

# Discrete Approximations to Continuous Shortest-Path: Application to Minimum-Risk Path Planning for Groups of UAVs\*

Jongrae Kim  
Dept. of Electrical & Computer Engineering  
University of California  
Santa Barbara, CA 93106, USA  
jongrae@ece.ucsb.edu

João P. Hespanha  
Dept. of Electrical & Computer Engineering  
University of California  
Santa Barbara, CA 93106, USA  
hespanha@ece.ucsb.edu

*Abstract*—This paper addresses the weighted anisotropic shortest-path problem on a continuous domain, i.e., the computation of a path between two points that minimizes the line integral of a cost-weighting function along the path. The cost-weighting depends both on the instantaneous position and direction of motion. We propose an algorithm for the computation of shortest-path that reduces the problem to an optimization over a finite graph. This algorithm restricts the search to paths formed by the concatenation of straight-line segments between points, from a suitably chosen discretization of the continuous region.

To maximize efficiency, the discretization of the continuous region should not be uniform. We propose a novel “honeycomb” sampling algorithm that minimizes the cost penalty introduced by discretization. The resulting path is not optimal but the cost penalty can be made arbitrarily small at the expense of increased computation.

This methodology is applied to the computation of paths for groups of Unmanned Air Vehicles (UAVs) that minimize the risk of being destroyed by ground defenses. We show that this problem can be formulated as a weighted anisotropic shortest-path optimization and show that the algorithm proposed can efficiently produce low-risk paths.

## I. INTRODUCTION

Consider a compact region  $\mathcal{R} \subset \mathbb{R}^n$  and two points  $x_i, x_f \in \mathcal{R}$ . Our goal is to compute a continuous path  $\rho$  from  $x_i$  to  $x_f$  that minimizes the line integral over  $\rho$  of a cost-weighting function  $\ell$  that depends on position and direction of motion. Motivated by the observation that the optimal path is a straight-line from  $x_i$  to  $x_f$  when the cost-weighting is constant, one often refers to this problem as a *shortest-path* optimization. To emphasize the fact that the cost-weighting is not uniform and that it depends on the direction of motion, we further qualify it as a *weighted anisotropic shortest-path* optimization.

To formalize the problem, we denote by  $\mathcal{P}$  the set of all unit-speed paths in  $\mathcal{R}$  from  $x_i$  to  $x_f$  that are continuous and piecewise twice continuously differentiable, i.e., the set of continuous functions  $\rho : [0, T] \rightarrow \mathcal{R}$ ,  $T > 0$  for which (i)  $\rho(0) = x_i$  and  $\rho(T) = x_f$ ; (ii)  $\dot{\rho}$  and  $\ddot{\rho}$  exist on  $[0, T]$ , except for a finite number of points; and (iii)  $\|\dot{\rho}\| = 1$  wherever this derivative exists. The problem under consideration can be formalized as follows:

\*This paper is based upon work supported by DARPA under the Space and Naval Warfare Systems Center, San Diego, Contract Number N66001-01-C-8076. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of DARPA or the Space and Naval Warfare Systems Center, San Diego.

*Problem 1 (Weighted anisotropic shortest-path):*  
Compute a path  $\rho^* \in \mathcal{P}$  such that<sup>1</sup>

$$J[\rho^*] = \min_{\rho \in \mathcal{P}} J[\rho],$$

where  $J : \mathcal{P} \rightarrow [0, \infty)$  denotes the functional defined by

$$J[\rho] := \int_0^T \ell(\rho(t), \dot{\rho}(t)) dt,$$

for each  $\rho : [0, T] \rightarrow \mathcal{R}$  in  $\mathcal{P}$ . □

We assume throughout the paper that the cost-weighting  $\ell : \mathcal{R} \times \mathbb{R}^n \rightarrow [0, \infty)$  is continuously differentiable.

The solution to this problem has numerous applications that range from mobile robotics to path planning on topographical maps. The specific application pursued here is the computation of paths for groups of Unmanned Air Vehicles (UAVs) that minimize the risk of being destroyed by ground defenses, e.g., Surface-to-Air Missiles (SAMs).

The computation of shortest-paths has a long history and is, in fact, the most basic problem in Calculus of Variations (cf., e.g., [7]). Assuming for simplicity that  $\mathcal{R} := \mathbb{R}^n$ , the optimization formulated above is equivalent to the optimal control problem of finding a terminal time  $T \geq 0$  and a control  $v : [0, T] \rightarrow \mathcal{V}$ , with  $\mathcal{V} := \{z \in \mathbb{R}^n : \|v\| = 1\}$  that minimizes the cost  $J := \int_0^T \ell(x(t), v(t)) dt$ , subject to the dynamics  $\dot{x} = v$  and initial and terminal conditions  $x(0) = x_i$ ,  $x(T) = x_f$ . The solution to this problem can be found using the Hamilton-Jacobi-Bellman (HJB) equation: Assuming that there exists a continuously differentiable solution  $V : \mathbb{R}^n \rightarrow \mathbb{R}$  to the HJB equation

$$0 = \min_{v \in \mathcal{V}} H(x, v, \nabla_x V(x)), \quad \forall x \in \mathbb{R}^n,$$

with boundary condition  $V(x_f) = 0$ , where the Hamiltonian  $H$  is defined by  $H(x, v, p) := \ell(x, v) + \langle p, v \rangle$ ,  $\forall x, v, p \in \mathbb{R}^n$ , the optimal control  $v^* : [0, T^*] \rightarrow \mathcal{V}$  is given by

$$v^*(t) = \arg \min_{v \in \mathcal{V}} H(x^*(t), v, \nabla_x V(x^*(t))),$$

where  $x^*$  denotes the optimal trajectory defined by  $\dot{x}^* = v^*$  and  $x^*(0) = x_i$ . However, this method generally fails when the HJB equation (or a relaxation of it) has no continuously differentiable solution. Even when an appropriate solution exists (perhaps only a viscosity solution), it is generally computationally very difficult to find.

<sup>1</sup>Without further assumptions, the minimum may not be achieved for a path in  $\mathcal{P}$ , but we will ignore this for now.

