Disambiguation Protocols Based on Risk Simulation

Donniell E. Fishkind, Carey E. Priebe, *Member, IEEE*, Kendall E. Giles, Leslie N. Smith, and Vural Aksakalli

***Abstract*—Suppose there is a need to swiftly navigate through a spatial arrangement of possibly forbidden regions, with each region marked with the probability that it is, indeed, forbidden. In close proximity to any of these regions, you have the dynamic capability of *disambiguating* the region and learning for certain whether or not the region is forbidden—only in the latter case may you proceed through that region. The central issue is how to most effectively exploit this disambiguation capability to minimize the expected length of the traversal. Regions are never entered while they are possibly forbidden, and thus, no risk is ever actually incurred. Nonetheless, for the sole purpose of deciding where to disambiguate, it may be advantageous to *simulate* risk, temporarily pretending that possibly forbidden regions are riskily traversable, and each potential traversal is weighted with its level of undesirability, which is a function of its traversal length and traversal risk. In this paper, the *simulated risk disambiguation protocol* is introduced, which has you follow along a shortest traversal—in this undesirability sense—until an ambiguous region is about to be entered; at that location, a disambiguation is performed on this ambiguous region. (The process is then repeated from the current location, until the destination is reached.) We introduce the *tangent arc graph* as a means of simplifying the implementation of simulated risk disambiguation protocols, and we show how to efficiently implement the simulated risk disambiguation protocols that are based on linear undesirability functions. The effectiveness of these disambiguation protocols is illustrated with examples, including an example that involves mine countermeasures path planning.**

***Index Terms*—Canadian traveller problem, disambiguation protocol, probabilistic path planning, random disambiguation path, visibility graph.**

I. Introduction

Section I-B provides an overview of this paper and a review of the literature, but we begin in Section I-A with a formulation of the setting in which we will be working. More discussion of this setting can be found in [1].

A. Random Disambiguation Paths

Let $\mathcal{S}$ be a marked point process on $S \subseteq \mathbf{R}^2$; this process generates random detections X_T, $X_F \subseteq S$ (respectively called *true* and *false* detections), and it generates random marks $\rho_T : X_T \to (0,1]$ and $\rho_F : X_F \to (0,1]$. When observing a realization of this process, you only see $X := X_T \bigcup X_F$ and $\rho := \rho_T \bigcup \rho_F$, but you may assume that, independently for all $x \in X$, $\rho(x)$ is the probability—conditioned on the observed values X and ρ—that $x \in X_T$. Indeed, for the remainder of this paper, the specific values X and ρ have been observed, and all discussion of probability is accordingly conditioned. See Section III for an example realization of such a marked point process; the detections are the centers of the discs visualized in Fig. 4.

For every detection x, the open disc about x of a given radius $r > 0$ is denoted R_x. Given a starting point $s \in \mathbf{R}^2$ and a destination point $t \in \mathbf{R}^2$, you seek a continuous s, t curve in $(\bigcup_{x \in X_T} R_x)^C$ of the shortest achievable arclength. Without means of verifying which detections in X are true, you could not do better than the shortest s, t curve in $(\bigcup_{x \in X} R_x)^C$, which is denoted $q_{s,t,X}$. (The curve $q_{s,t,X}$ can be computed using the visibility graph described in Section I-C.) However, what makes our setting interesting is a dynamic capability of *disambiguating* detections from the boundaries of their associated discs; that is to say, when the curve is on ∂R_x for any $x \in X$, you can dynamically discover whether $x \in X_T$ or $x \in X_F$, and in the latter case, the curve is permitted to proceed through R_x. However, a fixed cost $c \geq 0$ (reflecting the cost of disambiguation) is added to the Euclidean length of the curve for each disambiguation, and it is assumed that there are a maximum of K disambiguations that may be performed during an s, t traversal. The broad goal here is to efficiently exploit this disambiguation capability in order to minimize the traversal curve's (expected) Euclidean length.

A *disambiguation protocol* is a function $\mathcal{D}$ that, to any such s, t, X, ρ, K, assigns a detection $x \in X$ and a point $y \in \partial R_x$ (we explicitly allow $y = t$, in which case x is not defined). Given a disambiguation protocol $\mathcal{D}$, the *random disambiguation path* $p_{\mathcal{D}}$ (as in [1]) is the s, t curve in $(\bigcup_{x \in X_T} R_x)^C$, which is realized by the following procedure: Suppose $\mathcal{D}$ associates $x \in X$ and $y \in \partial R_x$ to s, t, X, ρ, K. Traverse $q_{s,y,X}$ from s to y (e.g., by finding the shortest path in an appropriate visibility graph, as described in Section I-C). If $y = t$, then terminate (in particular, if $K = 0$, then it is required that $y = t$), otherwise, disambiguate detection x. Recursively repeat this entire procedure using y in place of s, decrementing K by 1 and updating X and ρ as follows: If the disambiguation had just discovered that $x \in X_T$, then update $\rho(x) := 1$, and if the disambiguation had just discovered that $x \in X_F$, then remove x from X.

The random disambiguation path $p_{\mathcal{D}}$ is an s, t-curve-valued random variable even after X and ρ are observed since the emerging outcomes of the disambiguations dictated by the

Manuscript received July 5, 2005; revised February 13, 2006. The work of D. E. Fishkind, C. E. Priebe, and V. Aksakalli was supported in part by the Office of Naval Research under Grant N000140410483 and Grant N000140610013. The work of K. E. Giles was supported by the Office of Naval Research under Grant N000140410483. This paper was recommended by Associate Editor J. Lambert.

D. E. Fishkind, C. E. Priebe, and V. Aksakalli are with the Department of Applied Mathematics and Statistics, Whiting School of Engineering, The Johns Hopkins University, Baltimore, MD 21218 USA (e-mail: fishkind@ams.jhu.edu; cep@jhu.edu; ala@jhu.edu).

K. E. Giles is with the Department of Computer Science, The Johns Hopkins University, Baltimore, MD 21218 USA (e-mail: kgiles@cs.jhu.edu).

L. N. Smith is with the Naval Research Laboratory, Washington, DC 20375 USA (e-mail: lnsmith@ccs.nrl.navy.mil).

Digital Object Identifier 10.1109/TSMCA.2007.902634

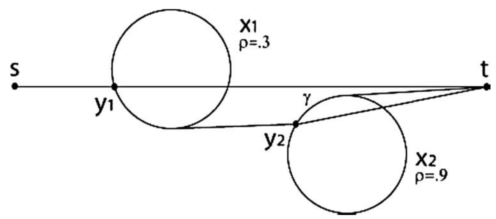


Fig. 1. Example of a random disambiguation path.

protocol are still random; in fact, the distribution of $p_{\mathcal{D}}$ is specified through ρ.

To illustrate, suppose $K = 2$ and s, t, X, ρ are given as shown in Fig. 1, and consider one particular disambiguation protocol $\mathcal{D}$ which, for instance, dictates that the next (i.e., first) disambiguation be of detection x_1 at point y_1. Now, suppose, if it is discovered that $x_1 \in X_F$, $\mathcal{D}$ would then dictate that no more disambiguations be performed, and the curve should proceed to t. Furthermore, suppose, if it is instead discovered that $x_1 \in X_T$, $\mathcal{D}$ would then dictate that x_2 should be disambiguated next, i.e., at point y_2. Whether x_2 is revealed to be a true or false detection, $\mathcal{D}$ would then dictate that we proceed directly to t, since no more disambiguations are available (currently, $K = 0$). There are three possible realizations of the random disambiguation path $p_{\mathcal{D}}$, each shown in Fig. 1: With $1 - \rho(x_1) = 1 - 0.3$ probability, $p_{\mathcal{D}}$ traverses the points s, y_1, t; with $\rho(x_1)\rho(x_2) = (0.3)(0.9)$ probability, $p_{\mathcal{D}}$ traverses the points s, y_1, y_2, t, employing the curve γ at the traversal conclusion; and with $\rho(x_1)(1 - \rho(x_2)) = (0.3)(1 - 0.9)$ probability, $p_{\mathcal{D}}$ traverses the points s, y_1, y_2, t, employing the line segment $\overline{y_2, t}$ at the traversal conclusion. (Note that in between disambiguations, the s, t curve traverses the shortest curves, avoiding all possibly forbidden risk regions—using the notion of a visibility graph described in Section I-C). If the lengths of these three paths are 6, 8, 7, respectively, and if the cost of disambiguation was $c = 5$, then the expected length of $p_{\mathcal{D}}$ is given by $(1 - 0.3)(6 + 1 * 5) + (0.3)(0.9)(8 + 2 * 5) + (0.3)(1 - 0.9)(7 + 2 * 5)$.

