorithm for routing various types of military aircraft. The CSP model is highly flexible and can account for terrain avoidance, terrain-masking of enemy radar, aircraft maneuverability constraints, varying aircraft speeds, and any number of ground-based threats such as surface-to-air missiles (SAMs). We have focused on an objective that minimizes an additive risk function, which is equivalent to minimizing the probability that the aircraft, or one aircraft in a group of aircraft, will be detected and shot down. Routes can be limited by any reasonable number of constraints on such factors as fuel consumption and flight time, although one of those factors could be moved into the objective and a limit on risk incorporated as a constraint.

Our basic CSP solution algorithm, the "LRE algorithm,"combines Lagrangian relaxation with enumeration of near-shortest paths. However, enhancements of the basic algorithm yield substantial computational improvements. In particular, "network reductions" identify edges that can be proven not to lie on an optimal path. We apply these reductions in a standard preprocessing mode before the main algorithm begins, but also after the first feasible solution has been identified, and even repeatedly during the enumeration process as that process identifies improving feasible solutions.

The enhanced LRE algorithm solves realistic routing problems—we have investigated the routing of strike aircraft and unmanned aerial vehicles—in 80 seconds or less on a desktop computer. The algorithm extends easily to incorporate turn-radius constraints, which offers a clear advantage over the alternative solution method described in the literature, a label-setting algorithm. We have also demonstrated the solution of a round-trip routing problem that incorporates separate ingress and egress corridors.

The probability that a particular SAM installation detects and then destroys an aircraft may depend on the aircraft's path. For example, early detection by distant radar systems, relayed through a command-and-control system, may increase detection probability and tracking accuracy for that installation. Our basic model cannot handle the resulting path-dependent probabilities. However, assuming that the "true" risk associated with a path can be computed quickly, and the model under independence provides a lower bound on that risk—under normal circumstances it will—our approach may be useful: (*i*) Begin enumerating feasible paths that are near-optimal under independence, (*ii*) evaluate each feasible path for its true risk, always saving the best as the incumbent solution, and (*iii*) halt when the enumeration procedure proves that the lower bound on risk over all unexplored feasible paths reaches or exceeds the incumbent solution's true risk. We can implement (*iii*) by modifying conditions within the path-enumeration procedure.

Future work will study path-dependent probabilities, as just described, specialized bounds to improve solution speeds for turn-constrained problems, and integer cutting planes, added as Lagrangianized side constraints, to tighten bounds and reduce enumeration.

Acknowledgments

The authors thank the Office of Naval Research and the Air Force Office of Scientific Research for funding this research. The authors are also grateful for information on air-mission planning obtained in discussions with Lieutenant Commanders Vic Bindi, Christopher Landon, and Chris McCarthy, and Major Brian Zacherl.

References

Ahuja, R.K., Magnanti, T.L., and Orlin, J.B. 1993. *Network Flows*, Prentice Hall, Englewood Cliffs, New Jersey.

Aneja, Y., Aggarwal, V., and Nair, K. 1983. "Shortest Chain Subject to Side Conditions," *Networks*, Vol. 13, No. 2, 295–302.

Beasley, J. and Christofides, N. 1989. "An Algorithm for the Resource Constrained Shortest Path Problem," *Networks*, Vol. 19, No. 4, 379–394.

Bindi, V. and McCarthy, C. 2004. United States Navy, Private Communication, 29 September.

Boroujerdi, A. and Uhlmann, J. 1998. "An Efficient Algorithm for Computing Least Cost Paths with Turn Constraints," *Information Processing Letters*, Vol. 67, No. 6, 317–321.

Bortoff, S.A. 2000. "Path Planning for UAVs," *Proceedings of the American Control Conference*, 28-30 June, 364–368.

Byers, T.H. and Waterman, M.S. 1984. "Determining Optimal and Near-Optimal Solutions When Solving Shortest Path Problems by Dynamic Programming," *Operations Research*, Vol. 32, No. 6, 1381–1384.

Caldwell, T. 1961. "On Finding Minimal Routes in a Network with Turning Penalties," *Communications of the ACM*, Vol. 4, No. 2, 107–108.

Carlyle, W.M., Royset, J.O., and Wood, R.K. 2007. "Lagrangian Relaxation and Enumeration for Solving Constrained Shortest-Path Problems," in review.

Chen, S., and Nahrstedt, K. 1998. "On Finding Multi-Constrained Paths," *Conference Record. 1998 IEEE International Conference on Communications, Vol. 2*, June 7-11, 874–879.

CLOAR 2007. Common Low Observable Autorouter, BAE Systems, Battle Management Systems Group, San Diego, California, http://www.baesystems.com (accessed 15 March 2007).

Desrochers, M. and Soumis, F. 1988. "A Generalized Permanent Labeling Algorithm for the Shortest Path Problem with Time Windows," *Information Systems and Operational Research*, Vol. 26, No. 3, 191–212.

Day, P.R. and Ryan, D.M., (1997), "Flight Attendant Rostering for Short-Haul Airline Operations," *Operations Research*, Vol. 45, No. 5, pp. 649-661.

DeWolfe, D., Stevens, J. and Wood, K. 1993. "Setting Military Reenlistment Bonuses," *Naval Research Logistics*, Vol. 40, No. 2, 143–160.

Dumitrescu, I. and Boland, N. 2003. "Improved Preprocessing, Labeling and Scaling Algorithm for the Weight-Constrained Shortest Path Problem," *Networks*, Vol. 42, No. 3, 135–153.

FalconView 2007. Georgia Tech Research Institute, Atlanta, Georgia, http://www.falconview.org (accessed 15 March 2007).

Fisher, M.L. 1981. "The Lagrangian Relaxation Method for Solving Integer Programming Problems," *Management Science*, Vol. 27, No. 1, 1–18.

FM 90-36. 1997. "Targeting: The Joint Targeting Process and Procedures for Targeting Time-Critical Targets," Army Training and Doctrine Command, Fort Monroe, VA, July.

Fox, B.L. and Landi, D.M. 1970. "Searching for the Multiplier in One-Constraint Optimization Problems," *Operations Research*, Vol. 18, No. 2, 253–262.