Several methods have been proposed to overcome this difficulty. These methods typically explore special structures for the cost-weighting  $\ell$  and/or compute paths that are only approximately optimal. We pursue here the latter approach and solve a discrete version of the continuous problem, which provides an approximate solution. This approximation yields solutions that are not necessarily optimal but whose cost can be made arbitrarily close to the optimal one. We start by sampling the region  $\mathcal{R}$  to extract a finite number of points  $\mathcal{X}$  and then restricting the search to paths consisting of the concatenation of line segments between points in  $\mathcal{X}$ . The search portion of the algorithms is performed by constructing a finite graph (whose nodes are the points in  $\mathcal{X}$ ), for which one then solves a shortest-path problem using standard algorithms.

This type of approach has been proposed before (cf., Section II) but because of the “curse of dimensionality” its successful application to nontrivial problems depends crucially on the algorithm used to sample the region  $\mathcal{R}$ . This paper addresses precisely this issue. We start by deriving in Section III a worst-case bound for the cost penalty introduced by discretization. Inspired by this bound, we then propose in Section IV an efficient algorithm to sample  $\mathcal{R}$  so as to achieve a small cost penalty with a relatively sparse sampling of  $\mathcal{R}$ . The key idea is to sample  $\mathcal{R}$  so that there will be more sample points in regions where the optimal path is more likely to deviate from a straight-line.

We apply the proposed algorithm to the computation of paths for groups of UAVs, which minimize their probability of destruction by ground defenses. This is an inherently three-dimensional, anisotropic shortest-path problem for which previous algorithms do not apply. The results obtained, which are summarized in Section V, validate the algorithm and show that it can solve the problem with reasonable computational effort. In this section we also compare the performance of several alternative sampling methods.

## II. RELATED WORK

The solution to shortest-path problems is an active area of research in computational geometry [18]. When strong assumptions on the cost-weighting  $\ell$  are imposed, efficient algorithms can be used to compute the shortest-path. Often  $\ell(x, v)$  is assumed independent of the velocity  $v$  and takes only two distinct values: a low value corresponding to “free-space” and a high value (in the limit  $+\infty$ ) corresponding to “obstacles.” Hershberger and Suri [8] proposed an algorithm for planar regions and obstacle spaces defined by polygons that runs in worst-case time  $O(n \log n)$  and requires  $O(n \log n)$  space, where  $n$  is the total number of obstacle vertices. Another algorithm that is competitive when the number  $m$  of disconnected obstacles is much smaller than the number of vertices has been proposed by Kapoor et al. [9] and runs in worst-case time  $O(n + m^2 \log n)$  and requires  $O(n)$  in space. The complexity of the problem increases when the cost-weighting  $\ell(x, v)$  is still independent of the velocity  $v$  and piecewise constant over space, but can take more than two values. However, polynomial-time algorithms can still be found that produce  $\epsilon$ -optimal solutions, i.e. paths

for which the cost is above the minimum by a factor no larger than  $(1 + \epsilon)$ . For example, Mitchell and Papadimitriou [14], Aleksandrov et al. [1], and Mata and Mitchell [12] provide  $\epsilon$ -optimal planar algorithms for any  $\epsilon > 0$  that run in worst-case times,  $O(n^8 \log \frac{n}{\epsilon})$ ,  $O(\frac{n}{\epsilon} \log \frac{1}{\epsilon} (\frac{1}{\sqrt{\epsilon}} + \log n))$ , and  $O(\frac{n^3}{\epsilon})$ , respectively. In three dimensions the complexity of the problem also increases but polynomial-time algorithms can still be found that produce  $\epsilon$ -optimal solutions. Aleksandrov et al. [1] provide an  $\epsilon$ -optimal algorithm for any  $\epsilon > 0$  that runs in worst-case time  $O(\frac{n}{\epsilon} \log \frac{1}{\epsilon} (\frac{1}{\sqrt{\epsilon}} + \log n))$ , where  $n$  is the number of convex regions needed to define the piecewise constant cost-weight  $\ell$ . The worst-case time decreases to  $O(\frac{n}{\epsilon^3} \log \frac{1}{\epsilon} \log n)$  when  $\ell$  only takes two very distinct values. The reader is referred to the survey by Mitchell [13] for additional results.

Much less work has been devoted to the anisotropic case, i.e., when  $\ell(x, v)$  depends both on the position  $x$  and velocity  $v$ . Rowe and Ross [17] considered the computation of the minimum-energy path for a ground vehicle moving between two points on a hilly terrain described by a triangular mesh (2 1/2-dimensional case). Rowe and Ross [17] used a simple model for the energy-cost that takes into account the grade of the climb and is therefore velocity dependent. They provided an exact algorithm with worst-time run complexity  $O(n^n)$ . Lanthier et al. [11] later provided an approximate algorithm with polynomial worst-case time.

The numerical solution to stochastic shortest-path problems has also been addressed in the control literature, where most of the focus has been on approximations to the HJB formulation. The reader is referred to [10] for a detailed discussion on this topic. Tsitsiklis [20] proposes two efficient algorithms to solve an isotropic shortest-path problem by solving a discretized HJB equation. By restricting the attention to isotropic cost-weights and sampling  $\mathcal{R}$  using a regular rectangular grid, the author constructs an algorithm whose complexity is linear on the number of sample points. An even more efficient algorithm for the same optimization is proposed in [5]. However, none of these algorithms seems to generalize to the anisotropic case [20].

A fundamentally different approximation to the HJB formulation was proposed in [19] for the isotropic case when  $\ell(x, v)$  only takes two distinct values:  $+\infty$  on a finite collection of non-intersecting “obstacles” and a fixed constant value elsewhere. The authors compute the solution to the HJB exactly for a single disk-like obstacle and then an approximation of it for the case of multiples obstacles, which is constructed considering the effect of one obstacle at a time. The method can be efficiently implemented and the paths generated are near-optimal and guaranteed to reach the goal. This type of approximation depends critically on the binary nature of  $\ell$ .