In the example above, we illustrated one particular disambiguation protocol $\mathcal{D}$; a different choice of protocol may, indeed, yield a significantly lower expected length. Unfortunately, choosing (from among all disambiguation protocols) an optimal protocol (which results in the minimum expected length) is not currently practical, either analytically or computationally, as we discuss in Section I-B. The purpose of this paper is to present a class of efficiently computable, suboptimal but effective disambiguation protocols; we call them *simulated risk disambiguation protocols*.

B. Overview

The problem that we describe here is a minor modification of the stochastic obstacle scene problem (SOSP) of Papadimitriou and Yannakakis [2], who also describe a discrete version of this problem, which they call the Canadian traveller's problem (CTP). (In CTP, a short traversal is desired through a finite graph whose edges are marked with their respective probabilities of being traversable, and every edge's status can be dynamically discovered when encountered.) Papadimitriou and Yannakakis prove the intractability of several variants of SOSP and CTP. (For more information on CTP, see [3].)

CTP is a special case of the stochastic shortest paths with recourse (SPR) problem of Andreatta and Romeo [4], who present a stochastic dynamic programming formulation for SPR and note its intractability. Polychronopoulos and Tsitsiklis [5] also present a stochastic dynamic programming formulation for SPR and then prove the intractability of several variants. Provan [6] proves that SPR is intractable even if the underlying graph is directed and acyclic.

The underlying difficulty in obtaining a tractable stochastic dynamic programming formulation of these problems—even in the discrete setting—is that, in order for actions to be considered at any given location, there is a need to know the current ambiguous/true/false status of all of the detections, and the exponentially many such possibilities need to be accordingly incorporated. Andreatta and Romeo [4] note that if there is a limit of $K = 1$ disambiguations allowed, then SPR can be efficiently solved. Indeed, we are willing to assume here a limit K on the number of allowed disambiguations, but solving our random disambiguation problem—even a discrete variant of it—is not currently practical, unless K has a very small value.

Heuristics are suggested for CTP and SPR in [7]–[9] and [5], but they would not be applicable to the problem that we address here in this paper without initially approximating and recasting our continuous setting to the setting of a finite graph, in which case the resolution of the discretization drives up the number of vertices and edges in the approximating graph. By contrast, the algorithm that we propose here is polynomial time solely in the number of detections $|X|$.

The principal aim of this paper is to introduce the simulated risk disambiguation protocol (which is, effectively, a particular policy in the stochastic dynamic programming formulation) and its associated random disambiguation path. They are defined in Section II-A, and their evaluation is greatly simplified through the use of the tangent arc graph introduced in Section II-B. The tangent arc graph is an extension of the visibility graph [10] detailed next in Section I-C. In Section II-C, we describe how to efficiently evaluate simulated risk disambiguation protocols that are generated by linear undesirability functions, and we show how to efficiently realize their associated random disambiguation paths. Then, in Section III, we illustrate these protocols by using an example that involves mine countermeasures path planning. Other examples are found in Section IV.

In practice, suppose you are presented with the problem described in Section I, i.e., you need to traverse from s to t and have observed X and ρ (and you are permitted K disambiguations), and further suppose that you have decided to, indeed, utilize a simulated risk disambiguation protocol to realize the associated random disambiguation path to accomplish your s, t traversal. The question still remains as to how to choose the most effective simulated risk disambiguation protocol from among the many simulated risk disambiguation protocols that exist. Section IV is concerned (given s, t, X, ρ, K) with how to select the best (or a nearly optimal) simulated risk disambiguation protocol from among the family of simulated

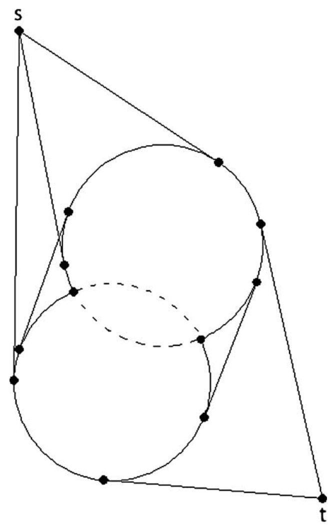


Fig. 2. Example of a visibility graph. The dashed arcs are not visibility graph edges since they are not in $\partial(\cup_{x\in X}R_x)$.

risk disambiguation protocols that are generated by linear undesirability functions.

C. Visibility Graph

We conclude this background section with the construction of the visibility graph associated with s, t, X; as mentioned, this visibility graph can be used to compute $q_{s,t,X}$. (In addition, the details of this construction will be quite relevant to the construction of the tangent arc graph introduced in Section II-B.) Our visibility graph is an adaptation of the visibility graph from [10] and [11], more similar to the generalized visibility graph in [12].

Let s, t and X be specified. For distinct points a, $b \in \{s,t\} \bigcup \partial(\cup_{x\in X}R_x)$, we call the closed line segment $\overline{a,b}$ a *tangent segment*, provided that 1) for all $r \in \{a,b\}\backslash\{s,t\}$, $\overline{a,b}$ is tangential to $\partial(\cup_{x\in X}R_x)$ at r and 2) the relative interior of $\overline{a,b}$ is contained in the interior of $[(\cup_{x\in X}R_x) \cup \{s,t\}]^C$.

The *visibility graph* associated with s, t, X is defined as follows. Its vertex set consists of s, t, all points of $\partial(\cup_{x\in X}R_x)$ that intersect a tangent segment, and all points of $\partial(\cup_{x\in X}R_x)$ at which two or more ∂R_x's intersect. The edge set of the visibility graph consists of all tangent segments and all connected components of $\partial(\cup_{x\in X}R_x)$ after the vertices of the visibility graph are removed (the latter edges are segments of arc from circles). The graph-theoretic endpoints of these edges are their line and arc endpoints, respectively, and each edge is weighted with its arclength. An example of a visibility graph is shown in Fig. 2.

It is a well-known (and true) folk theorem that $q_{s,t,X}$ is the shortest s, t path in the visibility graph associated with s, t, X. Since every pair of nonidentical ∂R_x's have at most four mutually tangential lines and two points of intersection, the number of vertices and edges in this visibility graph is $O(|X|^2)$ each. Thus, Dijkstra's algorithm, with a heap implementation applied to this visibility graph, yields $q_{s,t,X}$ in $O(|X|^2 \log |X|)$ operations, and the naive construction of the visibility graph performs $O(|X|^3)$ assignment, arithmetic, and trigonometric operations.

II. Simulating Risk

We now introduce our main idea—the simulated risk disambiguation protocol—which gives rise to an associated simulated risk random disambiguation path.

A. Simulated Risk Disambiguation Protocol

Of course, in our framework, you will never enter regions of the form $R_x : x \in X$ while they are possibly forbidden, and thus, you never experience actual risk. However, for the purpose of deciding the next disambiguation point, the simulated risk disambiguation protocol temporarily pretends (*simulates*) that the possibly forbidden regions are riskily traversable.

