Journal of the Operational Research Society (2010), 1–10 © 2010 Operational Research Society Ltd. All rights reserved. 0160-5682/10

www.palgrave-journals.com/jors/

Sensor information monotonicity in disambiguation protocols

X Ye[1], DE Fishkind[1], L Abrams[2] and CE Priebe[1]*

[1]Johns Hopkins University, Baltimore, Maryland, USA; and [2]George Washington University, NW Washington DC, USA

Previous work has considered the problem of swiftly traversing a marked traversal-medium where the marks represent probabilities that associated local regions are traversable, further supposing that the traverser is equipped with a dynamic capability to disambiguate these regions en route. In practice, however, the marks are given by a noisy sensor, and are only *estimates* of the respective probabilities of traversability. In this paper, we investigate the performance of disambiguation protocols that utilize such sensor readings. In particular, we investigate the difference in performance when a disambiguation protocol employs various sensors ranked by their estimation quality. We demonstrate that a superior sensor can yield superior traversal performance—so called *Sensor Information Monotonicity*. In so doing, we provide to the decision-maker the wherewithal to quantitatively assess the advantage of a superior (and presumably more expensive) sensor in light of the associated improvement in performance.

Journal of the Operational Research Society advance online publication, 3 February 2010
doi:10.1057/jors.2009.152

Keywords: probabilistic path planning; dynamic disambiguation

1. Overview

A *general disambiguation problem* is cast in some traversable medium (eg a graph or the plane $\mathbb{R}^2$) with some possibly nontraversable local regions (eg edges in the graph or subsets of $\mathbb{R}^2$). A starting location and ending location are specified, there is a notion of traversal arclength in the medium, and the overall goal is to find a start-end traversal of minimum (expected) arclength. It is further assumed that the traverser begins at the outset with some probabilistic information on the traversability status of all regions and, when in close proximity to the respective regions, the traverser can discover the actual traversability status of the region—at a cost. (Ambiguous regions are not traversed unless and until disambiguation reveals them to be traversable).

An informal example, cast in $\mathbb{R}^2$, is illustrated in Figure 1. The five ambiguous discs $\mathscr{D}_1, \mathscr{D}_2, \mathscr{D}_3, \mathscr{D}_4, \mathscr{D}_5$ are traversable with probabilities 0.9, 0.2, 0.1, 0.6, and 0.3, respectively. Note that if there is not a disambiguation capability then finding the shortest s, t traversal (labelled γ_1 in Figure 1) is simply a deterministic shortest path problem, easily solvable with Dijkstra's algorithm (eg Ahuja *et al* (1993)) applied either to a suitable visibility graph (as in Priebe *et al* (2005)) or to a graph discretization. However, as mentioned, there is a disambiguation capability; how can it best be used? One particular policy for obtaining an s, t traversal is illustrated in Figure 1; it dictates traversing from s to the red icon labelled 1, where $\mathscr{D}_1$ is disambiguated. If $\mathscr{D}_1$ is traversable, then the traversal continues on to the red icon labelled 4, where $\mathscr{D}_5$ is disambiguated. According as $\mathscr{D}_5$ is or is not traversable, the traversal continues to t through $\mathscr{D}_5$ or counterclockwise around $\mathscr{D}_5$. However, if $\mathscr{D}_1$ was not traversable, then the traversal would proceed clockwise around $\mathscr{D}_1$ to the red icon labelled 2, where $\mathscr{D}_2$ is disambiguated. If $\mathscr{D}_2$ is traversable, then the traversal continues through $\mathscr{D}_2$ to the red icon labelled 3, where $\mathscr{D}_4$ is disambiguated. According as $\mathscr{D}_4$ is or is not traversable, the traversal continues to t through $\mathscr{D}_4$ or clockwise around $\mathscr{D}_4$. However, if $\mathscr{D}_2$ was not traversable, then the traversal would proceed to t going around $\mathscr{D}_2$ clockwise. Thus, in particular, the s, t traversal would either be $\gamma_2, \gamma_3, \gamma_4, \gamma_5$, or γ_6 with respective probabilities (0.9)(0.3), (0.9)(0.7), (0.1)(0.2)(0.6), (0.1)(0.2)(0.4), and (0.1)(0.8). Now, if the lengths of these respective traversals were 12, 15, 13, 17, 16, and if the cost of each disambiguation was 1.1, then the expected length of the s, t traversal dictated by this policy would be $(0.9)(0.3)[12 + 2.2] + (0.9)(0.7)[15 + 2.2] + (0.1)(0.2)(0.6)[13 + 3.3] + (0.1)(0.2)(0.4)[17 + 3.3] + (0.1)(0.8)[16 + 2.2]$. This is just an example of one particular policy, and it is not necessarily the best one.