## III. DISCRETIZATION

To overcome the difficulties that arise in solving the Weighted Anisotropic Shortest-Path Problem 1 exactly, one may finely grid the region  $\mathcal{R}$  and force the path to consist of the concatenation of several straight segments between points in the grid. For a finite number of grid points, this procedure

converts the original continuous shortest-path problem into a shortest-path problem on a finite graph.

Given a finite set of points  $\mathcal{X}$  that includes  $x_i$  and  $x_f$ , we say that a path  $\rho : [0, T] \rightarrow \mathcal{R}$  in  $\mathcal{P}$  is *piecewise linear with respect to  $\mathcal{X}$*  if there exists a sequence of points

$$\{x_0 = x_i, x_1, x_2, \dots, x_m = x_f\} \subset \mathcal{X}$$

such that

$$\rho(t) = x_{k-1} + \frac{x_k - x_{k-1}}{\|x_k - x_{k-1}\|}(t - t_{k-1}), \quad (1)$$

$\forall t \in [t_{k-1}, t_k]$ ,  $k \in \{1, 2, \dots, m\}$ , where the  $t_k$  are defined recursively by

$$t_0 := 0, \quad t_k = t_{k-1} + \|x_k - x_{k-1}\|, \quad \forall k \in \{1, 2, \dots, N\},$$

and  $T = t_m$ . The set of all such paths is denoted by  $\mathcal{P}_{\mathcal{X}}$ . The following results show that restricting our search to this type of path can provide solutions arbitrarily close to the optimal.

**Theorem 1:** For any  $A, \epsilon > 0$ , there exists a finite set  $\mathcal{X}$  of  $N_\epsilon$  points in  $\mathcal{R}$  such that for every initial and final points  $x_i, x_f \in \mathcal{R}$

$$\inf_{\rho \in \mathcal{P}_{\bar{\mathcal{X}}}} J[\rho] - \epsilon \leq \inf_{\rho \in \mathcal{P}_{\|\ddot{\rho}\| \leq A}} J[\rho] \leq \inf_{\rho \in \mathcal{P}_{\bar{\mathcal{X}}}} J[\rho],$$

where  $\bar{\mathcal{X}} := \mathcal{X} \cup \{x_i, x_f\}$  and  $\mathcal{P}_{\|\ddot{\rho}\| \leq A}$  denotes the set of twice continuously differentiable paths in  $\mathcal{P}$  with second derivative bounded by  $A$ .  $\square$

The proof of Theorem 1 is constructive and will inspire the construction of sampling algorithms that attempt to minimize the worst-case cost penalty  $\epsilon$  introduced by discretization. The following two technical results are needed to prove Theorem 1. The proofs of the following Lemmas are omitted due to space limitations.

**Lemma 1:** Given any positive constants  $A, \epsilon_x, \epsilon_v, \epsilon_\delta$  (without loss of generality assuming that  $\epsilon_x \leq \epsilon_\delta$ ), there exists a finite subset  $\mathcal{X}$  of  $\mathcal{R}$  such that for every twice continuously differentiable path  $\rho : [0, T] \rightarrow \mathcal{R}$  in  $\mathcal{P}$  with second derivative bounded by  $A$ , one can find

1) a sequence of times  $\{\tau_0, \tau_1, \dots, \tau_N\} \subset [0, T]$ , with  $\tau_0 := 0 \leq \tau_{k-1} < \tau_k \leq \tau_N := T$ , and

2) a sequence of points  $\{x_0, x_1, \dots, x_N\} \in \mathcal{X}$

such that

$$\|x_k - x_{k-1}\| \leq \epsilon_\delta, \quad \forall k \in \{1, \dots, N\}, \quad (2)$$

$$\|\rho(t) - x_{k-1}\| \leq \epsilon_\delta, \quad \forall t \in [\tau_{k-1}, \tau_k], \quad k \in \{1, \dots, N\}, \quad (3)$$

$$\|\rho(\tau_k) - x_k\| \leq \epsilon_x, \quad \forall k \in \{0, 1, \dots, N\}, \quad (4)$$

$$\left\| \dot{\rho}(\tau) - \frac{x_k - x_{k-1}}{\|x_k - x_{k-1}\|} \right\| \leq \epsilon_v, \quad \forall \tau \in (\tau_{k-1}, \tau_k), \quad k \in \{1, \dots, N\}, \quad (5)$$

$$\sum_{k=1}^N \|x_k - x_{k-1}\| \leq \frac{1 + \epsilon_v}{2 + \epsilon_v} 2T. \quad (6)$$

Moreover, as  $\epsilon_x$  decreases to zero, the integer  $N$  remains uniformly bounded.  $\square$

**Lemma 2:** Given two paths  $\rho_i : [0, T_i] \rightarrow \mathcal{R}$  in  $\mathcal{P}$  and two intervals  $(t_i, \tau_i) \subset [0, T_i]$ ,  $i \in \{1, 2\}$  on which the corresponding  $\rho_i$  are twice continuously differentiable,

$$\begin{aligned} & \int_{t_2}^{\tau_2} \ell(\rho_2(t), \dot{\rho}_2(t)) dt - \int_{t_1}^{\tau_1} \ell(\rho_1(t), \dot{\rho}_1(t)) dt \\ & \leq |\ell(\rho_2(t_2), \dot{\rho}_2(t_2^+)) - \ell(\rho_1(t_1), \dot{\rho}_1(t_1^+))| \Delta \\ & \quad + |\ell(\rho_2(\tau_2), \dot{\rho}_2(\tau_2^-)) - \ell(\rho_1(\tau_1), \dot{\rho}_1(\tau_1^-))| \Delta + (\tau_2 - t_2 - 2\Delta)h \\ & \quad + (g_{x,1} + g_{x,2} + a_1 g_{v,1} + a_2 g_{v,2}) \Delta^2 \end{aligned} \quad (7)$$

where<sup>2</sup>

$$\Delta := \frac{1}{2} \min \{ \|\rho_1(\tau_1) - \rho_1(t_1)\|, \|\rho_2(\tau_2) - \rho_2(t_2)\| \},$$

$$h := \sup_{s \in (t_2 + \Delta, \tau_2 - \Delta)} \ell(\rho_2(s), \dot{\rho}_2(s))$$

$$a_i := \sup_{s \in (t_i, \tau_i)} \|\ddot{\rho}_i(s)\|, \quad i \in \{1, 2\},$$

$$g_{(\cdot), i} := \sup_{s \in (t_i, \tau_i)} \|\nabla_{(\cdot)} \ell(\rho_i(s), \dot{\rho}_i(s))\|, \quad (\cdot) = x \text{ or } v.$$