Under this simulation of risk, for **any** s, t curve p (allowing intersection with $\bigcup_{x\in X} R_x$), define its Euclidean length $\ell^e p$ in the usual way, and its *risk length* $\ell^r p := -\log \prod_{x\in X:p\cap R_x\neq\emptyset}(1-\rho(x))$; this negative logarithm of the probability that p is permissibly traversable is a measure of the risk in traversing p—if you were willing to take on risk. An *undesirability function* is any function $g : \mathbf{R}_{\geq 0} \times \mathbf{R}_{\geq 0} \to \mathbf{R}$ that is monotonically nondecreasing in its arguments; that is to say, for all $u_1, u_2, v_1, v_2 \in \mathbf{R}_{\geq 0}$ such that $u_1 \leq u_2$ and $v_1 \leq v_2$, it holds that $g(u_1, v_1) \leq g(u_2, v_2)$. The number $g(\ell^e p, \ell^r p)$ is thought of as a measure of the undesirability of p in the sense that, if you were required to traverse from s to t under the simulation of risk and without a disambiguation capability, you would want to traverse the s, t curve $\phi_g := \arg\min_{s,t \text{ curves } p} g(\ell^e p, \ell^r p)$. For this s, t curve ϕ_g, let $y \in \mathbf{R}^2$ be the last point of ϕ_g before ϕ_g intersects $\bigcup_{x\in X} R_x$, and say $x' \in X$ is the detection whose associated region $R_{x'}$ the curve ϕ_g was entering at y. (If there is no intersection between ϕ_g and $\bigcup_{x\in X} R_x$, then $y := t$.) Back in our setting (where there is a disambiguation capability and you may not experience risk), the simulated risk disambiguation protocol $\mathcal{D}_g$ is defined as assigning this x' and y to s, t, X, ρ, K (provided that $K > 0$).

Thus, the simulated risk random disambiguation path $p_{\mathcal{D}_g}$ follows a shortest s, t curve (in the sense of g and under the simulation that potentially forbidden disks are riskily traversable) until it encounters an ambiguous region (which, in actuality, it cannot enter without disambiguation), at which point a disambiguation is performed, and the whole process is repeated using the current location in place of s (and the updated information on ρ, X, and K).

Note that for the particular undesirability function $g(u,v) = u + \infty_{v>0}$ (where $\infty_{v>0}$ denotes ∞ or 0 accordingly as $v > 0$ or $v = 0$), it holds that $p_{\mathcal{D}_g} = q_{s,t,X}$, which is the s, t curve that you would traverse if you did not have the disambiguation capability (and you were still not permitted to take any risk). In Section IV, we will discuss how, in practice, you would select an undesirability function g to use; we advocate choosing—from among a specific family of undesirability functions—the undesirability function whose associated random disambiguation path has the minimum expected length. As long as this family of undesirability functions that you

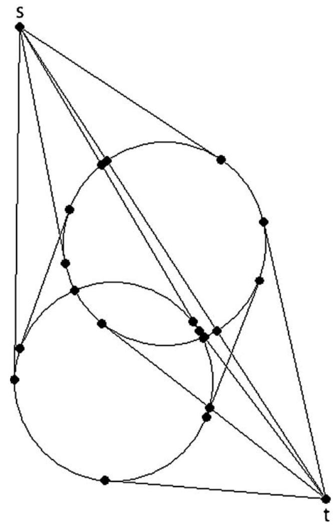


Fig. 3. Example of a tangent arc graph. Here, there were 11 general tangent segments; in particular, one of these was the line segment $\overline{s,t}$, which intersected boundaries of discs in two places. These two intersections became vertices in $\text{TAG}_{s,t,X}$ and, upon removal from $\overline{s,t}$, three $\text{TAG}_{s,t,X}$ edges are created in $\overline{s,t}$'s place. In total, this $\text{TAG}_{s,t,X}$ has 21 vertices and 37 edges; 16 of these edges are line segments, and 21 are segments of arc from circles.

choose from also includes the function $g(u,v) = u + \infty_{v>0}$ (which is, indeed,[1] the case for the strategy that we advocate in Section IV), you are always guaranteed to do no worse (in expectation) than the s, t curve that you would follow if you did not have a disambiguation capability (and you were not permitted any risk).

B. Tangent Arc Graph

Given an undesirability function g, in order to evaluate $\mathcal{D}_g$ and realize $p_{\mathcal{D}_g}$, one must be able to compute $\phi_g := \arg\min_{s,t \text{ curves } p} g(\ell^e p, \ell^r p)$. Although there are uncountably infinitely many s, t curves over which to minimize, we will use the monotonicity of g to show that ϕ_g must be a path in the *tangent arc graph* $\text{TAG}_{s,t,X}$, which is defined in the next paragraph and illustrated in Fig. 3, so that ϕ_g solves the finite optimization problem $\min_{s,t \text{ paths } p \text{ in } \text{TAG}_{s,t,X}} g(\ell^e p, \ell^r p)$.

For any distinct points $a, b \in \{s,t\} \cup (\bigcup_{x\in X} \partial R_x)$, we say that the closed line segment $\overline{a,b}$ is a *general tangent segment*, provided that, for all $r \in \{a,b\}\backslash\{s,t\}$, $\overline{a,b}$ is tangential to ∂R_x for some $x \in X$. The vertex set of $\text{TAG}_{s,t,X}$ consists of s, t, all points of intersection between any general tangent segment and any ∂R_x (over all $x \in X$), and all points of intersection between two or more ∂R_x's. The edge set of $\text{TAG}_{s,t,X}$ consists of all connected components of all general tangent segments after the vertices of $\text{TAG}_{s,t,X}$ are removed, and all connected components of $\cup_{x\in X}\partial R_x$ after the vertices of $\text{TAG}_{s,t,X}$ are removed. An example of a $\text{TAG}_{s,t,X}$ is shown in Fig. 3.

Note that the graph $\text{TAG}_{s,t,X}$ is a topological superimposition of all the (exponentially many) visibility graphs generated by s, t, Y over all $Y \subseteq X$. Hence, if ϕ_g is the shortest s, t curve in the sense of g, then for $Y = \{x \in X : \phi_g \cap R_x = \emptyset\}$, we have that ϕ_g is a path in the visibility graph associated with s, t, Y. Thus, in particular, ϕ_g is a path in $\text{TAG}_{s,t,X}$, as claimed.

However, there are only $O(|X|^2)$ general tangent segments, each intersecting $O(|X|)$ regions of the form $R_x : x \in X$, so we have $O(|X|^3)$ vertices and $O(|X|^3)$ edges in $\text{TAG}_{s,t,X}$, and the number of operations to set up TAG is $O(|X|^3 \log |X|)$. In particular, there are only a finite number of s, t paths p that are candidates for being ϕ_g.

[1]The disambiguation protocol $\mathcal{D}_g$ associated with $g(u,v) = u + \infty_{v>0}$ is precisely the protocol $\mathcal{D}_\alpha$ discussed later in this paper with $\alpha = \infty$.

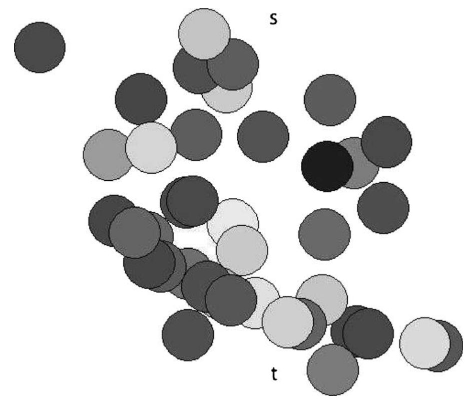


Fig. 4. Marked point process realization. Gray scale of discs reflects ρ of detections at respective disc's centers.

C. Linear Undesirability Functions

The simplest undesirability functions are the linear ones, where $g(u,v) = u + \alpha \cdot v$ for some fixed parameter $\alpha \geq 0$; in this case, we abbreviate the disambiguation protocol $\mathcal{D}_g$ to $\mathcal{D}_\alpha$. We next show that for any fixed value of α, $\mathcal{D}_\alpha$ is efficiently computable, and $p_{\mathcal{D}_\alpha}$ is efficiently realizable.