Indeed, the problem of finding an optimal traversal policy in a general disambiguation problem is a stochastic dynamic programming problem (eg Andreatta and Romeo (1988), Blei and Kaelbling (1999)), and many special cases have been studied. Particular special cases of the general disambiguation problem include the *stochastic obstacle scene problem*

Correspondence: CE Priebe, Department of Applied Mathematics and Statistics, Johns Hopkins University, Baltimore, MD 21218-2682, USA.
E-mail: cep@jhu.edu

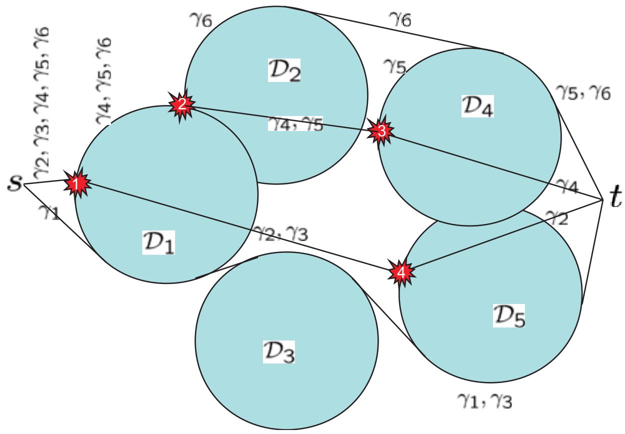

Figure 1 An example of a general disambiguation problem (in $\mathbb{R}^2$), and a policy for it.

(Papadimitriou and Yannakakis (1991)), the *Canadian travellers problem* (Bar-Noy and Schieber (1991), Papadimitriou and Yannakakis (1991)), and the *stochastic shortest paths with recourse problem* (Andreatta and Romeo (1988)). Unfortunately, even small instances of these problems can generate very large state spaces in the stochastic dynamic programming formulation. So, while very special problem structure may admit efficient algorithms that produce optimal solutions (Provan (2003)), there are intractability results for many general cases (Andreatta and Romeo (1988), Bar-Noy and Schieber (1991), Papadimitriou and Yannakakis (1991), Provan (2003)). In general these problems are very difficult to solve exactly. Heuristics for various problems of these types can be found in Aksakalli *et al* (2008), Baglietto *et al* (2003), Blei and Kaelbling (1999), Fishkind *et al* (2007), Polychronopoulos and Tsitsiklis (1996), and Priebe *et al* (2005).

In all of the work cited above, the primary focus had been on the development of useful disambiguation policies, assuming the correctness of the regions' respective probabilities of traversability that are given to the traverser at the outset. In practice, however, it is imperfect sensors that provide the traverser at the outset with marks for respective regions, and these are just *estimates* of the probabilities that the respective regions are traversable. Our focus in this paper is to investigate, for particular disambiguation protocols, how 'better' sensors might yield improved traversals.

In Section 2, we formally describe the disambiguation problem context in which we will work; we formally define protocols, sensors, and introduce *Sensor Information Monotonicity*. We also state and prove several basic theorems. In Section 3, we provide statistical evidence for the strong monotonicity of a particular protocol (called the Reset Protocol) in the context of mine countermeasures path planning. In Section 3.3, we provide further illustration using actual minefield data from the Coastal Battlefield Reconnaissance and Analysis (COBRA) Program for Minefield Detection.

2. Sensor information monotonicity and results

In Section 1, Figure 1 we illustrated an example of a general disambiguation problem, and assumed the correctness of the disc's probability marks. In practice, however, it is imperfect sensors that provide the traverser at the outset with marks for respective regions, and these are just estimates of the probabilities that the respective regions are traversable. To systematically treat this context, we will step back and consider the underlying random process that generates these traversable and untraversable regions; the traverser will not know which are which, and will just perceive these regions as being ambiguous. As part of this random process, random marks are assigned to each region from one of two distributions, according as the region is traversable or not. The traverser will see these marks, and may interpret them as approximations for the probabilities that the respective regions are traversable, conditioned on what the traverser observes of the random process at the outset. Thus, the example in Figure 1 could be an instance of such a random process, as observed by the traverser (and the marks 0.9, 0.2, 0.1, 0.6, and 0.3 are given by sensors). If the traverser follows the policy illustrated in Figure 1 then the associated computation that we performed would provide the traverser with an approximation of the expected length of the s, t traversal, conditioned on what the traverser had observed at the outset. A disambiguation protocol will be defined in Section 2.1 as an assignment of a policy for each possible realization of the random process (as observed by the traverser at the outset). The effectiveness of a protocol can be assessed by the distribution—relative to the probability distribution of the random process—of the length of the s, t traversal obtained by following the policy dictated by the protocol. We will make all of this more formal next in Section 2.1.