The inequality (7) still holds if we replace  $(g_{x,1} + g_{x,2} + a_1 g_{v,1} + a_2 g_{v,2}) \Delta^2$  by  $(g_{x,1} + g_{x,2}) \Delta^2 + 2(g_{v,1} + g_{v,2}) \Delta$ .  $\square$

**Proof of Theorem 1:** 1 Let  $\mathcal{X}$  be the set whose existence is guaranteed by Lemma 1 for parameters  $\epsilon_x, \epsilon_v, \epsilon_\delta$  to be defined shortly, and let  $\rho^* : [0, T^*] \rightarrow \mathcal{R}$  be a path in  $\mathcal{P}$  for which

$$J[\rho^*] \leq \inf_{\rho \in \mathcal{P}} J[\rho] + \delta,$$

where  $\delta$  is a constant that can be made arbitrarily small ( $\delta$  is only needed when the infimum is not a minimum). From Lemma 1, we know that there exists a sequence of times  $\{\tau_0, \tau_1, \dots, \tau_N\} \subset [0, T]$ , with  $\tau_0 := 0 \leq \tau_{k-1} < \tau_k \leq \tau_N := T$ , and a sequence of points  $\{x_0, x_1, \dots, x_N\} \in \mathcal{X}$  such that  $\dot{\rho}^*$  and  $\ddot{\rho}^*$  exist on any interval  $(\tau_{k-1}, \tau_k)$ ,  $k \in \{1, \dots, N\}$  and (2)–(6) hold for the path  $\rho^*$ . We can then use these points to construct a path  $\rho : [0, T] \rightarrow \mathcal{R}$  in  $\mathcal{P}_{\bar{\mathcal{X}}}$  defined as in (1). We proceed to compare the costs associated with  $\rho^*$  and  $\rho$ . To this effect, we expand

$$\begin{aligned} J[\rho] - J[\rho^*] &= \sum_{k=1}^N \int_{t_{k-1}}^{t_k} \ell(\rho(s), \dot{\rho}(s)) ds \\ & \quad - \int_{\tau_{k-1}}^{\tau_k} \ell(\rho^*(s), \dot{\rho}^*(s)) ds. \end{aligned}$$

Applying Lemma 2 to each pair of intervals  $(\tau_{k-1}, \tau_k)$  and  $(t_{k-1}, t_k)$ ,  $\forall k \in \{1, \dots, N\}$ , we conclude that

$$\begin{aligned} J[\rho] - J[\rho^*] &\leq \\ & \sum_{k=1}^N |\ell(x_{k-1}, \dot{\rho}(t_{k-1}^+)) - \ell(\rho^*(\tau_{k-1}), \dot{\rho}^*(\tau_{k-1}^+))| \Delta_k \\ & \quad + |\ell(x_k, \dot{\rho}(t_k^-)) - \ell(\rho^*(\tau_k), \dot{\rho}^*(\tau_k^-))| \Delta_k \\ & \quad + (t_k - t_{k-1} - 2\Delta_k) h_k + 2(g_{x,k} + A g_{v,k}) \Delta_k^2, \end{aligned} \quad (8)$$

<sup>2</sup>We denote by  $\nabla_x \ell$  and  $\nabla_v \ell$  the gradient of  $\ell(x, v)$  with respect to  $x$  and  $v$ , respectively.

where

$$\Delta_k := \frac{1}{2} \min \{ \|x_k - x_{k-1}\|, \|\rho^*(\tau_k) - \rho^*(\tau_{k-1})\| \}, \quad (9)$$

$$h_k := \sup_{\substack{x \in \mathcal{R}, v \in \mathcal{V} \\ \|x - x_{k-1}\| \leq \epsilon_\delta}} \ell(x, v), \quad (10)$$

$$g_{x,k} := \sup_{\substack{x \in \mathcal{R}, v \in \mathcal{V} \\ \|x - x_{k-1}\| \leq \epsilon_\delta}} \|\nabla_x \ell(x, v)\|, \quad (11)$$

$$g_{v,k} := \sup_{\substack{x \in \mathcal{R}, v \in \mathcal{V} \\ \|x - x_{k-1}\| \leq \epsilon_\delta}} \|\nabla_v \ell(x, v)\|. \quad (12)$$

From (6), we further conclude that

$$\sum_{k=1}^N \Delta_k \leq \frac{1}{2} \sum_{k=1}^N \|x_k - x_{k-1}\| \leq \frac{1 + \epsilon_v}{2 + \epsilon_v} T. \quad (13)$$

In case  $\|x_k - x_{k-1}\| \leq \|\rho^*(\tau_k) - \rho^*(\tau_{k-1})\|$  we have that  $2\Delta_k = \|x_k - x_{k-1}\| = t_k - t_{k-1}$  and therefore  $t_k - t_{k-1} - 2\Delta_k = 0$ . On the other hand, if  $\|x_k - x_{k-1}\| > \|\rho^*(\tau_k) - \rho^*(\tau_{k-1})\|$ , then  $2\Delta_k = \|\rho^*(\tau_k) - \rho^*(\tau_{k-1})\|$  and therefore

$$\begin{aligned} t_k - t_{k-1} &= \|x_k - x_{k-1}\| \leq \|x_k - \rho^*(\tau_k)\| \\ &\quad + \|\rho^*(\tau_k) - \rho^*(\tau_{k-1})\| + \|x_{k-1} - \rho^*(\tau_{k-1})\| \\ &\leq 2(\epsilon_x + \Delta_k). \end{aligned}$$