For each edge $f = \{a,b\}$ in $\text{TAG}_{s,t,X}$, we assign its Euclidean length $\ell^e f$ in the usual manner, and we define its risk length as $\ell^r f := -\sum_{x\in X} \mathcal{I}_{f\cap R_x \neq \emptyset}(\mathcal{I}_{a\in\partial R_x}/2 + \mathcal{I}_{b\in\partial R_x}/2)\log(1-\rho(x))$ where $\mathcal{I}$ is the indicator function; this definition is consistent with that of risk length for an s, t curve since, for any s, t path p in $\text{TAG}_{s,t,X,\rho}$, it holds that $\ell^r p = \sum_{\text{edges} f\in p} \ell^r f$ (provided that p never revisits any region R_x twice). Thus, ϕ_g may be found by running Dijkstra's algorithm on $\text{TAG}_{s,t,X}$ using the lengths $\ell^e f + \alpha \cdot \ell^r f$ for each edge f in $\text{TAG}_{s,t,X}$. The running time for Dijkstra's algorithm with a heap implementation is $O(|X|^3 \log |X|)$, so $\mathcal{D}_\alpha$ can be computed in $O(|X|^3 \log |X|)$ operations, and $p_{\mathcal{D}_\alpha}$ is thus realizable in $O(|X|^3 \log |X|)$ operations since K is a constant. (In particular, if there was no limit K on the number of disambiguations permitted, then $p_{\mathcal{D}_\alpha}$ is realizable in $O(|X|^4 \log |X|)$ operations.)

III. Mine Countermeasures Example

Minefield detection and localization is an important problem that is currently receiving much attention in the scientific and engineering literature (see, for instance, [13] and the references cited there). Witherspoon *et al.* [14] depict the operational concept for minefield reconnaissance via an unmanned aerial vehicle. Multispectral imagery of an area of interest is processed and a mine detection algorithm identifies locations of potential

TABLE I
x AND y COORDINATES AND ρ'S FOR MARKED POINT PROCESS REALIZATION

x-coordinates	y-coordinates	ρ	x-coordinates	y-coordinates	ρ	x-coordinates	y-coordinates	ρ
321.17	158.27	.59017	54.23	201.12	.54178	158.17	516.48	.43525
215.13	428.31	.61890	-145.67	703.06	.61714	-151.01	572.15	.56076
221.12	557.31	.64047	-166.36	299.42	.49173	296.16	163.31	.11649
163.31	186.14	.65636	28.31	205.03	.15269	-79.26	709.99	.56085
100.40	376.47	.51487	-105.75	262.20	.25748	185.31	182.18	.65266
116.39	110.84	.44124	-128.60	274.12	.62001	-61.19	345.12	.17183
-91.27	664.45	.16675	-82.87	248.29	.58308	105.47	509.80	.85147
-19.93	568.04	.59937	-310.23	402.92	.65428	-320.73	532.23	.33092
-35.11	242.61	.10329	-169.99	438.90	.64163	95.39	248.12	.18868
-78.75	396.14	.07310	-245.28	372.05	.52154	-166.45	180.33	.61082
-134.53	769.27	.19386	-258.45	641.03	.65670	111.60	640.10	.56529
-219.32	313.68	.57449	-455.72	742.57	.63987	-157.10	441.96	.64444
-242.22	321.51	.65655	-237.86	546.19	.13793	-269.98	379.65	.52802

mines (see [15]), with the collection of these points constituting our point process realization. The marks are posterior probabilities that the respective detections represent actual mines, as rendered by a postprocessing classification rule [16]–[20].

The following marked point process realization, as shown in Fig. 4, is referred to in [16] and [21] and has 39 potential mines whose x and y coordinates are listed in Table I.

The associated marks ρ in Table I were generated by the postclassification rule in [16]. Each disc R_x has a radius of 50, s is the point (0, 800), and t is the point (0, 100). Suppose a maximum of $K = 4$ disambiguations may be performed.

For $\alpha = 2000$, the first disambiguation is at the point $(-10.42, 286.09)$ and, regardless of the result, no more disambiguations are performed; thus, there are only two possible realizations of $p_{\mathcal{D}_{2000}}$, and they are shown in Fig. 5, along with their respective Euclidean lengths and their respective probabilities. In particular, $E\ell^e p_{\mathcal{D}_{2000}} = 0.89671 \cdot (707.97 + 1 \cdot c) + 0.10330 \cdot (1116.19 + 1 \cdot c) = 750.14 + c$, where c is the disambiguation cost.

For $\alpha = 100$, all seven possible realizations of $p_{\mathcal{D}_{100}}$ are shown in Fig. 6, which are labeled with their respective Euclidean lengths and probabilities. In particular, we can compute $E\ell^e p_{\mathcal{D}_{100}} = 0.89671 \cdot (707.97 + 1 \cdot c) + 0.040105 \cdot (714.90 + 3 \cdot c) + 0.038472 \cdot (859.37 + 4 \cdot c) + 0.012796 \cdot (831.04 + 3 \cdot c) + 0.0089469 \cdot (1188.77 + 4 \cdot c) + 0.0019532 \cdot (1185.40 + 4 \cdot c) + 0.0010226 \cdot (958.43 + 4 \cdot c) = 721.14 + 1.2570 \cdot c$.

Note that the coefficient of c in $E\ell^e p_{\mathcal{D}_\alpha}$ is the expected number of disambiguations. Observe also that $E\ell^e p_{\mathcal{D}_{100}} < E\ell^e p_{\mathcal{D}_{2000}}$ (i.e., $\alpha = 100$ is a better choice than $\alpha = 2000$) precisely when $c < 112.80$.

In practice, you will want to select a nonnegative value for α of minimum $E\ell^e p_{\mathcal{D}_\alpha}$. In Fig. 7, we plot $E\ell^e p_{\mathcal{D}_\alpha}$ as a function of α for the specific cost $c = 5$; here, $\arg\min_{\alpha \geq 0} E\ell^e p_{\mathcal{D}_\alpha}$ is seen to be the interval (30.23, 55.09). It turns out that $p_{\mathcal{D}_\alpha}$ is, in fact, identical to $p_{\mathcal{D}_{50}}$ for all values of α in (30.23, 55.09); a maximal interval I where $p_{\mathcal{D}_\alpha}$ is identical for all values of α in I will be called an *indifference interval*.[2] Here, in total, there are 11 indifference intervals, which are listed in Table II with their respective values of $E\ell^e p_{\mathcal{D}_\alpha}$ and the range of costs c where that interval is precisely $\arg\min_{\alpha \geq 0} E\ell^e p_{\mathcal{D}_\alpha}$.

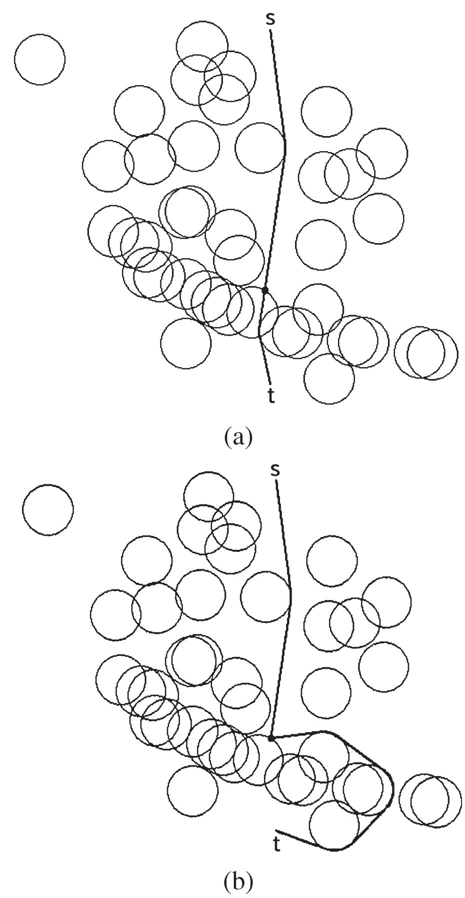


Fig. 5. All (two) possible realizations of $p_{\mathcal{D}_{2000}}$. (a) Length = 707.97, and prob = 0.89671. (b) Length = 1116.19, and prob = 0.10329.