2.1. *The formal context*

In this paper we will first focus on a discrete context of the general disambiguation problem. Later, in Section 3, we will treat a continuous context in mine countermeasures path planning by discretizing it to this Section 2.1 context.

Let (V, A) be a graph, with each edge $a \in A$ designated as *deterministic* or *probabilistic*; the set of probabilistic edges will be denoted B. (Each edge may be directed or undirected; if directed then it may only be traversed from tail to head.) For each $a \in A$, let $\ell_a \in \mathbb{R}_{\geqslant 0}$ be the *length* of edge a, and let $\vec{\ell}$ denote the vector with coordinate ℓ_a for each $a \in A$. For each $a \in B$, let $c_a \in \mathbb{R}_{\geqslant 0}$ be the *disambiguation cost* of edge a, and let $\vec{c}$ denote the vector with coordinate c_a for each $a \in B$. Also, for each $a \in B$, let $p_a \in (0, 1)$ be the probability that edge a is traversable, and let $\vec{p}$ denote the vector with coordinate p_a for each $a \in B$; in particular, let the random vector $\vec{X}$ be defined as having an independent Bernoulli(p_a)-distributed coordinate for each $a \in B$, coordinate X_a being 1 or 0 according as a is traversable or not.

For any random variable Θ, let F_Θ denote the cumulative sdistribution function of Θ; also, for any two random variables Θ and Θ', let $\Theta \geqslant_{sto} \Theta'$ denote *stochastic ordering*, meaning that $F_\Theta(\theta) \leqslant F_{\Theta'}(\theta)$ for all $\theta \in \mathbb{R}$. A *sensor* (F_T, F_N) is an ordered pair of (cumulative) distribution functions of interval-(0,1)-valued random variables such that $F_T \geqslant_{sto} F_N$. To every sensor (F_T, F_N) is associated its *sensor random vector* $\vec{\varrho}^{F_T,F_N}$, which is a random vector whose coordinates $\vec{\varrho}_a^{F_T,F_N}$, for all $a \in B$, are each independent random variables with distribution F_T or F_N, according as edge a is traversable or not. The components of $\vec{\varrho}^{F_T,F_N}$ are understood to be (possibly biased) estimators for the probabilities that the respective edges are traversable. If (F_T, F_N) and $(F_{T'}, F_{N'})$ are any two sensors, we say that sensor (F_T, F_N) is *at least as sensitive as* sensor $(F_{T'}, F_{N'})$ if $F_T \geqslant_{sto} F_{T'}$ and $F_{N'} \geqslant_{sto} F_N$. Thus, the partial ordering of sensors by sensitivity allows for meaningful comparison of sensors according to their quality of estimation; greater sensitivity reflects a greater ability of the sensor to discern the traversability of edges. (See the Appendix for more details.)

Consider a tuple $G = (V, A, B, s, t, \vec{\ell}, \vec{c})$ where (V, A) is a graph with probabilistic edges B, a specified *start* vertex $s \in V$, a specified *destination* vertex $t \in V$, arc lengths $\vec{\ell}$, and disambiguation costs $\vec{c}$. A *disambiguation problem setting* (with true traversability probabilities) is a tuple $(G, \vec{p})$ consisting of a G together with the probabilities $\vec{p}$. A *disambiguation problem instance* (with estimated traversability probabilities) is a tuple $(G, \vec{\rho})$ consisting of a G and a realization $\vec{\rho}$ of some sensor random vector $\vec{\varrho}^{F_T,F_N}$, based on some unobserved realization of $\vec{X}$. It will be convenient to always assume the existence of some (perhaps very long) s, t path in $(V, A\backslash B)$. If there are any directed edges in A then we further assume that from each $v \in V$ there is a path to t in $(V, A\backslash B)$.

Given a disambiguation problem instance $(G, \vec{\rho})$, the typical operational goal is to find an s, t walk in (V, A) of traversable edges that is shortest in the sense of minimizing the sum of the edges' lengths together with dynamically incurred disambiguation costs. The disambiguation costs arise as follows: When located at an endpoint of any $a \in B$ (if a is directed, then only at the tail of a) a dynamic traverser has the option to learn whether a is traversable or not, for a cost c_a added to the length of the traversal. A *protocol* Δ is a function that, to each possible disambiguation problem instance where $s \neq t$, assigns an edge $a \in A$ such that s is an endpoint of a (if a is directed then s is the tail of a). The interpretation is that Δ is dictating the next action to take; if $a \in A\backslash B$ then Δ is dictating that a is to be traversed and the other endpoint of a will become the new s, whereas if $a \in B$ then Δ is dictating that a is to be disambiguated, meaning that if a is found to be traversable then a is to be removed from B, else a is to be removed from A. (Note that this disambiguation does not dictate that a is to be traversed.) It is also implicitly understood that this protocol Δ will be called again—and again, iteratively, until t is reached—using the updated disambiguation problem instance.