Garey, M.R. and Johnson, D.S. 1979. *Computers and Intractability: A Guide to the Theory of NP-Completeness*, W.H. Freeman and Co., San Francisco, California.

Grignon, L., Prasetio, Y., Toktas, B., Yen, J., and Zabinsky, Z. 2002. "A Basis Space-Time Network Formulation for Aircraft Rerouting with Airspace Closures," Technical Report, Department of Industrial Engineering, University of Washington, Seattle, Washington.

Handler, G. and Zang, I. 1980. "A Dual Algorithm for the Constrained Shortest Path Problem," *Networks*, Vol. 10, No. 4, 293–310.

Helgason, R.V., Kennington, J.L. and Lewis, K.R. 2001. "Cruise Missile Mission Planning: A Heuristic Algorithm for Automatic Path Generation," *Journal of Heuristics*, Vol. 7, No. 5, 473–494.

Hebert, J.M. 2001. "Air Vehicle Path Planning," PhD dissertation, Graduate School of Engineering and Management, Air Force Institute of Technology, Wright-Patterson AFB, Ohio, November.

Inanc, T., Misovec, K., and Murray, R. M. 2004. "Nonlinear Trajectory Generation for Unmanned Air Vehicles with Multiple Radars," *Proceedings of the 43rd IEEE Conference on Decision and Control (CDC)*, 14-17 December, 3817–3822.

Jane's 2005, "Northrop Grumman/IAI RQ-5A Hunter," *Jane's Unmanned Aerial Vehicles and Targets*, Jane's Information Group, London, 28 November.

JMPS 2007. Joint Mission Planning System Program, http://www.globalsecurity.org/military/systems/aircraft/systems/jmps.htm (accessed 15 March 2007).

Joksch, H. 1966. "The Shortest Route Problem with Constraints," *Journal of Mathematical Analysis and Application,* Vol. 14, No. 2, 191–197.

Juttner, A., Szviatovszki, B., Mecs, I., and Rajko, Z. 2001. "Lagrangian Relaxation Based Method for the QoS Routing Problem," *Proceedings of the IEEE INFOCOM 2001 Conference, Vol. 2*, 22-26 April, 859–868.

Kim, J. and Hespanha, J. 2003. "Discrete Approximations to Continuous Shortest-Path: Application to Minimum-Risk Path Planning for Groups of UAVs," *42nd IEEE Conference on Decision and Control*, 9-12 December.

Korkmaz, T. and Krunz, M. 2001. "Multi-Constrained Optimal Path Selection," in *Proceedings of the IEEE INFOCOM 2001 Conference, Vol. 2*, 22-26 April, 834–843.

Kuipers, F. Korkmaz, T. Krunz, M. and Van Mieghem, P. 2004. "Performance Evaluation of Constraint-Based Path Selection Algorithms," *IEEE Network*, Vol. 18, No. 5, 16–23.

Landon, C. 2004. United States Navy, Private Communication, 4 October.

Leary, J. 1995. "Search for a Stealthy Flight Path Through a Hostile Radar Defense Network," Master's Thesis, Naval Postgraduate School, Monterey, California, March.

Lee, S.H.K. 1995. "Route Optimization Model for Strike Aircraft," Master's Thesis, Naval Postgraduate School, Monterey, California, September.

Lewis, D.H. 1988. "Optimal Three-Dimensional Path Planning Using Visibility Constraints," Master's Thesis, Naval Postgraduate School, Monterey, California, December.

Lorenz, D.H., and Raz, D. 2001. "A Simple Efficient Approximation Scheme for the Restricted Shortest Path Problem," *Operations Research Letters*, Vol. 28, No. 5, 213–219.

Marcum, J.I. 1947. "A Statistical Theory of Target Detection by Pulsed Radar," The Rand Corporation, Research Memorandum RM-754, 1 December.

Murphey, R., Uryasev, S., and Zabarankin, M. 2003a. "Optimal Path Planning in a Threat Environment" in *Recent Developments in Cooperative Control and Optimization*, P. Pardalos, ed., Kluwer Academic Publishers, Dordrecht, 349–406.

Murphey, R., Uryasev, S., and Zabarankin, M. 2003b. “Trajectory Optimization in a Threat Environment,” Research Report 2003-9, Dept. of Industrial & Systems Engineering, University of Florida, Gainesville, FL.

Nachtigal, K. 1995. “Time Depending Shortest-path Problems with Applications To Railway Networks,” *European Journal of Operational Research*, Vol. 83, No. 1, 154–166.

National Geospatial-Intelligence Agency, http://geoengine.nga.mil/ (accessed 15 October 2004).

Northrop Grumman 2005, “MQ-5B Hunter,” TDEA-7968, Northrop Grumman Corporation, San Diego, California, 8 September.

Novy, M.C. 2001. “Air Vehicle Optimal Trajectories for Minimization of Radar Exposure,” PhD dissertation, Graduate School of Engineering and Management, Air Force Institute of Technology, Wright-Patterson AFB, Ohio, March.

OPUS 2007. OR Concepts Applied, Whittier, California, http://www.orconceptsapplied.com (accessed 15 March 2007).

Pfeiffer, B., Batta, R., Klamroth, K., and Nagi, R. 2005. “Path Planning for UAVs in the Presence of Threat Zones Using Probabilistic Modeling,” http://www.am.uni-erlangen.de/ pfeiffer/documents/prob120705.pdf (accessed 12 April 2007)

Polymenakos, L.C., Bertsekas, D.P., and Tsitsiklis, J.N. 1998. “Implementation of Efficient Algorithms for Globally Optimal Trajectories,” *IEEE Transactions on Automatic Control*,” Vol. 43, No. 2, 278–283.

Russell, S. and Norvig, P. 1995. *Artificial Intelligence: A Modern Approach*, Prentice Hall, Upper Saddle River, New Jersey.

SAIC Mission Planning System 2007. Science Applications International Corp., San Diego, California, http://www.saic.com/products/aviation/saicmps (accessed 15 March 2007).

Shorack, G.R., 1964. “Algorithms and Analog Computers for the Most Reliable Route through a Network,” *Operations Research*, Vol. 12, No. 4, 632–633.