In other words, suppose you were presented—in practice—the specific X and ρ given in Table I, with $s = (0, 800)$, $t = (0, 100)$, $K = 4$, and some disambiguation cost $c \geq 0$. If you would choose to traverse from s to t via a random disambiguation path based on a simulated risk disambiguation protocol using a linear undesirability function, then the particular value of the parameter α you should select depends on the particular disambiguation cost c. For example, if $c \in (0.0000, 4.1013)$ then, by comparing (for the various indifference intervals)

[2]Note that the specific value of c is relevant to neither $\mathcal{D}_\alpha$ nor $p_{\mathcal{D}_\alpha}$, so c has no influence in the establishment of indifference intervals. However, c will affect $\ell^e p_{\mathcal{D}_\alpha}$, $E\ell^e p_{\mathcal{D}_\alpha}$, and $\arg\min_{\alpha \geq 0} E\ell^e p_{\mathcal{D}_\alpha}$, so c will influence our choice of α.

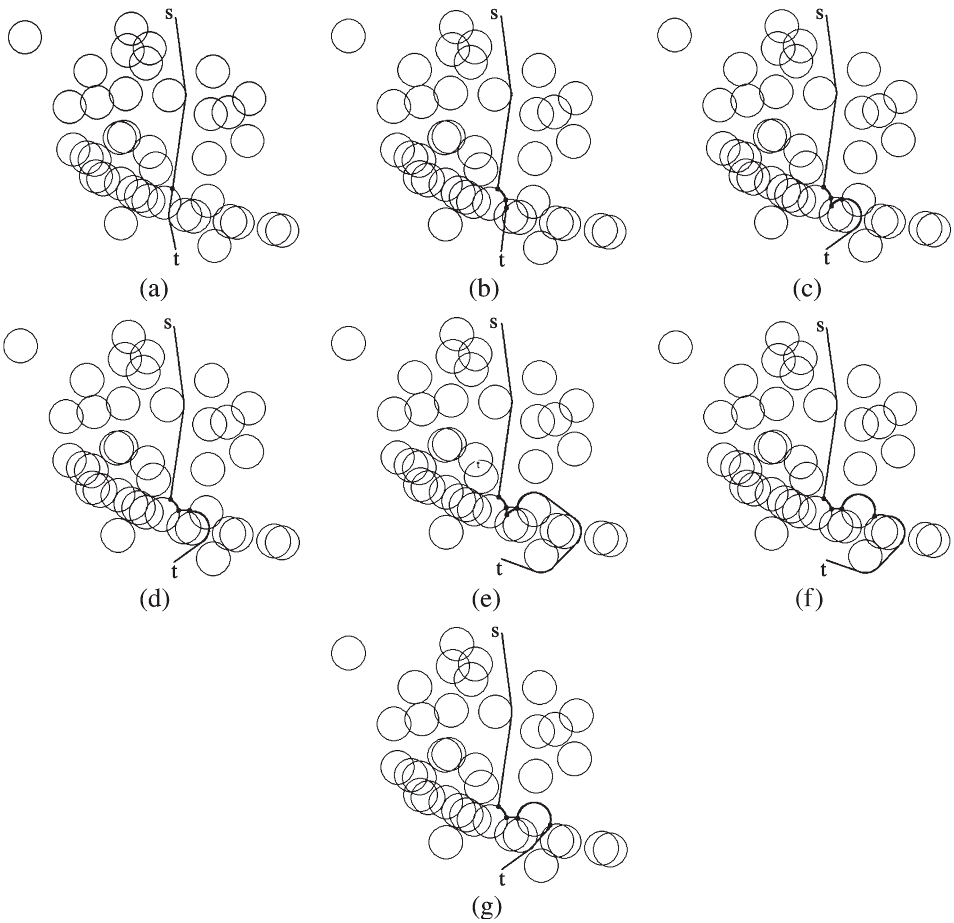


Fig. 6. All (seven) possible realizations of $p_{\mathcal{D}_{100}}$. The probabilities are the respective probabilities of the path realizations. (a) Length $= 707.97$, and prob $= 0.89671$. (b) Length $= 714.90$, and prob $= 0.040105$. (c) Length $= 859.37$, and prob $= 0.038472$. (d) Length $= 831.04$, and prob $= 0.012796$. (e) Length $= 1188.77$, and prob $= 0.0089469$. (f) Length $= 1185.40$, and prob $= 0.0019532$. (g) Length $= 958.43$, and prob $= 0.0010226$.

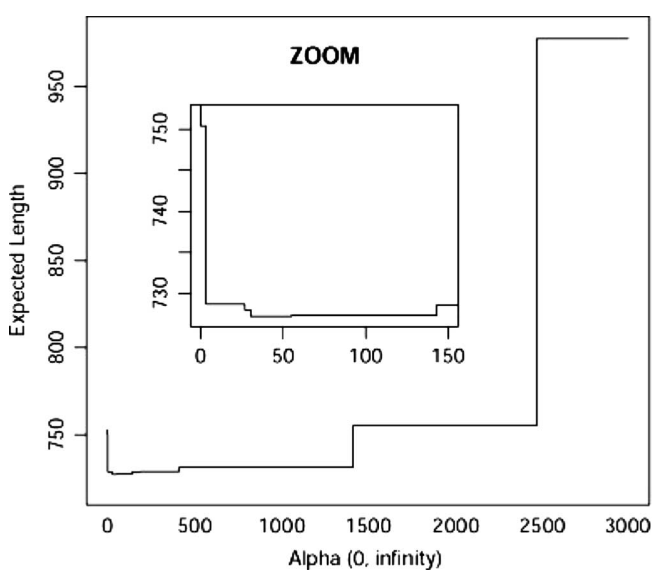


Fig. 7. $E\ell^e p_{\mathcal{D}_\alpha}$ as a function of α for $c = 5$.

$E\ell^e p_{\mathcal{D}_\alpha}$ in the second column of Table II (as linear functions in c), it is clear that, for $c \in (0.0000, 4.1013)$, the fourth indifference interval listed there is best, and you should use any $\alpha \in (26.77, 30.23)$ in practice. If $c \in (4.1013, 15.9145)$ then by similar reasoning you should use any $\alpha \in (30.23, 55.09)$; see the third column of Table II. Furthermore, in a similar fashion, for each possible value of $c \geq 0$, consider such α minimizing $E\ell^e p_{\mathcal{D}_\alpha}$; the values of $\min_{\alpha\geq 0} E\ell^e p_{\mathcal{D}_\alpha}$ are plotted as a (piecewise linear) function of cost c in Fig. 8.

Now, observe that for all $c < 228.1639$, it holds that $\min_{\alpha\geq 0} E\ell^e p_{\mathcal{D}_\alpha} < E\ell^e p_{\mathcal{D}_\infty} = \ell^e q_{s,t,X}$, which means that, for disambiguation costs less than 23.34% of $\ell^e q_{s,t,X} = 977.54$, the optimal simulated risk disambiguation path based on a linear undesirability function yields a strict (expected) improvement over the curve $q_{s,t,X}$, which would be traversed if the disambiguation capability was not available at all and risk was not permitted.

Next, in Section IV, we address the issue of how, in general, to select an optimal or near-optimal value for α.

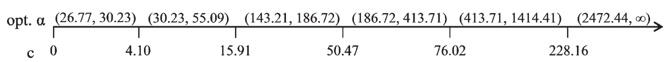
IV. Minimizing $E\ell^e p_{\mathcal{D}_\alpha}$ Over $\alpha \geq 0$

Given s, t, X, ρ, K and assuming that you will use a linear undesirability function in the establishment of a simulated risk disambiguation protocol, you still need to determine the value of the parameter α to use; once the value of α is chosen, then you can efficiently realize $p_{\mathcal{D}_\alpha}$, as described in Section II-C. Thus, what is needed is a practical way to minimize $E\ell^e p_{\mathcal{D}_\alpha}$ over $\alpha \geq 0$, exactly or approximately.