In either case it is true that

$$t_k - t_{k-1} - 2\Delta_k \leq 2\epsilon_x. \quad (14)$$

Using the Mean Value Theorem, (4), and (5) we also conclude that

$$\begin{aligned} &|\ell(x_{k-1}, \dot{\rho}(\tau_{k-1}^+)) - \ell(\rho^*(\tau_{k-1}), \dot{\rho}^*(\tau_{k-1}^+))| \\ &\leq g_{x,k}\epsilon_x + g_{v,k}\epsilon_v, \end{aligned} \quad (15)$$

$$\begin{aligned} &|\ell(x_k, \dot{\rho}(\tau_k^-)) - \ell(\rho^*(\tau_k), \dot{\rho}^*(\tau_k^-))| \\ &\leq g_{x,k}\epsilon_x + g_{v,k}\epsilon_v. \end{aligned} \quad (16)$$

We can now use the bounds provided by (13), (14), and (15)–(16) in (8) to obtain

$$J[\rho] - J[\rho^*] \leq \frac{2 + 2\epsilon_v}{2 + \epsilon_v} g T + 2N\epsilon_x h, \quad (17)$$

where  $g := \sup_k (g_{x,k}\epsilon_x + g_{v,k}\epsilon_v + (g_{x,k} + Ag_{v,k})\epsilon_\delta)$  and  $h := \sup_k h_k$ . This finishes the proof, since the right-hand-side of (17) can be made arbitrarily small by selecting  $\epsilon_x$ ,  $\epsilon_\delta$ , and  $\epsilon_v$  sufficiently small. Note that Lemma 2 guarantees that  $N$  will not grow unboundedly as  $\epsilon_x$  is decreased. ■

#### IV. NON-UNIFORM SAMPLING

In principle solving the optimization over paths in  $\mathcal{P}_{\bar{x}}$  is a simple problem that can be solved using standard tools to determine shortest-paths on finite graphs. The main difficulty with this approach is that the required number of points  $N_\epsilon$  may grow very fast as  $\epsilon$  decreases. However, this can be minimized by carefully selecting the location of the points.

The idea behind the proof of Theorem 1 is that we can construct a piecewise linear approximation  $\rho$  to an optimal path  $\rho^*$  by approximately sampling  $\rho^*$  so that the following constraints are satisfied:

- 1) the distance between consecutive sample points  $x_k$  should not exceed a given constant  $\epsilon_\delta$ ;
- 2) each sample point  $x_k$  should be in an  $\epsilon_x$ -ball of the corresponding point  $\rho^*(\tau_k)$  in the path;
- 3) the difference between the derivatives of  $\rho^*$  and its approximation  $\rho$  should not exceed  $\epsilon_v$ .

By forcing  $\epsilon_\delta$ ,  $\epsilon_x$ , and  $\epsilon_v$  to be sufficiently small, one can get piecewise linear paths whose costs are arbitrarily close to  $\rho^*$ . It turns out that if one examines closely the proof of Theorem 1, in particular equation (17), one concludes that  $\epsilon_\delta$  can actually be large and that it suffices that both  $\epsilon_x$  and

$$g_{x,k}\epsilon_x + g_{v,k}\epsilon_v + (g_{x,k} + Ag_{v,k})\|x_k - x_{k-1}\| \quad (18)$$

be small for every  $k$ , where the constants  $g_{x,k}$  and  $g_{v,k}$  are defined in (12). This shows that one can actually allow  $\|x_k - x_{k-1}\|$  to be large in some regions, provided that  $g_{x,k} + g_{v,k}A$  be sufficiently small on those regions. To take advantage of this, one should not sample  $\mathcal{R}$  uniformly. The honeycomb-like sampling shown in Figure 1 can be used to keep (18) small with sparse sampling. Indeed, by keeping the

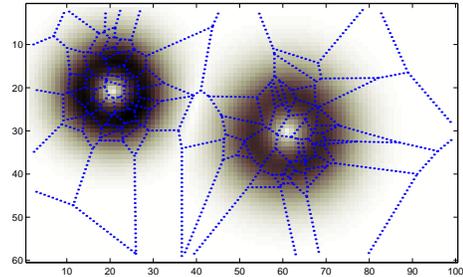


Fig. 1. Honeycomb sampling

diameter of the cells inversely proportional to  $g_{x,k} + g_{v,k}A$ , one makes sure that it is possible to pick the  $x_k$  appropriately spaced, while minimizing the overall number of sample points. Note that to be able to keep  $x_k$  in a small  $\epsilon_x$ -ball of  $\rho^*(\tau_k)$ , the edges of the cells should be finely sampled. This type of spacing can be efficiently obtained using the following procedure:

*Algorithm 1 (Honeycomb sampling):*

- 1) Extract randomly  $K$  points  $\mathcal{Z} := \{z_k \in \mathcal{R} : k = 1, 2, \dots, K\} \subset \mathbb{R}^n$ , with a spacial probability density over  $\mathcal{R}$  proportional to

$$\left( \sup_{v \in \mathcal{V}} \|\nabla_x \ell(x, v)\| + A \|\nabla_v \ell(x, v)\| \right)^n.$$

The distance between points in a particular area is then roughly inversely proportional to

$$K^{\frac{1}{n}} \sup_{v \in \mathcal{V}} \|\nabla_x \ell(x, v)\| + A \|\nabla_v \ell(x, v)\|. \quad (19)$$

- 2) Compute the Voronoi diagram generated by the points in  $\mathcal{Z}$ . The size of the resulting cells in a particular area is roughly proportional to the distance between the points in that area, which is inversely proportional to (19).

- 3) Construct  $\mathcal{X}$  by sampling the edges of the Voronoi diagram sufficiently finely so that it is possible to choose points  $x_k$  in an  $\epsilon_x$ -ball of any point where an optimal path would cross the cell boundary. (See Lemma 1)  $\square$

This algorithm was used to produce the sampling in Figure 1. In this figure the background color represents the magnitude of  $\sup_{v \in \mathcal{V}} \|\nabla_x \ell(x, v)\| + A \|\nabla_v \ell(x, v)\|$ , with dark representing large values. Increasing the number of generators  $K$ , decreases the term  $(g_{x,k} + Ag_{v,k}) \|x_k - x_{k-1}\|$  in (18), whereas increasing the density of sampling over the edges of the Voronoi diagram decreases  $\epsilon_x$ . It should be noted that this type of sampling does *not* correspond to a finite-elements approximation to the continuous HJB equation.