Tharp, J. 2003. “JRAPS for Mission Planning,” Northrop Grumman Information Technology, TASC, Sterling, Virginia.

Tsitsiklis, J.N. 1995. “Efficient Algorithms for Globally Optimal Trajectories,” *IEEE Transactions on Automatic Control*, Vol. 40, No. 9, 1528–1538.

Vance, P.H., Barnhart, C., Johnson, E.L. and Nemhauser, G.L. 1997. “Airline Crew Scheduling: A New Formulation and Decomposition Algorithm,” *Operations Research*, Vol. 45, No. 2, 188–200.

Van Mieghem, P., De Neve, H., and Kuipers, F.A. 2001. “Hop-by-hop Quality of Service Routing,” *Computer Networks*, Vol. 37, No. 3-4, 407–423.

Vian, J.L. and Moore, J.R. 1989. “Trajectory Optimization with Risk Minimization for Military Aircraft,” *Journal of Guidance, Control, and Dynamics*, Vol. 12, No. 3, 311–317.

Wolsey, L.A. 1998. *Integer Programming*, John Wiley & Sons, New York, New York.

Zabarankin, M., Uryasev, S., and Pardalos, P. 2002. “Optimal Risk Path Algorithms” in *Cooperative Control and Optimization*, R. Murphey and P. Pardalos, eds., Kluwer Academic Publishers, Dordrecht, Netherlands, 273–299.

Zabarankin, M., Uryasev, S., and Murphey, R. 2006. “Aircraft Routing under the Risk of Detection,” *Naval Research Logistics*, Vol. 53, No. 8, 728–747.

Zacherl, B.J. 2006. United States Marine Corps, Private Communication, 17 January.

Ziegelmann, M. 2001. “Constrained Shortest Paths and Related Problems,” PhD Dissertation, Universität des Saarlandes, Saarbrücken, Germany.

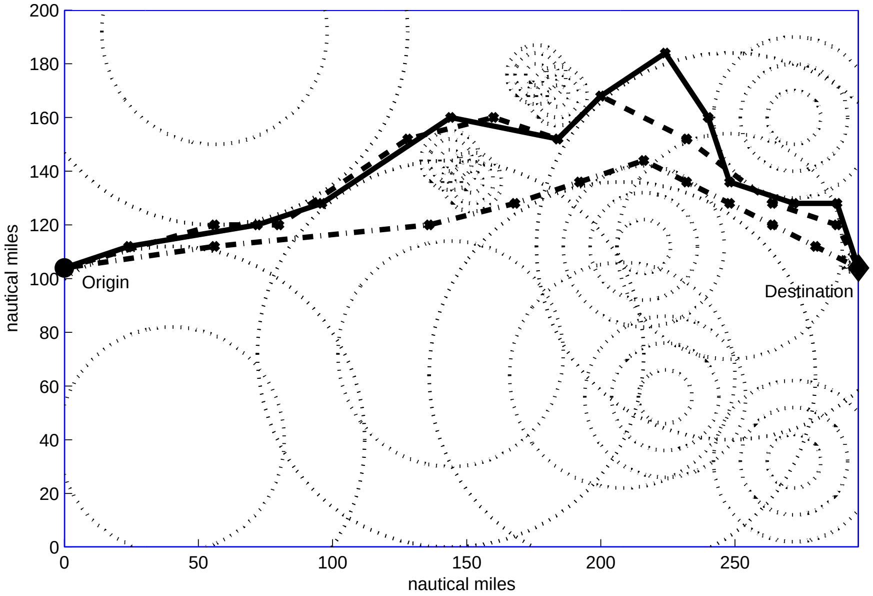

Figure 1: Minimum-risk routes for an F-A/18 strike group subject to various fuel limits. Graph structure F is used. Concentric circles represent different levels of risk surrounding a central SAM site. Probabilities of mission success are 0.7261, 0.9287, and 0.9335, for fuel limits of 310 (·−), 340 (−−), and 370 (solid line), respectively. (See Table 3, column two.)

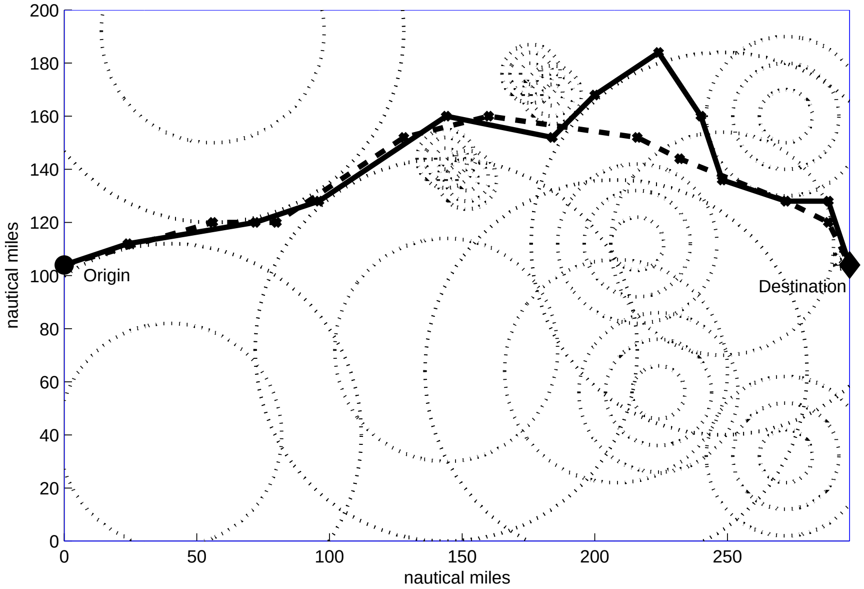

Figure 2: Minimum-risk routes for an F-A/18 strike group with fuel limit 370, flying at constant speed (solid line) or variable speed (dashed line). Graph structure F is used. The probabilities of mission success are 0.9335 and 0.9596, respectively. Constant speed results in a long detour to exploit a marginally safer approach, while variable speed conserves fuel initially for a high-speed, direct approach. (See Table 3, columns two and four.)