Thus, to any disambiguation problem instance $(G, \vec{\rho})$ and realization of $\vec{X}$, the protocol Δ specifies one particular s, t-walk, denote it $\gamma_G(\vec{X}, \vec{\rho}, \Delta)$. This s, t-walk may be thought of as random relative to the distribution of $\vec{X}$ (dictated by $\vec{p}$) and the sensor random vector $\vec{\varrho}^{F_T,F_N}$ (from which $\vec{\rho}$ is realized); we denote this s, t-walk-valued random variable $\Gamma_{G,\vec{p}}(\vec{\varrho}^{F_T,F_N}, \Delta)$, where $(G, \vec{p})$ is the disambiguation problem setting. Its length $\ell\Gamma_{G,\vec{p}}(\vec{\varrho}^{F_T,F_N}, \Delta)$ is a real-valued random variable.

Let $(G, \vec{p})$ be any disambiguation problem setting. A protocol Δ is called *strongly monotone for* $(G, \vec{p})$ if, for any two sensors (F_T, F_N) and $(F_{T'}, F_{N'})$ such that sensor (F_T, F_N) is at least as sensitive as sensor $(F_{T'}, F_{N'})$, it holds that $\ell\Gamma_{G,\vec{p}}(\vec{\varrho}^{F_T,F_N}, \Delta) \leqslant_{sto} \ell\Gamma_{G,\vec{p}}(\vec{\varrho}^{F_{T'},F_{N'}}, \Delta)$. We will say a protocol Δ *is weakly monotone for* $(G, \vec{p})$ if, for any sensors (F_T, F_N) and $(F_{T'}, F_{N'})$ such that sensor (F_T, F_N) is at least as sensitive as sensor $(F_{T'}, F_{N'})$,it holds that $E\ell\Gamma_{G,\vec{p}}(\vec{\varrho}^{F_T,F_N}, \Delta) \leqslant E\ell\Gamma_{G,\vec{p}}(\vec{\varrho}^{F_{T'},F_{N'}}, \Delta)$. Of course, by the properties of stochastic order, strong monotonicity of Δ for $(G, \vec{p})$ implies weak monotonicity of Δ for $(G, \vec{p})$. (See the Appendix for more details.)

2.2. *Threshold and penalty protocols*

For any $\alpha \in [0, 1]$, we define the *α-threshold protocol* to be the protocol Δ_α that assigns, to each disambiguation problem instance $(G, \vec{\rho})$, the first edge on a deterministic shortest s, t path in $G\backslash\{a \in B : \rho_a \leqslant \alpha\}$, where 'shortest' is relative to the weights $\ell_a + c_a$ for each edge $a \in A$; we will denote $c_a := 0$ for all $a \in A\backslash B$, even though such edges do not need disambiguation.

A *penalty function* is a function $f : \mathbb{R}_{\geqslant 0} \times \mathbb{R}_{\geqslant 0} \times (0, 1) \to \mathbb{R}$. such that $f(z_1, z_2, z_3)$ is nondecreasing in its first argument, nondecreasing in its second argument, and nonincreasing in its third argument, with limit ∞ as $z_3 \to 0$, and with limit $z_1 + z_2$ as $z_3 \to 1$. The *f-penalty protocol*, denoted Δ_f, is defined as assigning, to each disambiguation problem instance $(G, \vec{\rho})$, the first edge on a deterministic shortest s, t path in G, where 'shortest' is relative to the weights $f(\ell_a, c_a, \rho_a)$ for each edge $a \in A$; we use the convention, for all $a \in A\backslash B$, that $\rho_a := 1$ and $f(\ell_a, 0, 1) := \ell_a$. Of course, for any $\alpha \in (0, 1)$, the α-threshold protocol is the particular *f*-penalty protocol with f defined by $f(z_1, z_2, z_3) := z_1 + z_2 + \delta(z_3, \alpha)$, where $\delta(z_3, \alpha)$ is defined as ∞ or 0, according as $z_3 \leqslant \alpha$ or not.

One important example of a specific penalty function is the function $f(z_1, z_2, z_3) = z_1 + \frac{z_2}{z_3}$; the associated *f*-penalty protocol Δ_f is the *Reset Protocol* from Aksakalli *et al* (2008).