TABLE II
ALL INDIFFERENCE INTERVALS, THE $E\ell^e p_{\mathcal{D}_\alpha}$ FOR VALUES OF α IN THE RESPECTIVE INDIFFERENCE INTERVALS, AND THE RANGE OF DISAMBIGUATION COSTS c WHERE THIS INDIFFERENCE INTERVAL IS OPTIMAL. FOR EXAMPLE, FOR ANY VALUE $0 < c < 4.1013$, THE OPTIMAL VALUE OF α IS ANY $26.77 < \alpha < 30.23$ AND, AS SUCH, $E\ell^e p_{\mathcal{D}_\alpha} = 717.22 + 2.1665\,c$

Indifference interval I	$\mathrm{E}\ell^e p_{\mathcal{D}_\alpha}$ for $\alpha \in I$	Range of c such that $I = \arg\min_{\alpha \geq 0} \mathrm{E}\ell^e p_{\mathcal{D}_\alpha}$
$(0.00, 0.05)$	$735.02 + 3.5587\,c$	$\varnothing$
$(0.05, 3.23)$	$734.89 + 3.1033\,c$	$\varnothing$
$(3.23, 26.77)$	$717.92 + 2.1665\,c$	$\varnothing$
$(26.77, 30.23)$	$717.22 + 2.1665\,c$	$(0.0000, 4.1013)$
$(30.23, 55.09)$	$720.89 + 1.2698\,c$	$(4.1013, 15.9145)$
$(55.09, 143.21)$	$721.14 + 1.2570\,c$	$\varnothing$
$(143.21, 186.72)$	$722.92 + 1.1423\,c$	$(15.9145, 50.4720)$
$(186.72, 413.71)$	$723.07 + 1.1393\,c$	$(50.4720, 76.0167)$
$(413.71, 1414.41)$	$725.81 + 1.1033\,c$	$(76.0167, 228.1639)$
$(1414.41, 2472.44)$	$750.14 + 1.0000\,c$	$\varnothing$
$(2472.44, \infty)$	977.54	$(228.1639, \infty)$

Optimal values of α against the values of c. Note that if $c > 228.1639$ then the optimal $p_{\mathcal{D}_\alpha}$ turns out to be exactly $q_{s,t,X}$.



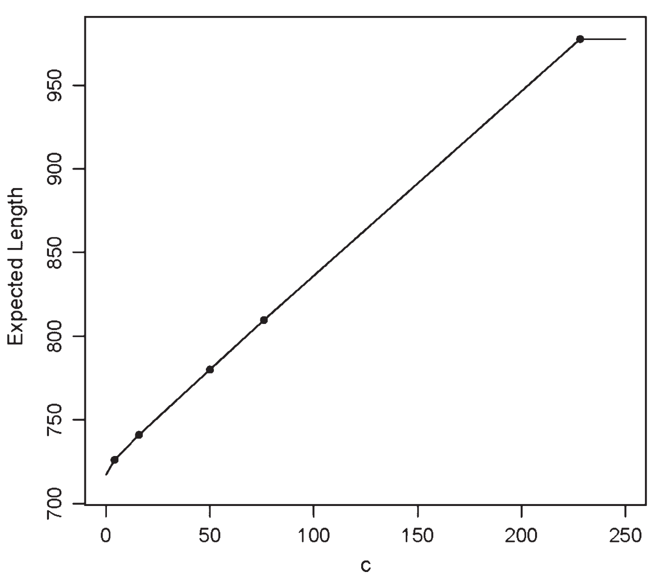


Fig. 8. $E\ell^e p_{\mathcal{D}_\alpha}$ for the optimal α as a function of c.

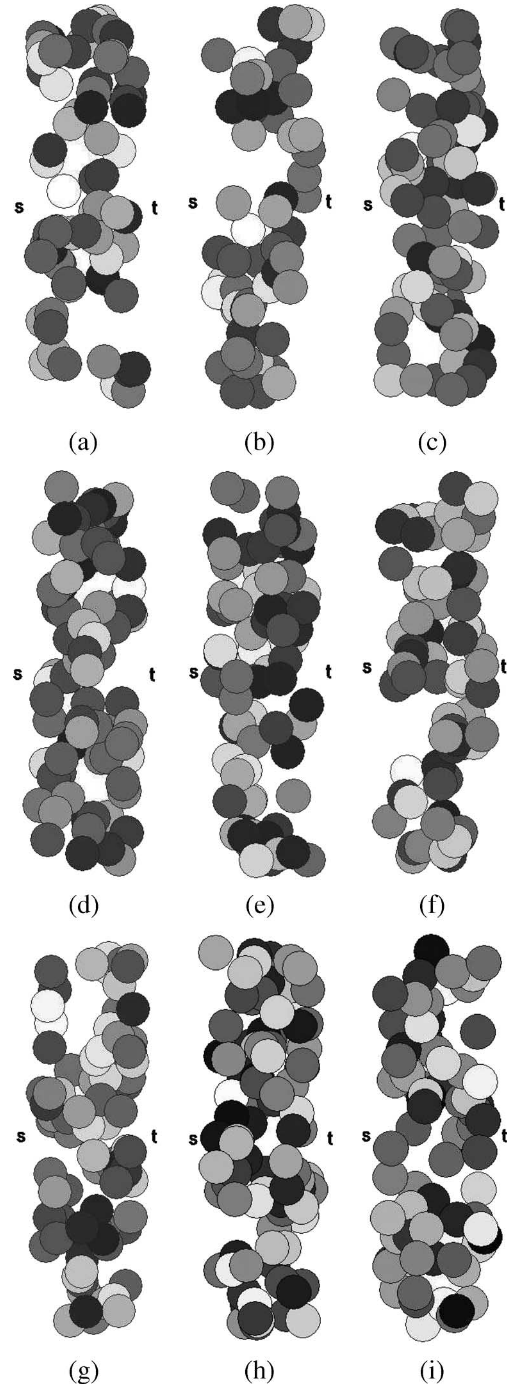


Fig. 9. Nine realizations of a marked point process.

As a first step, suppose it is desired to evaluate $E\ell^e p_{\mathcal{D}_\alpha}$ for just one particular value of α. This may be accomplished by considering all possible outcomes of the disambiguations dictated by $\mathcal{D}_\alpha$ and encountered by $p_{\mathcal{D}_\alpha}$ (which can be done via straightforward recursion) and then weighting the lengths of the possible s, t curves that $p_{\mathcal{D}_\alpha}$ can assume by their respective probabilities. Indeed, we used this procedure to compute $E\ell^e p_{\mathcal{D}_\alpha}$ in Section III.

The number of different s, t curves that can be realized by $p_{\mathcal{D}_\alpha}$ is bounded by 2^K. Note that this is just an upper bound; in the example of Section III, where K was 4, the number of different s, t curves that could be realized by $p_{\mathcal{D}_\alpha}$ for α in the 11 different indifference intervals was 13, 12, 9, 9, 8, 7, 6, 5, 3, 2, and 1, respectively. In particular, since K is fixed, $E\ell^e p_{\mathcal{D}_\alpha}$ can be efficiently evaluated for a particular α; the time required is $O(|X|^3 \log |X|)$, with a multiplicative constant that depends on K. When K is small, it can be practical to evaluate $E\ell^e p_{\mathcal{D}_\alpha}$ for a particular α in this manner.

Therefore, it may be practical to compute $E\ell^e p_{\mathcal{D}_\alpha}$ for a mesh of different values of α and to then adopt the best value α' from the mesh as your parameter, hoping that $E\ell^e p_{\mathcal{D}_{\alpha'}} \approx \min_{\alpha:\alpha \geq 0} E\ell^e p_{\mathcal{D}_\alpha}$.