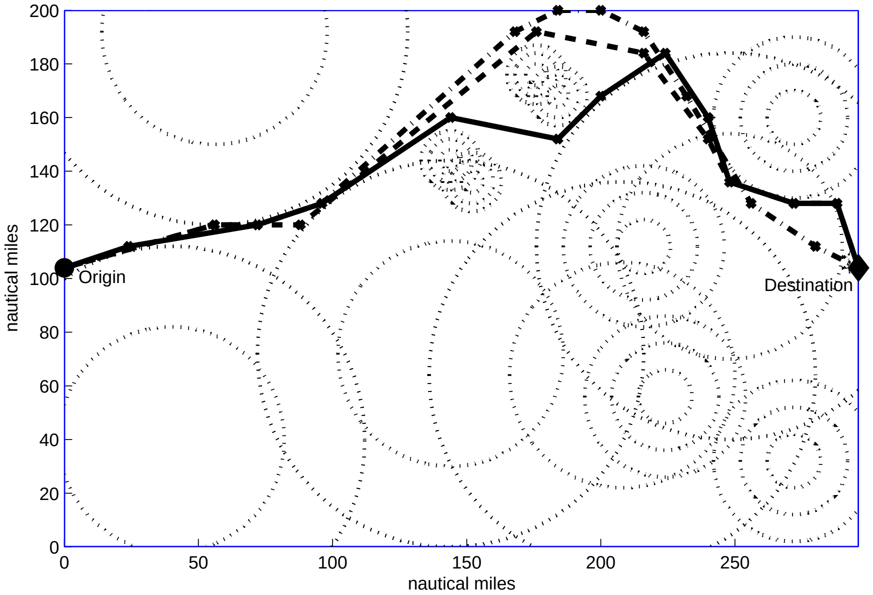

Figure 3: Minimum-risk routes for an F-A/18 strike group subject to a fuel limit of 370 and constraints disallowing turns greater than 30 degrees (·−) and 60 degrees (−−), and allowing all turns (solid line). The respective probabilities of mission success are 0.9311, 0.9320, and 0.9335. All paths are computed using a 1% optimality tolerance.

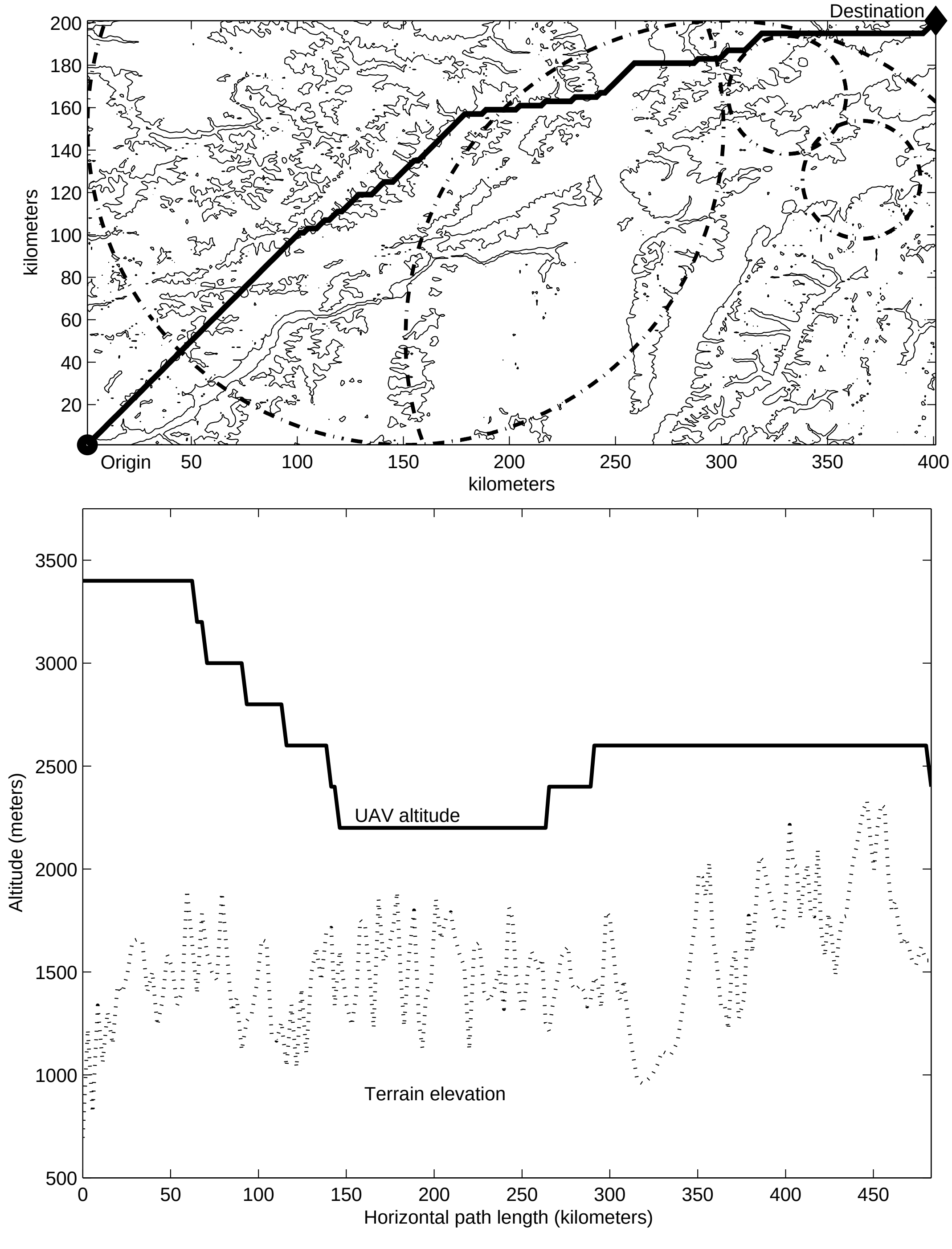

Figure 4: Horizontal (a) and vertical (b) views of a minimum-risk route for a UAV. Contours in the horizontal view lie at 800, 1600 and 2400 meters. Four circles represent area within range of four SAM sites. Blocked line-of-sight eliminates threat. The optimal path uses terrain masking to avoid the SAMs' radars, but tries to stay high to avoid a diffuse ground threat. The fuel limit is 485 and the resulting probability of mission success is 0.7288. (See Table 4.)