We considered a couple of alternative sampling algorithms and contrasted them with honeycomb sampling. The first method is the simplest and does not explore the structure of the cost weight  $\ell$ :

*Algorithm 2 (Randomized uniform sampling):* Construct  $\mathcal{X}$  by randomly extracting  $N$  points, with a uniform spacial probability density over  $\mathcal{R}$ .  $\square$

The following method is also inspired by (18) but simply attempts to minimize the term  $(g_{x,k} + Ag_{v,k}) \|x_k - x_{k-1}\|$ .

*Algorithm 3 (Randomized gradient-based sampling):* Construct  $\mathcal{X}$  by randomly extracting  $N$  points, with a spacial probability density over  $\mathcal{R}$  proportional to  $(\sup_{v \in \mathcal{V}} \|\nabla_x \ell(x, v)\| + A \|\nabla_v \ell(x, v)\|)^n$ .  $\square$

In the next section we compare the performance of these three sampling algorithms in the context of minimum-risk path planning.

## V. MINIMUM-RISK PATH PLANNING FOR UAVS

Consider a group of  $m$  UAVs flying in a region  $\mathcal{R} \subset \mathbb{R}^3$  under the threat of  $k$  SAM sites located at positions  $z_1, z_2, \dots, z_k \in \mathbb{R}^3$ . In this context one would like to compute a path  $\rho : [0, T] \rightarrow \mathcal{R}$  for the group of UAVs that starts at an initial position  $x_i$  and ends at a final position  $x_f$ , maximizing the probability that the UAVs will survive the journey.

In general different UAVs may have different defensive/stealth capabilities and therefore their probabilities of survival are distinct. Because of this, minimal-risk path planning is a multi-criteria optimization problem. We will pursue here Pareto-optimal paths, i.e., paths for which the probability of any single UAV surviving cannot be improved without decreasing the survivability of another UAV in the group.

Denoting by  $p_j^{\text{survive}}[\rho]$  the probability that the  $j$ th UAV safely reaches the destination, Pareto-optimal (maximal) paths can be obtained as the solution to single-criteria optimization problems of the form

$$\max_{\rho} \sum_{j=1}^m \lambda_j p_j^{\text{survive}}[\rho],$$

where the  $\lambda_j$  denote positive constants [4]. Assuming that velocities are normalized so the maximum speed of the slowest UAV is equal to one, the optimization should be

performed as  $\rho$  ranges over the set  $\mathcal{P}$  considered in the previous sections. Note that in general, risk is minimized for the maximum speed so there is no reason to consider paths with speeds smaller than the maximum. We ignore here all constraints posed by the aircraft dynamics, other than its maximum speed. The path generated by this optimization would serve as a reference trajectory for the group, to be used by algorithms such as the ones proposed in [6, 15, 16]. One could also take fuel consumption and path length as additional criteria for Pareto-optimality. Although we do not pursue this here, it would be straightforward.

### A. Risk Model

The probability of the  $j$ th UAV being hit by the  $i$ th SAM in an elementary time interval  $dt$  is assumed to be given by

$$\eta_{ij}(x, \dot{x}, z_i) dt,$$

where  $x$  and  $\dot{x}$  denote the position and velocity of the group of UAVs. The function  $\eta_{ij}$  is called the *risk density for the  $j$ th UAV with respect to the  $i$ th SAM*. When the SAMs operate independently, the probability of the  $j$ th UAV surviving all the SAMs is given by

$$\prod_{i=1}^n (1 - \eta_{ij}(x, \dot{x}, z_i) dt).$$

The main reason why risk density functions depend on the position of the SAMs and the position and velocity of the UAVs, is that the radar signature of the UAVs is a function of their distance and flight angle with respect to the SAMs.

Suppose now that the group of UAVs fly along a path  $\rho : [0, T] \rightarrow \mathcal{R}$ . Assuming that the probability of being hit over disjoint path elements is independent, the probability  $p_j^{\text{survive}}[\rho]$  of the  $j$ th UAV surviving the whole path  $\rho$  is given by the limit as  $dt \rightarrow 0$  of

$$\prod_{k=0}^{T/dt} \prod_{i=1}^n \left(1 - \eta_{ij}(\rho(kdt), \dot{\rho}(kdt), z_i) dt\right).$$

Taking log and making  $dt \rightarrow 0$ , we obtain<sup>3</sup>

$$\log p_j^{\text{survive}}[\rho] = - \sum_{i=1}^n \int_0^T \eta_{ij}(\rho(t), \dot{\rho}(t), z_i) dt.$$

We therefore conclude that, under the previous risk model, we can express  $p_{\text{survive}}[\rho]$  as

$$p_j^{\text{survive}}[\rho] = e^{-\int_0^T \ell_j(\rho(t), \dot{\rho}(t)) dt},$$

where

$$\ell_j(x, v) := \sum_{i=1}^n \eta_{ij}(x, v, z_i). \quad (20)$$

This model is consistent with the expectation that if one remains under danger for a long amount of time, the probability of survival eventually converges to zero.

Since the function  $s \mapsto e^{-s}$  is monotone decreasing, paths that are Pareto-optimal (maximal) with respect to the rewards

<sup>3</sup>Here, we used the fact that  $\log(1 - \epsilon) \approx -\epsilon$  for  $\epsilon \ll 1$ .

$p_j^{\text{survive}}[\rho]$  are also Pareto-optimal (minimal) with respect to the costs

$$J_j[\rho] := \int_0^T \ell_j(\rho(t), \dot{\rho}(t)) dt. \quad (21)$$

We can therefore find these paths by solving the Weighted Anisotropic Shortest-Path Problem 1 considered in Sections III–IV.

### B. Numerical results

In this section we analyze the performance of the algorithms proposed to solve Problem 1, in the context of the minimum-risk path planning problem formulated above. We utilize a cost of the form (21), computed from a realistic risk density function.