To illustrate, we obtained 11 realizations of a particular marked spatial point process on $[0, 55] \times [0, 220] \subseteq \mathbf{R}^2$, where the true and false detections are Poisson(20) and Poisson(50), respectively, and the true and false marks are Beta(6, 2) and

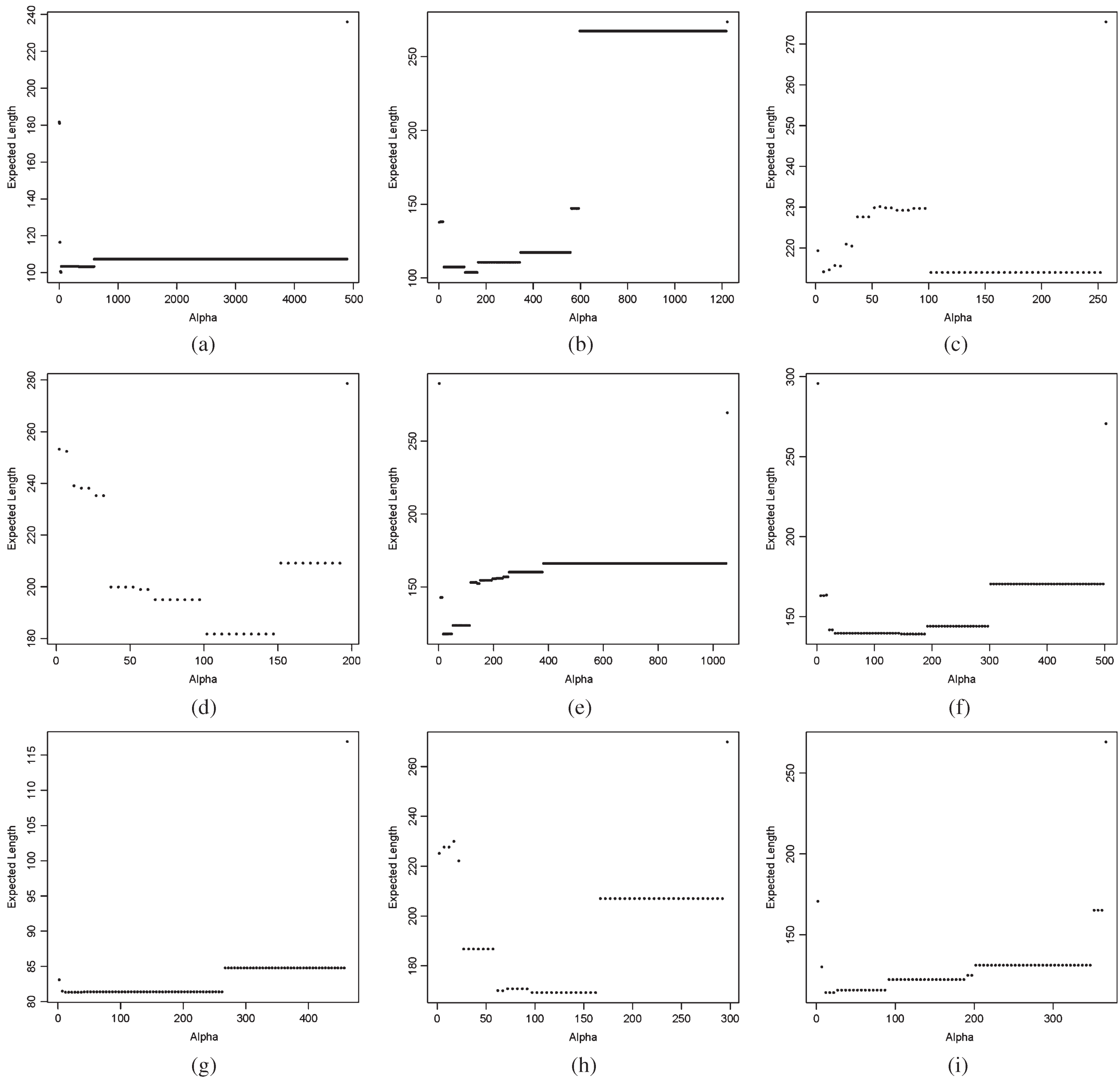


Fig. 10. Plots of $E\ell^e p_{\mathcal{D}_\alpha}$ against $\alpha = 2, 7, 12, 17, \ldots$ for the respective nine realizations in Fig. 9. Note that in each of these plots, for the largest plotted value of α, we have $p_{\mathcal{D}_\alpha} = p_{\mathcal{D}_\infty}$.

Beta(2, 6). We adopted the starting point $s = (-11, 110)$, destination point $t = (66, 110)$, disc radius $r = 10$, disambiguation cost $c = 1$, and the number of available disambiguations $K = 4$. Two of these 11 point process realizations had nearly unobstructed s, t paths and were therefore trivial; the other nine realizations are shown in Fig. 9. The discs in Fig. 9 are gray scaled to reflect the marks of the associated detections; discs are darker and lighter accordingly as the marks are closer to 1 and 0.

In each of these nine nontrivial realizations, we computed $E\ell^e p_{\mathcal{D}_\alpha}$ for $\alpha = 2, 7, 12, 17, \ldots$ until the values of α are large enough so that no disambiguations are performed, i.e., until the values of α are large enough so that $p_{\mathcal{D}_\alpha} = p_{\mathcal{D}_\infty}$. Fig. 10 shows the plots of $E\ell^e p_{\mathcal{D}_\alpha}$ against α for each of these nine realizations. From among the mesh of values $\alpha = 2, 7, 12, 17, \ldots$, we found that the value of α of minimum $E\ell^e p_{\mathcal{D}_\alpha}$ are, for the respective nine realizations, 32, 162, 252, 147, 22, 187, 37, 162, and 22. These respective nine values would be the ones to choose for the parameter α in (linear undesirability function based) simulated risk disambiguation protocols if you encounter these respective nine realizations.

In general, since $E\ell^e p_{\mathcal{D}_\alpha}$ can be efficiently evaluated for any single value of α, the usual numerical optimization methods for real functions of the real line are applicable in minimizing $E\ell^e p_{\mathcal{D}_\alpha}$ over $\alpha \geq 0$.

For large values of K, where an exact evaluation of $E\ell^e p_{\mathcal{D}_\alpha}$ may not be practical (even for a single value of α), Monte Carlo simulations of $p_{\mathcal{D}_\alpha}$ can yield approximate values of $E\ell^e p_{\mathcal{D}_\alpha}$.

V. CONCLUSION

Given a starting point s, destination t, and observations X and ρ, we have illustrated in Section IV how to practically choose a simulated risk disambiguation protocol based on a linear undesirability function to efficiently (Section II-C) traverse a realization of the associated random disambiguation path from s to t. In future work, we intend to seek faster methods of choosing a useful linear undesirability function, perhaps based on easily computed descriptive measures of s, t, X, ρ, K.

It might also be useful to seek nonlinear undesirability functions g that are more effective (i.e., that yield lower $E\ell^e p_{\mathcal{D}_g}$) than linear undesirability functions. However, we would then need a practical way to compute ϕ_g, since only when g is linear can we use Dijkstra's algorithm to find a shortest (in the sense of g) path in $\text{TAG}_{s,t,X,\rho}$. We intend to also consider this direction in future work.

Finally, another, and different, research direction would be to explore possible computational advantages in modeling the underlying problem that utilizes fuzzy numbers in place of probabilities, which is similar to the fuzzy network flow problems in [22].

ACKNOWLEDGMENT

The authors would like to thank the anonymous referees and the Associate Editor for a thorough and thoughtful review of a previous version of this paper. This final version is improved because of these editorial efforts.

REFERENCES

[1] C. E. Priebe, D. E. Fishkind, L. Abrams, and C. D. Piatko, "Random disambiguation paths," *Nav. Res. Logist.*, vol. 52, no. 3, pp. 285–292, 2005.

[2] C. H. Papadimitriou and M. Yannakakis, "Shortest paths without a map," *Theor. Comput. Sci.*, vol. 84, no. 1, pp. 127–150, Jul. 1991.

[3] A. Bar-Noy and B. Schieber, "The Canadian traveller problem," in *Proc. 2nd Annu. ACM-SIAM SODA*, 1991, pp. 261–270.

[4] G. Andreatta and L. Romeo, "Stochastic shortest paths with recourse," *Networks*, vol. 18, no. 3, pp. 193–204, 1988.