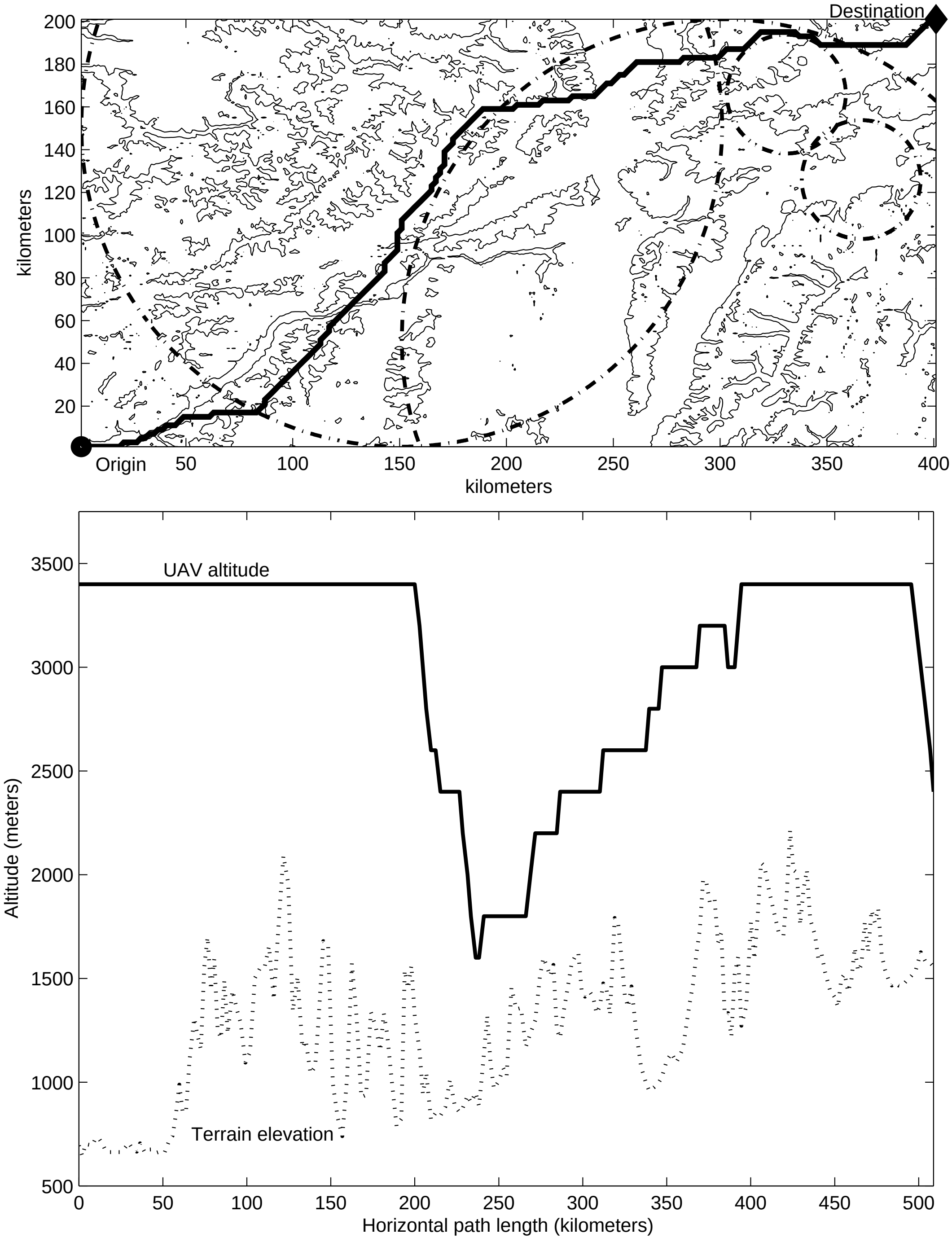

Figure 5: Horizontal (a) and vertical (b) views of a minimum-risk route for a UAV. Contours in the horizontal view lie at 800, 1600 and 2400 meters. The data for this problem are identical to those in Figure 4 except the fuel limit increases to 530. Because of this increase, the UAV can dive and climb more to take advantage of terrain-masking, and the probability of mission success increases to 0.9893. (See Table 4.)