The risk density function  $\eta_{ij}$  defined before depends on the distance and attitude of the UAVs with respect to the SAM sites. This is because SAMs are usually guided by a tracking radar that locks on the target UAV and guides the SAM until interception. Therefore,  $\eta_{ij}$  depends significantly on the aircraft’s Radar Cross Section (RCS), which is a measure of its ability to reflect radar signals in the direction of the radar receiver. It turns out that the RCS of an aircraft is a function of the azimuth and the elevation angles of the line-of-sight vector from the UAV to the radar. Since these angles are defined in the body frame, the RCS—and consequently the risk density functions  $\eta_{ij}$ —depend on the relative attitude of the aircraft with respect to the radar and therefore on the aircraft’s position and velocity. This leads to an anisotropic cost-weighting  $\ell_j$  given by (20). The results summarized below use the strongly anisotropic risk density functions consistent with RCSs taken from a challenge problem set forth by Boeing for the DARPA program Mixed Initiative Control of Automa-teams (MICA) [2, 3]. The positions and characteristics of the SAM sites were also taken from the same challenge problem.

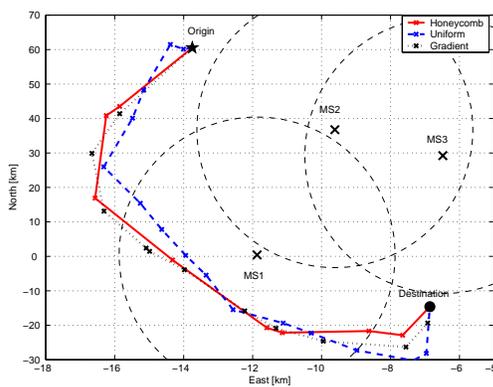


Fig. 2. The Scenario and Typical Minimum-Risk Paths

In this study we consider the scenario shown in Figure 2, which is representative of typical minimal-risk path planning problem. This scenario considers three medium range SAM sites (MS1, MS2, and MS3), appearing in the figure surrounded by circles that indicate their maximum firing

range. However, one should emphasize that it is possible for the UAVs to fly well inside the SAMs’ ranges without being threatened, provided that their RCSs are kept low by appropriate choice of position and attitude. The figure also shows three minimum-risk paths, each obtained using a different sampling method.

Figure 3 shows typical paths obtained using the honeycomb, uniform, and gradient-based sampling methods. These plots illustrate how honeycomb sampling can produce better result with a much smaller sampling density than the other methods.

Figure 4 summarizes the results obtained from running a large number of optimizations on the test scenario. In the vertical axis we plot the optimal cost obtained and in the horizontal axis the time it took to compute it. The algorithms were implemented in MATLAB and ran on a Dell Dimension 4500 workstation (Pentium 4 processor, 2GHz clock, 768Mb of RAM) under Linux. We compare the results obtained using the three sampling algorithms described in Section IV: honeycomb, randomized uniform, and randomized gradient-based. For each algorithm we considered several sampling densities, leading to distinct execution times and distinct optimal costs. We can see that, for the same computation time, honeycomb sampling consistently produces lower costs than the other methods. Moreover, for computation times larger than roughly 15 seconds, the majority of the honeycomb optimizations resulted in very good paths with costs below  $10^4$ . The same is not true for the other sampling methods that proved much less consistent.

TABLE I  
MEAN AND STANDARD DEVIATION OF THE COSTS

|                                 | honeycomb | uniform | gradient |
|---------------------------------|-----------|---------|----------|
| Mean                            | 8.6892    | 9.3878  | 9.7667   |
| Standard Deviation ( $\sigma$ ) | 0.8294    | 0.8286  | 0.8006   |
| $3\sigma$ Upper Bound (99.7%)   | 11.1774   | 11.8767 | 12.1685  |

This is confirmed by Table I, which shows the mean value and the standard deviation of the costs obtained with each sampling method. The average cost from honeycomb sampling is about 7% smaller than the other ones and its  $3\sigma$  upper bound is also smaller. It should be noted that a 7% cost improvement is significant, because the costs obtained using sampling do not seem to be very far from the best achievable in this scenario (which is probably just below 7).

### VI. CONCLUSION

In this paper we propose an algorithm for the computation of weighted anisotropic shortest-paths, which reduces the continuous problem to an optimization over a finite graph that can be efficiently solved. The algorithm proposed restricts the search to paths formed by the concatenation of straight-line segments between points from a suitably chosen discretization of the continuous region. To maximize efficiency, the discretization should not be uniform. We propose a

<sup>4</sup>Essentially, most honeycomb optimizations fit below the dashed line except for four points indicated by arrows. They resulted from particularly unfavorable configurations of cells. This type of situation is unavoidable with randomized methods.