[5] G. H. Polychronopoulos and J. N. Tsitsiklis, "Stochastic shortest path problems with recourse," *Networks*, vol. 27, no. 2, pp. 133–143, 1996.

[6] J. S. Provan, "A polynomial-time algorithm to find shortest paths with recourse," *Networks*, vol. 42, no. 2, pp. 115–125, Mar. 2003.

[7] M. Baglietto, G. Battistelli, F. Vitali, and R. Zoppoli, "Shortest path problems on stochastic graphs: A neuro dynamic programming approach," in *Proc. 42nd IEEE Conf. Decision Control*, 2003, pp. 6187–6193.

[8] D. M. Blei and L. P. Kaelbling, "Shortest paths in a dynamic uncertain domain," in *Proc. IJCAI Workshop Adaptive Spatial Representations Dynam. Environ.*, 1991.

[9] D. Ferguson, A. Stentz, and S. Thrun, "PAO* for planning with hidden state," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2004, pp. 2840–2847.

[10] T. Lozano-Perez and M. A. Wesley, "An algorithm for planning collision-free paths among polyhedral obstacles," *Commun. ACM*, vol. 22, pp. 560–570, 1975.

[11] N. S. V. Rao and S. S. Iyengar, "Autonomous robot navigation in unknown terrains: Incidental learning and environmental exploration," *IEEE Trans. Syst., Man, Cybern.*, vol. 20, no. 6, pp. 1443–1449, Nov./Dec. 1990.

[12] N. S. V. Rao, "Robot navigation in unknown generalized polygon terrains using vision sensors," *IEEE Trans. Syst., Man, Cybern.*, vol. 25, no. 6, pp. 947–962, Jun. 1995.

[13] D. L. Smith, "Detection technologies for mines and minelike targets," in *Proc. SPIE, Detection Technol. Mines and Minelike Targets*, Orlando, FL, Apr. 1995, vol. 2496, pp. 404–408.

[14] N. H. Witherspoon, J. H. Holloway, K. S. Davis, R. W. Miller, and A. C. Dubey, "The coastal battlefield reconnaissance and analysis (COBRA) program for minefield detection," in *Proc. SPIE, Detection Technol. Mines and Minelike Targets*, Orlando, FL, Apr. 1995, vol. 2496, pp. 500–508.

[15] Q. A. Holmes, C. R. Schwartz, J. H. Seldin, J. A. Wright, and L. J. Wieter, "Adaptive multispectral CFAR detection of land mines," in *Proc. SPIE, Detection Technol. Mines and Minelike Targets*, Orlando, FL, Apr. 1995, vol. 2496, pp. 421–432.

[16] T. Olson, J. S. Pang, and C. E. Priebe, "A likelihood-MPEC approach to target classification," *Math. Program.*, vol. 96, pp. 1–31, 2002.

[17] C. D. Piatko, C. E. Priebe, L. J. Cowen, I.-J. Wang, and P. McNamee, "Path planning for mine countermeasures command and control," *Proc. SPIE*, vol. 4394, pp. 836–843, 2001.

[18] C. D. Piatko, C. P. Diehl, P. McNamee, C. Resch, and I.-J. Wang, "Stochastic search and graph techniques for MCM path planning," *Proc. SPIE*, vol. 4742, pp. 583–592, 2002.

[19] C. E. Priebe, D. Q. Naiman, and L. Cope, "Importance sampling for spatial scan analysis: Computing scan statistic p-values for marked point processes," *Comput. Stat. Data Anal.*, vol. 35, no. 4, pp. 475–485, Feb. 2001.

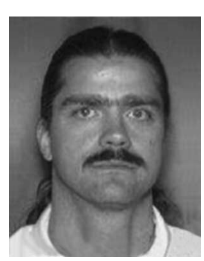
[20] C. E. Priebe, J. S. Pang, and T. Olson, "Optimizing sensor fusion for classification performance," in *Proc. CISST Int. Conf.*, Las Vegas, NV, Jun. 1999, pp. 397–403.

[21] C. E. Priebe, T. E. Olson, and D. M. Healy, Jr., "Exploiting stochastic partitions for minefield detection," *Proc. SPIE*, vol. 3079, pp. 508–518, 1997.

[22] S.-T. Liu and C. Kao, "Network flow problems with fuzzy arc lengths," *IEEE Trans. Syst., Man, Cybern. B, Cybern.*, vol. 34, no. 1, pp. 765–769, Feb. 2004.

Donniell E. Fishkind received the Ph.D. degree from The Johns Hopkins University (JHU), Baltimore, MD, in 1998.

He was with the faculty of the Department of Mathematics and Statistics, University of Southern Maine, Portland, for several years. He has been with the faculty of the Department of Applied Mathematics and Statistics, Whiting School of Engineering, JHU, for the past six years, where he is currently an Associate Research Professor. His research interests are graph theory, combinatorics, probabilistic path planning, matrix analysis, and topological correction in medical imaging.



Carey E. Priebe (M'03) received the B.S. degree in mathematics from Purdue University, West Lafayette, IN, in 1984, the M.S. degree in computer science from San Diego State University, San Diego, CA, in 1988, and the Ph.D. degree in information technology (computational statistics) from George Mason University, Fairfax, VA, in 1993.

From 1985 to 1994, he was a Mathematician and Scientist in the U.S. Navy research and development laboratory system. Since 1994, he has been a Professor with the Department of Applied Mathematics and Statistics, Whiting School of Engineering, The Johns Hopkins University (JHU), Baltimore, MD. At JHU, he holds joint appointments in the Department of Computer Science and the Center for Imaging Science. He is a past President of the Interface Foundation of North America—Computing Science and Statistics, a past Chair of the Section on Statistical Computing of the American Statistical Association, and on the editorial boards of the *Journal of Computational and Graphical Statistics*, *Computational Statistics and Data Analysis*, and *Computational Statistics*. His research interests are in computational statistics, kernel and mixture estimates, statistical pattern recognition, statistical image analysis, and statistical inference for high-dimensional and graph data.

Dr. Priebe is a Fellow of the American Statistical Association.

Kendall E. Giles received the B.S. degree in electrical engineering from the Virginia Polytechnic Institute and State University, Blacksburg, in 1991, the M.S. degree in electrical engineering from Purdue University, West Lafayette, IN, in 1993, the M.S. degree in information systems from Virginia Commonwealth University, Richmond, in 2002, and the M.S. degree in computer science from The Johns Hopkins University, Baltimore, MD, in 2004, where he is currently working toward the Ph.D. degree in computer science.

He has over 15 years of work experience in industry and government environments, most recently as a Systems Engineer at Raytheon, Waltham, MA. His research interests include computer network security, high-dimensional data analysis, and statistical pattern recognition.

Leslie N. Smith received the B.S. and M.S. degrees (*summa cum laude*) in physical chemistry from the University of Connecticut, Storrs, in 1976, and the Ph.D. degree in chemical physics from the University of Illinois, Urbana, in 1979, with his research on quantum mechanical phenomena.

He is currently a Research Physicist with the Naval Research Laboratory, Washington, DC, where he develops novel algorithms for a variety of systems. His research interests include developing improved methodologies and prototypes for infrared systems in signal processing, target detection and tracking, IED detection, estimation, situation assessment, image stabilization, dynamic range compression, and image contrast enhancement.

Vural Aksakalli received the B.S. degree in mathematics and the M.S. degree in industrial engineering and operations research from the North Carolina State University, Raleigh, and the M.S. degree in applied mathematics and statistics from The Johns Hopkins University (JHU), Baltimore, MD, where he is currently working toward the Ph.D. degree in the Department of Applied Mathematics and Statistics, Whiting School of Engineering.

Prior to joining JHU, he was an Operations Research/Business Analyst for a revenue and pricing management company in Atlanta, GA, and an Optimization Software Engineer for a supply chain management company in Rockville, MD. His research interests are in stochastic path planning, Markov decision processes, combinatorial optimization, stochastic optimization, and applied probability and statistics.