2.3. *Results*

We have proven Sensor Information Monotonicity results for two simple cases:

Theorem 1 *If* $(G, \vec{p})$ *is any disambiguation problem setting such that* $V = \{s, t\}$ *then, for any* $\alpha \in [0, 1]$, Δ_α *is weakly monotone for* $(G, \vec{p})$.

Proof Consider all disambiguation problem settings where $V = \{s, t\}$, and let $\alpha \in [0, 1]$ be given; we show that Δ_α is weakly monotone by induction on $|A|$. If $|A| = 1$ then this sole edge must be deterministic, and the result is trivially true as the traversal under Δ_α is deterministic. Now, assume the weak monotonicity of Δ_α in all cases where $|A|$ is a particular positive integer, and consider an arbitrary disambiguation problem setting $(G, \vec{p})$ such that $|A|$ is 1 larger. Suppose a sensor (F_T, F_N) is at least as sensitive as another sensor $(F_{T'}, F_{N'})$. In particular, we have $F_T(\alpha) \leqslant F_{T'}(\alpha) \leqslant F_{N'}(\alpha) \leqslant F_N(\alpha)$; we want to show $\mathrm{E}\ell\Gamma_{G,\vec{p}}(\vec{\varrho}^{F_T,F_N}, \Delta_\alpha) \leqslant \mathrm{E}\ell\Gamma_{G,\vec{p}}(\vec{\varrho}^{F_{T'},F_{N'}}, \Delta_\alpha)$.

Let $a \in A$ be such that $c_a + \ell_a$ is minimum (over A). If $a \notin B$ then the traversal under Δ_α is deterministic, and monotonicity of Δ_α is trivial, so we need only consider the case $a \in B$. For convenience, we will denote $\gamma := \mathrm{E}\ell\Gamma_{G\backslash\{a\},\vec{p}}(\vec{\varrho}^{F_T,F_N}, \Delta_\alpha)$ and $\gamma' := \mathrm{E}\ell\Gamma_{G\backslash\{a\},\vec{p}}(\vec{\varrho}^{F_{T'},F_{N'}}, \Delta_\alpha)$; by the induction hypothesis we have $\gamma \leqslant \gamma'$, and it is also clear that $\gamma \geqslant c_a + \ell_a$ since every realization of $\ell\Gamma_{G\backslash\{a\},\vec{p}}(\vec{\varrho}^{F_T,F_N}, \Delta_\alpha)$ is at least $c_b + \ell_b$ for the edge $b \in A\backslash\{a\}$ actually traversed in the realization.

We focus first on the traversal using sensor (F_T, F_N). If $\rho_a > \alpha$ then a will be disambiguated first and—if found traversable—a would be immediately traversed (since it would continue to be on the relevant shortest path). However, if $\rho_a \leqslant \alpha$, then a will never be disambiguated. The probabilities of the four events $(X_a = 1, \rho_a > \alpha)$, $(X_a = 0, \rho_a > \alpha)$, $(X_a = 1, \rho_a \leqslant \alpha)$, $(X_a = 0, \rho_a \leqslant \alpha)$ are, respectively, $p_a(1 - F_T(\alpha))$, $(1 - p_a)(1 - F_N(\alpha))$, $p_a(F_T(\alpha))$, $(1 - p_a)(F_N(\alpha))$, and the expected traversal length conditioned on these four events are $c_a + \ell_a, c_a + \gamma, \gamma$, and γ, respectively. Thus $\mathrm{E}\ell\Gamma_{G,\vec{p}}(\vec{\varrho}^{F_T,F_N}, \Delta_\alpha)$ is precisely

$$\begin{aligned} &p_a(1 - F_T(\alpha))(c_a + \ell_a) + (1 - p_a)(1 - F_N(\alpha))(c_a + \gamma) \\ &\quad + p_a(F_T(\alpha))\gamma + (1 - p_a)(F_N(\alpha))\gamma \\ &= p_a(c_a + \ell_a) + p_a(\gamma - c_a - \ell_a)F_T(\alpha) \\ &\quad + (1 - p_a)c_a(1 - F_N(\alpha)) + (1 - p_a)\gamma. \end{aligned}$$

This is less than or equal to

$$\begin{aligned} &p_a(c_a + \ell_a) + p_a(\gamma' - c_a - \ell_a)F_{T'}(\alpha) \\ &\quad + (1 - p_a)c_a(1 - F_{N'}(\alpha)) + (1 - p_a)\gamma' \\ &= p_a(1 - F_{T'}(\alpha))(c_a + \ell_a) + (1 - p_a)(1 - F_{N'}(\alpha)) \\ &\quad \times (c_a + \gamma') + p_a(F_{T'}(\alpha))\gamma' + (1 - p_a)(F_{N'}(\alpha))\gamma', \end{aligned}$$

which is $\mathrm{E}\ell\Gamma_{G,\vec{p}}(\vec{\varrho}^{F_{T'},F_{N'}}, \Delta_\alpha)$, and Theorem 1 is shown. □