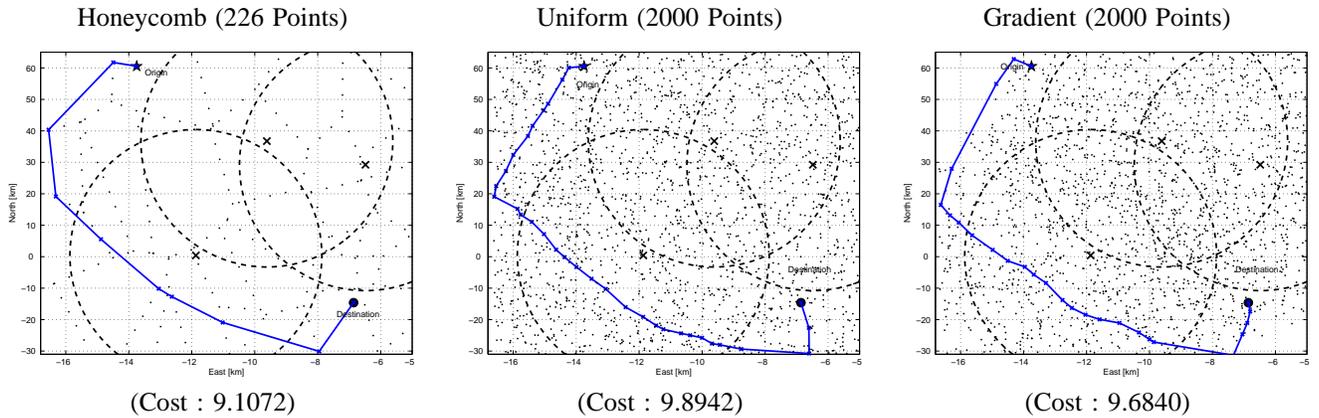


Fig. 3. Sampling Points and Corresponding Optimal Path for Each Method

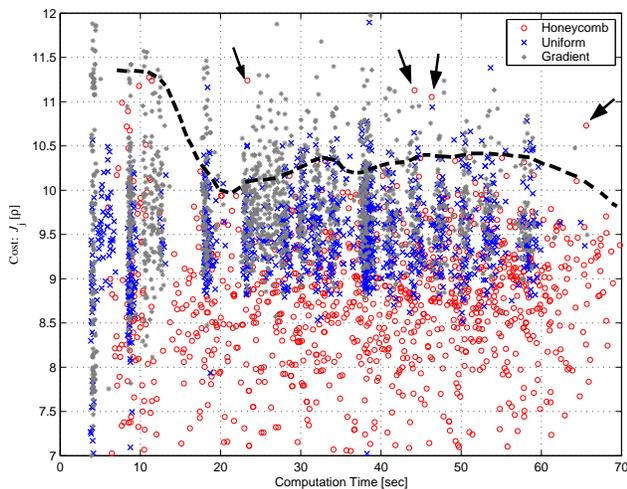


Fig. 4. Cost Comparisons with respect to Computation Time

“honeycomb” sampling algorithm that minimizes the cost penalty introduced.

This methodology is applied to the computation of paths for groups of UAVs that minimize the risk of being destroyed by ground defenses. We show that this problem can be formulated as a shortest-path optimization and that honeycomb sampling method can efficiently produce low-risk paths with less computation time than other methods, for the same or better costs.

Future work will include the computation of non-conservative bounds on the cost penalty introduced by discretization with honeycomb and other sampling algorithms. We are also studying the integration of sampling and the graph optimization algorithms to avoid oversampling.

#### REFERENCES

- [1] L. Aleksandrov, A. Maheshwari, and J.-R. Sack. Approximation algorithms for geometric shortest path problems. In *Proc. of the thirty-second annual ACM symposium on Theory of computing*, pages 286–295. ACM Press, 2000. ISBN 1-58113-184-4.
- [2] Boeing. Challenge problem #1.1, Feb. 2003. DARPA’s program “Mixed Initiative Control for Automa-teams.”.
- [3] *User Guide for the Open Experimental Platform (OEP), version 1.1, revision 11*. Boeing, Mar. 2003.
- [4] S. P. Boyd and C. H. Barratt. *Linear Controller Design: Limits of Performance*. Prentice-Hall, New Jersey, 1991.
- [5] L. C. Polymenakos, D. P. Bertsekas, and J. N. Tsitsiklis. Implementation of efficient algorithms for globally optimal trajectories. *IEEE Trans. on Automat. Contr.*, 43(2):278–283, Feb. 1998.
- [6] J. A. Fax and R. M. Murray. Information flow and cooperative control of vehicle formations. In *Proc. of the 15th World Congress of Int. Federation of Automat. Contr.*, July 2002.
- [7] I. M. Gelfand and S. V. Fomin. *Calculus of Variations*. Selected Publications in the Mathematical Sciences. Prentice-Hall, Englewood Cliffs, N. J., 1963.
- [8] J. Hershberger and S. Suri. An optimal algorithm for euclidean shortest paths in the plane. *SIAM J. on Computing*, 28(6):2215–2256, 1999. ISSN 0097-5397.
- [9] S. Kapoor, S. N. Maheshwari, and J. S. B. Mitchell. An efficient algorithm for euclidean shortest paths among polygonal obstacles in the plane. *Discrete Comput. Geometry*, 18:377–383, 1997.
- [10] H. J. Kushner and P. G. Dupuis. *Numerical Methods for Stochastic Control Problems in Continuous Time*. Springer-Verlag, New York, 1992.
- [11] M. Lanthier, A. Maheshwari, and J.-R. Sack. Shortest anisotropic paths on terrains. In J. Wiedermann, P. van Emde Boas, and M. Nielsen, editors, *Proc. 26th Int. Colloquium Automata, Languages and Programming (ICALP’99)*, volume 1644 of *Lecture Notes in Computer Science*, pages 524–533. Springer-Verlag, Berlin, 1999.
- [12] C. S. Mata and J. S. B. Mitchell. A new algorithm for computing shortest paths in weighted planar subdivisions (extended abstract). In *Proc. of the thirteenth annual symposium on Computational geometry*, pages 264–273. ACM Press, 1997. ISBN 0-89791-878-9.
- [13] J. S. B. Mitchell. Geometric shortest paths and network optimization. In Sack and Urrutia [18].
- [14] J. S. B. Mitchell and C. H. Papadimitriou. The weighted region problem: finding shortest paths through a weighted planar subdivision. *J. of the ACM*, 38(1):18–73, 1991. ISSN 0004-5411.
- [15] P. Ogren, E. Fiorelli, and N. E. Leonard. Formations with a mission: Stable coordination of vehicle group maneuvers. In *Proc. of the Int. Symposium on the Mathematical Theory of Networks and Syst.*, Aug. 2002.
- [16] A. Pant, P. Seiler, and K. Hedrick. Mesh stability of look-ahead interconnected systems. *IEEE Trans. on Automat. Contr.*, 47(2):403–407, Feb. 2002.
- [17] N. C. Rowe and R. S. Ross. Optimal grid-free path planning across arbitrarily contoured terrain with anisotropic friction and gravity effects. *IEEE Trans. Robot. Automat.*, 6(5):540–553, Oct. 1990.
- [18] J.-R. Sack and J. Urrutia, editors. *Handbook of Computational Geometry*. Elsevier, Amsterdam, 2000.
- [19] S. Sundar and Z. Shiller. Optimal obstacle avoidance based on the hamilton-jacobi-bellman equation. *IEEE Trans. Robot. Automat.*, 13(2):305–310, 1997.
- [20] J. N. Tsitsiklis. Efficient algorithms for globally optimal trajectories. *IEEE Trans. on Automat. Contr.*, 40(9):1528–1538, Sept. 1995.