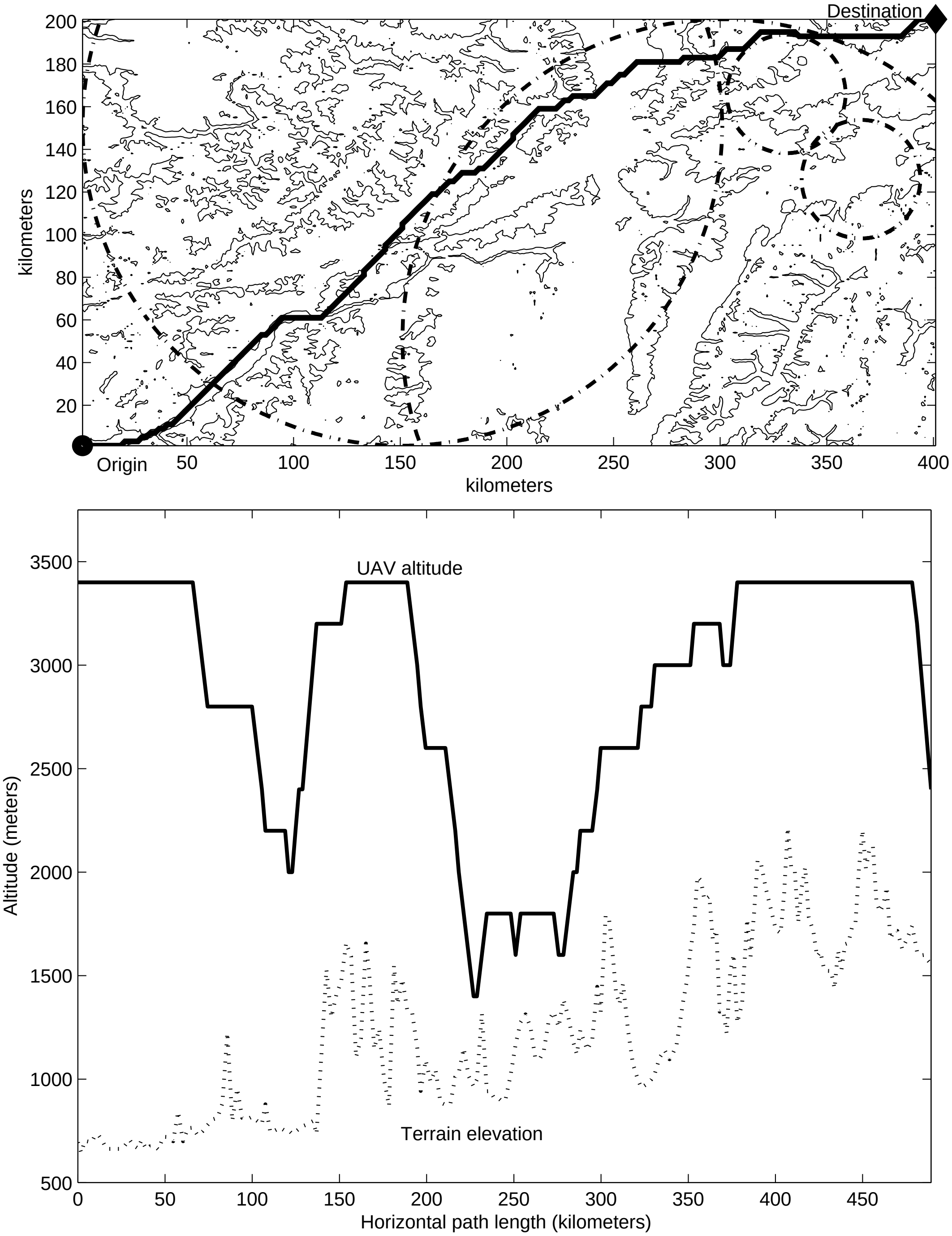

Figure 6: Horizontal (a) and vertical (b) views of a minimum-risk route for a UAV with both a fuel limit (530) and flight-time limit (245). Contours in the horizontal view lie at 800, 1600 and 2400 meters. This figure demonstrates that the applied limits allow a significant amount of terrain-masking. The probability of mission success is 0.9887. (See Table 5.)

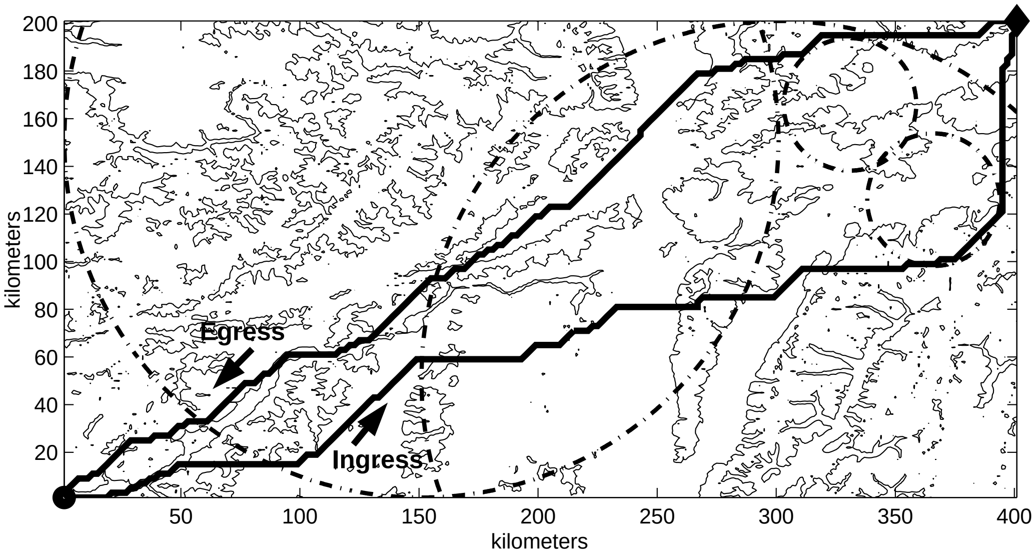

Figure 7: Horizontal view of a minimum-risk ingress and egress route for a UAV with fuel limit 1060. The ingress corridor lies below southwest-northeast diagonal, and the egress corridor lies above. Contours in the horizontal view lie at 800, 1600 and 2400 meters. The probability of mission success is 0.9719.

Unofficial Appendix

Military Operations Research does not print in color, but the reviewer may wish to see the color versions of Figures 4-7.