Theorem 2 *If $(G, \vec{p})$ is any disambiguation problem setting such that $|B| = 1$ then, for any penalty function f, Δ_f is strongly monotone for $(G, \vec{p})$.*

Proof We first prove that the threshold protocol Δ_α is strongly monotone for any $\alpha \in [0, 1]$. Let $a \in B$ be the probabilistic edge. Consider a shortest s, t path in G among those using the edge a; let θ_1 be the length of this path, denote by v the first vertex of a encountered on this path, denote by $\hat{\theta}_1$ the length of this path from s to v, and denote by $\tilde{\theta}_1$ the length of this path from the other endpoint of a to t. (In particular, $\theta_1 = \hat{\theta}_1 + c_a + \ell_a + \tilde{\theta}_1$.) Let θ_2 be the length of a shortest s, t path in $G\backslash\{a\}$, and let θ_3 be c_a+ the length of a shortest s,t path in $G\backslash\{a\}$ among those using vertex v. Note that $\theta_1 \leqslant \theta_2 \leqslant \theta_3$, or else the result is trivial. Suppose sensor (F_T, F_N) is at least as sensitive as sensor $(F_{T'}, F_{N'})$.

The only values that the random variable $\ell\Gamma_{G,\vec{p}}(\vec{\varrho}^{F_T,F_N}, \Delta_\alpha)$ can assume are θ_1, θ_2, and θ_3 and, with sensor (F_T, F_N) and protocol Δ_α, the respective probabilities of these three values are $p_a(1 - F_T(\alpha))$, $p_a F_T(\alpha) + (1 - p_a)F_N(\alpha)$, and $(1 - p_a)(1 - F_N(\alpha))$. Thus, for all $u \in \mathbb{R}$,

$$F_{\ell\Gamma_{G,\vec{p}}(\vec{\varrho}^{F_T,F_N},\Delta_\alpha)}(u) = \begin{cases} 0 & \text{if } u < \theta_1 \\ p_a(1 - F_T(\alpha)) & \text{if } \theta_1 \leqslant u < \theta_2 \\ p_a + (1 - p_a)F_N(\alpha) & \text{if } \theta_2 \leqslant u < \theta_3 \\ 1 & \text{if } \theta_3 \leqslant u. \end{cases}$$

Together with the fact that $F_T(\alpha) \leqslant F_{T'}(\alpha)$ and $F_{N'}(\alpha) \leqslant F_N(\alpha)$, it is now clear that for all $u \in \mathbb{R}$, $F_{\ell\Gamma_{G,\vec{p}}(\vec{\varrho}^{F_T,F_N},\Delta_\alpha)}(u) \geqslant F_{\ell\Gamma_{G,\vec{p}}(\vec{\varrho}^{F_{T'},F_{N'}},\Delta_\alpha)}(u)$, and strong monotonicity of Δ_α is thus shown.

If f is any penalty function then a would be disambiguated by Δ_f—when using sensor (F_T, F_N)—if and only if $\hat{\theta}_1 + f(\ell_a, c_a, \rho_a) + \tilde{\theta}_1 < \theta_2$. Define $\alpha^* := \inf\{\alpha \in [0, 1] : \hat{\theta}_1 + f(\ell_a, c_a, \alpha) + \tilde{\theta}_1 < \theta_2\}$. Since f is nonincreasing in its third argument we have that, for $\rho > \alpha^*$, Δ_f will have a disambiguated and, for $\rho < \alpha^*$, Δ_f will not have a disambiguated. The case $\rho = \alpha^*$ is trivial. Thus Δ_f acts here precisely as the threshold protocol Δ_{α^*}, and so Δ_f is strongly monotone by the strong monotonicity of threshold protocols. □

Sensor Information Monotonicity for more general disambiguation problem settings, and an ability to determine how much improvement is provided by one sensor over another, will result in a tool of significant value to the decision-maker. Although penalty protocols (which are a more general class) are harder to analyse than threshold protocols (which are special cases of penalty protocols), we have examples of disambiguation problem settings for which a penalty protocol is strictly superior to all threshold protocols. In particular, the Reset protocol introduced in Aksakalli *et al* (2008) is shown there to be an optimal policy in a general disambiguation problem setting for graphs where $V = \{s, t\}$ and the true marks p are available, yet the Reset protocol is not in general a threshold protocol in that context.

3. Computational investigation

In this section we present simulation and experimental results from applying a particular penalty protocol to disambiguation problems in minefield scenarios (see for example Washburn (1999) for more discussion on these kinds of minefield scenarios). In particular, we are looking here for evidence of strong and weak monotonicity. This particular penalty protocol, called the *Reset Protocol*, was introduced in Aksakalli *et al* (2008) as a useful and practical heuristic

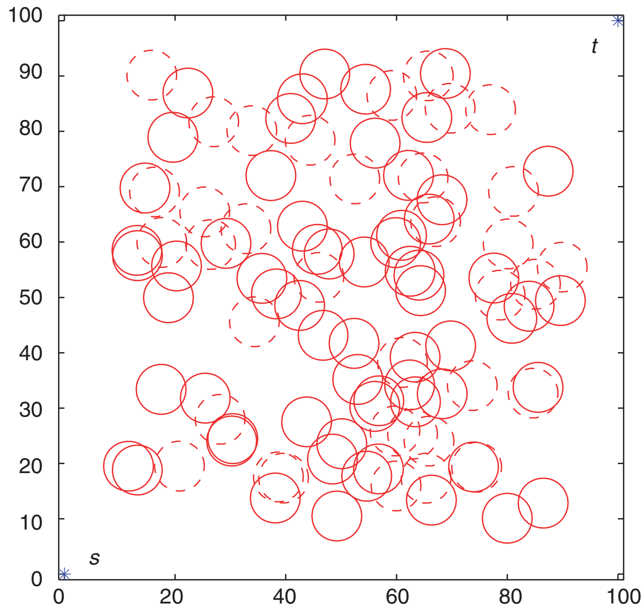

Figure 2 A realization of $\mathscr{D}_N$ and $\mathscr{D}_T$, generated by the process of Section 3.1. Disks of $\mathscr{D}_N$ are depicted with solid lines, disks of $\mathscr{D}_T$ with dashed lines.

for general disambiguation problems. It was also shown in Aksakalli *et al* (2008) that the Reset protocol provides an optimal policy in a general disambiguation problem setting for graphs where $V = \{s, t\}$ and the true marks p are available.

3.1. Simulation setting

We model a minefield as a set $\mathscr{D}$ of (possibly intersecting) open disks in $[0, 100] \times [0, 100] \subseteq \mathbb{R}^2$. Each disk is either a *true detection* (representing a nontraversable hazard) or *false detection* (which can be traversed); the set of true and false detections are respectively denoted $\mathscr{D}_N$ and $\mathscr{D}_T$. The graph (V, A) is a directed graph with vertex set $V := \mathbb{Z}^2 \cap ([0, 100] \times [0, 100])$ (ie, a subset of the two-dimensional integer lattice) and edge set A defined by the eight-adjacency relation; that is, for all $v, w \in V$ distinct, we have that vw is (a directed edge) in A precisely when, in both plane coordinates, v differs from w by 1, −1, or 0. (Of course, if $vw \in A$ then $wv \in A$.) We use $s = (0, 0)$, $t = (100, 100)$.

In each experiment, we use $|\mathscr{D}| = 100$; the number of members of $\mathscr{D}$ that are chosen to be true detections is a realization of a Binomial(100, 0.6) random variable, the centre of each disk is independently and uniform-randomly selected from $[10, 90] \times [10, 90]$, and the disks' radii are all 4.5. A realization of this process is shown in Figure 2. Each detection $D \in \mathscr{D}$, according as it is a true or false detection, receives a mark ρ_D realized from the distribution Beta$(3.5 - \lambda, 3.5 + \lambda)$ or Beta$(3.5 + \lambda, 3.5 - \lambda)$, respectively, for a parameter $\lambda \in [0, 3.5]$ that we preselect; ρ_D is to be interpreted as an estimate of the probability that D is traversable. Note that, in particular, if $\lambda = 0$ then the marks contain no information on whether their corresponding disks are true or false detections and, at the other extreme, if $\lambda = 3.5$ then the marks provide such information perfectly. In particular, for any $\lambda > \lambda'$, the sensor based on λ is at least as sensitive as the sensor based on λ'.