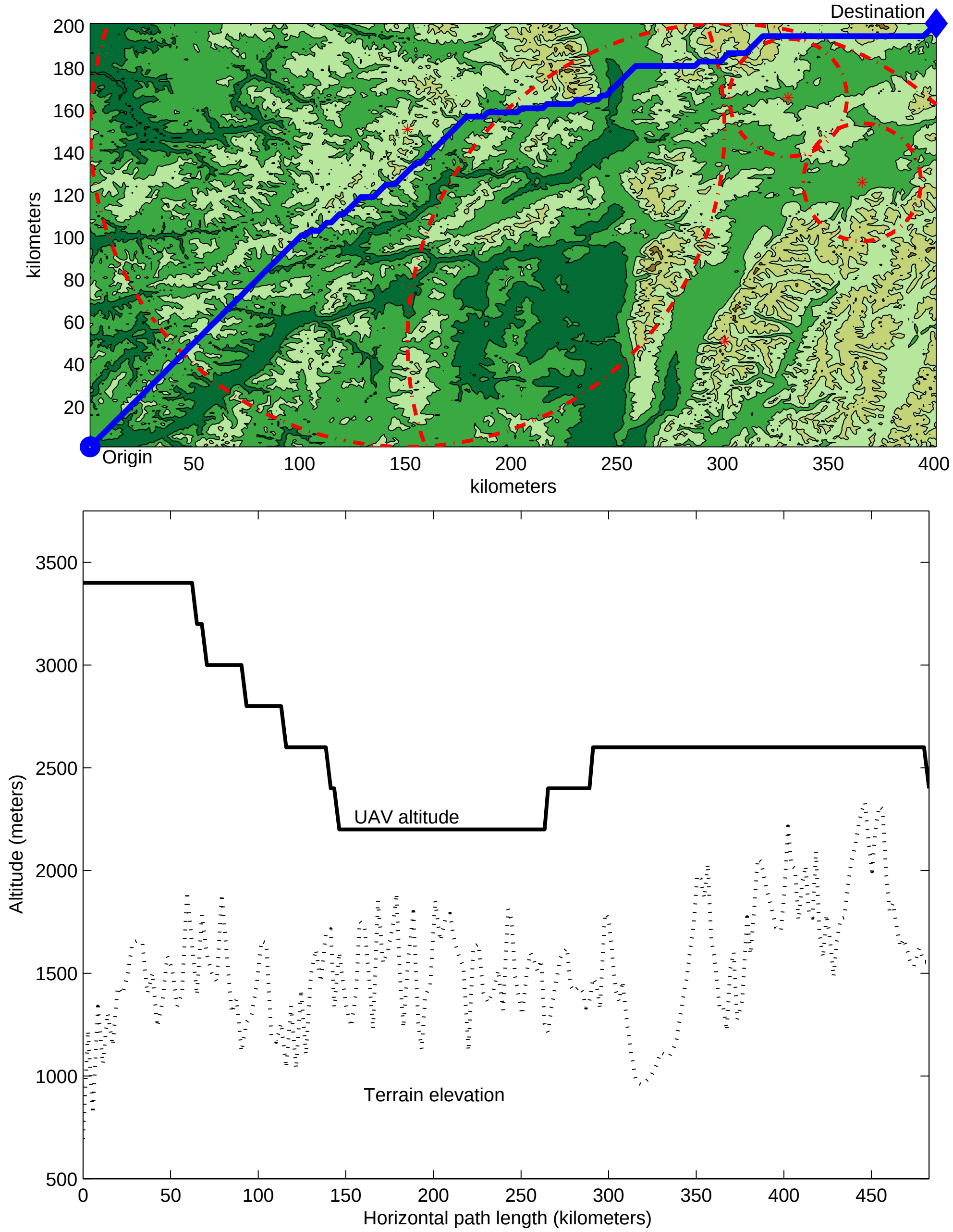

Figure 8: Color version of Figure 4

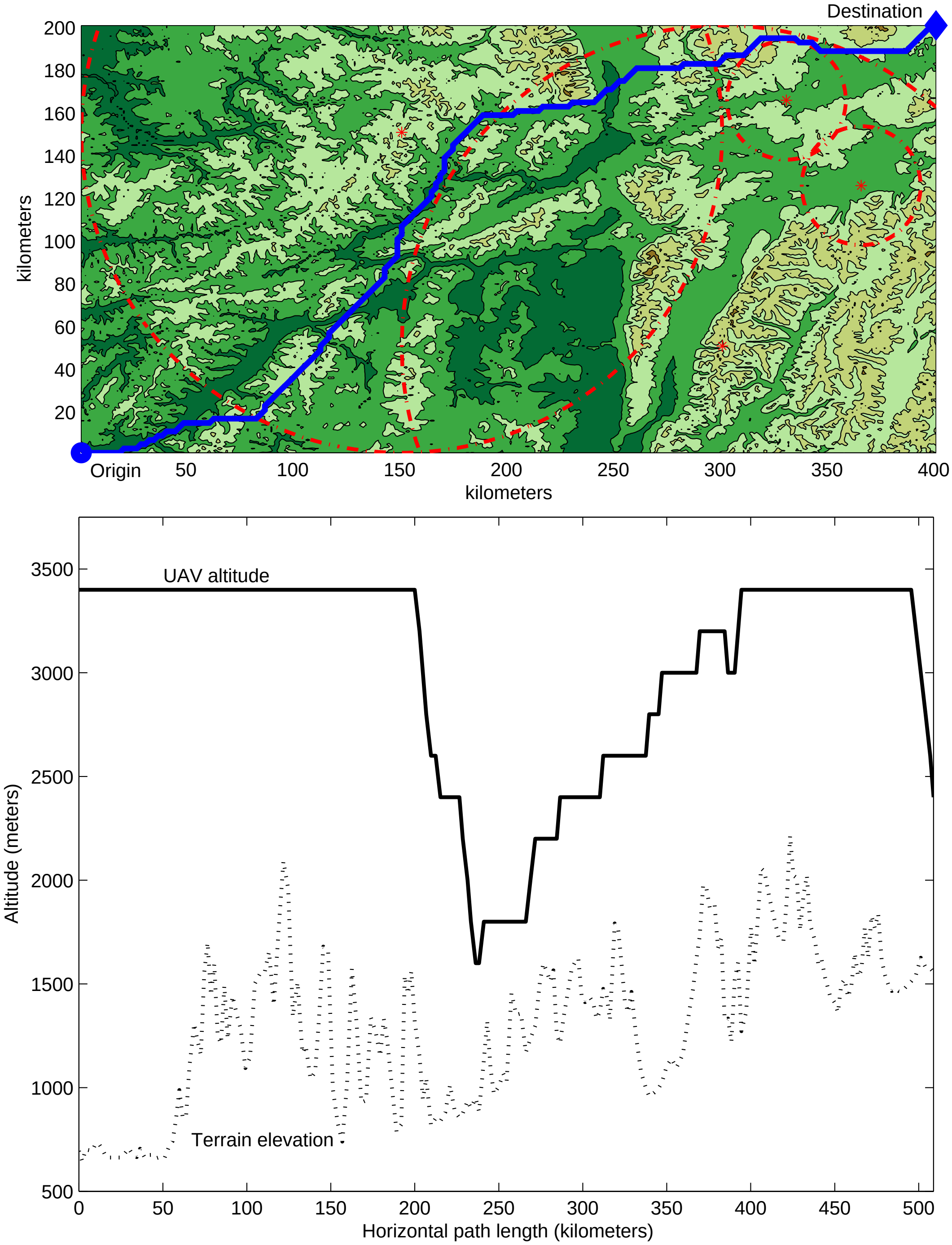

Figure 9: Color version of Figure 5

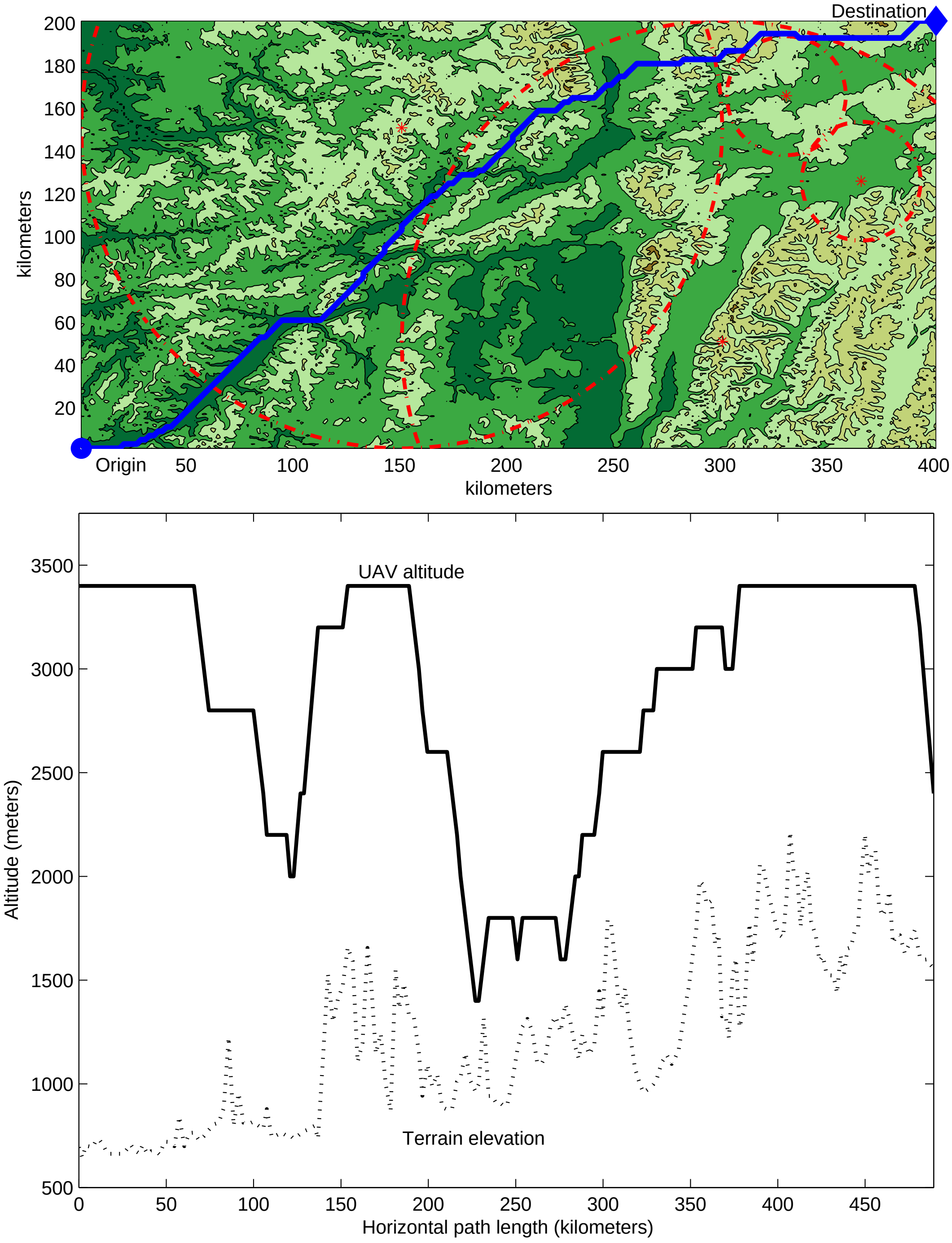

Figure 10: Color version of Figure 6

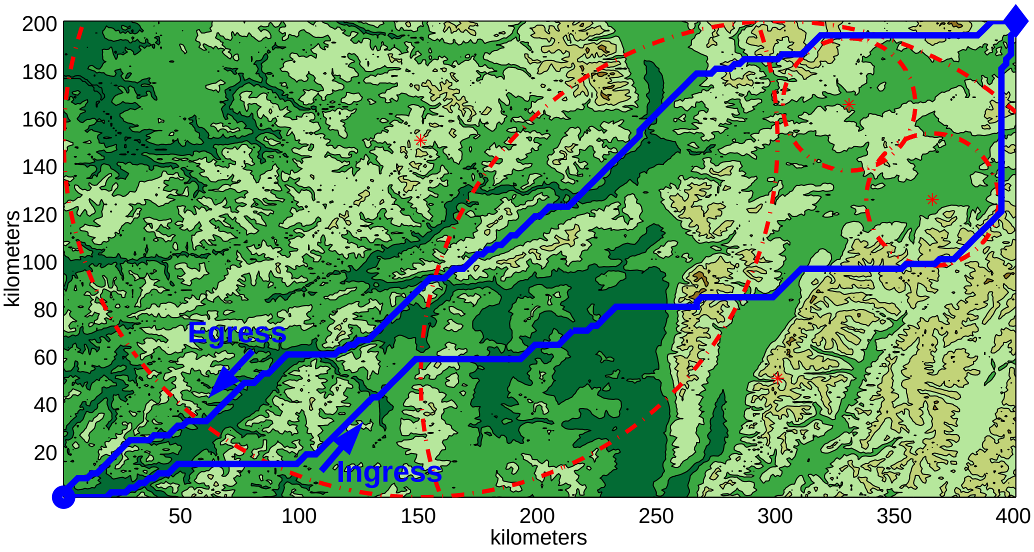

Figure 11: Color version of Figure 7