For each edge $vw \in A$, the length ℓ_{vw} is defined to be the Euclidean length of vw, which here is always 1 or $\sqrt{2}$. Note that edges (are directed and) are considered probabilistic only if they enter an ambiguous disk, since edges that exit from or are internal to an ambiguous disk could not anyway be used until the disk was entered, and at the time of entry the disk would be disambiguated and traversable. For each edge $vw \in A$, and each $D \in \mathscr{D}$, we define $\rho_{vw,D}$ to be ρ_D if v is not in D but vw intersects D, and we define $\rho_{vw,D}$ to be 1 otherwise. Then, for each edge $vw \in A$, we define $\rho_{uv} := \prod_{D \in \mathscr{D}} \rho_{vw,D}$; indeed, vw is a probabilistic edge if and only if $\rho_{vw} < 1$, and the assignment of ρ_{vw} to each probabilistic edge vw plays the role of a realization of the sensor random vector, since it is an estimate of the probability that the edge is traversable (based on the ρ_D for the relevant D). For each edge $vw \in A$, we use the disambiguation cost $c_{vw} := (2.25) \cdot |\{D : \rho_{vw}, D < 1\}|$, since disambiguating an edge will consist of disambiguating all disks the edge enters, at a fixed cost per disk.

We use the specific penalty function $f(z_1, z_2, z_3) = z_1 + \frac{z_2}{z_3}$; thus the shortest path problem implicit in Δ_f is relative to the weights $\ell_{vw} + \frac{c_{vw}}{\rho_{vw}}$ for each edge $vw \in A$. This protocol is precisely the *Reset Protocol* from Aksakalli *et al* (2008). The one modification to the utilization of this protocol Δ_f is as follows: If this protocol Δ_f returns probabilistic edge vw, then the disambiguation performed is of each detection D such that $\rho_{vw,D} < 1$; for each such D, if D is now identified as a false detection then $\rho_{a,D}$ is set to 1 for all $a \in A$, and if D is now identified as a true detection then all edges a such that previously $\rho_{a,D} < 1$ are removed from A.

3.2. Simulation results

In Section 3.1, we described a problem setting, a particular class of sensors parameterized by λ, and a particular penalty disambiguation protocol called the Reset protocol. Applying the Reset Protocol to a random instance of the Section 3.1 setting yields a random s, t walk denoted by $\Gamma(\lambda)$ with (random) length $\ell\Gamma(\lambda)$. When we condition on a particular $\mathscr{D}_N$ and $\mathscr{D}_T$ occurring, the random s, t walk will be denoted $\Gamma_{\mathscr{D}_N,\mathscr{D}_T}(\lambda)$.

Our goal in this section is to provide statistical evidence that for all λ, λ' such that $\lambda < \lambda'$ it holds that $\ell\Gamma(\lambda) \geqslant_{sto} \ell\Gamma(\lambda')$; this is a statement of strong monotonicity, that is, that the Reset Protocol is strongly monotone in the scenario of Section 3.1, restricted to the particular class of sensors described in Section 3.1. We also provide statistical evidence that in general, when conditioning on particular $\mathscr{D}_N$ and $\mathscr{D}_T$ occurring, it holds that $\ell\Gamma_{\mathscr{D}_N,\mathscr{D}_T}(\lambda) \geqslant_{sto} \ell\Gamma_{\mathscr{D}_N,\mathscr{D}_T}(\lambda')$ whenever $\lambda < \lambda'$, which is strong monotonicity when conditioning

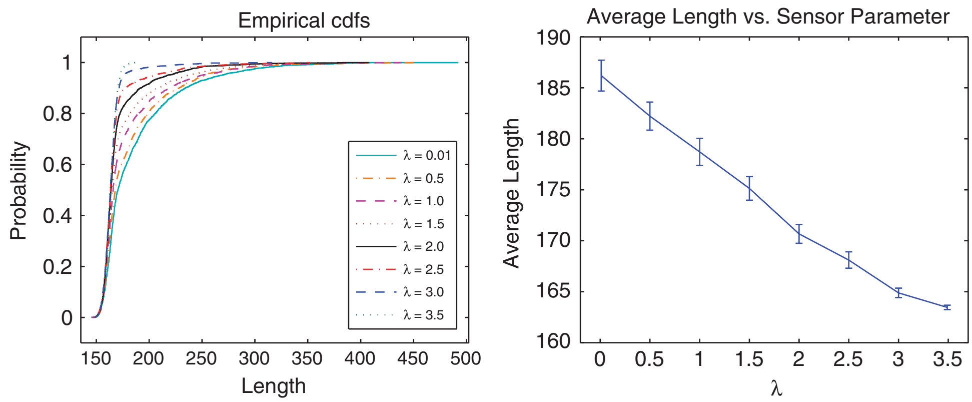

Figure 3 For each of $j = 1, 2, \ldots, 8$, we obtained 2500 realizations of $\ell\Gamma(\lambda_j)$; in the left panel, we present the respective empirical cdfs $\hat{F}_{\ell\Gamma(\lambda_j)}$ for each λ_j, and in the right panel we present the respective sample means and 95% confidence intervals for $\mathrm{E}\ell\Gamma(\lambda_j)$ plotted against λ_j.

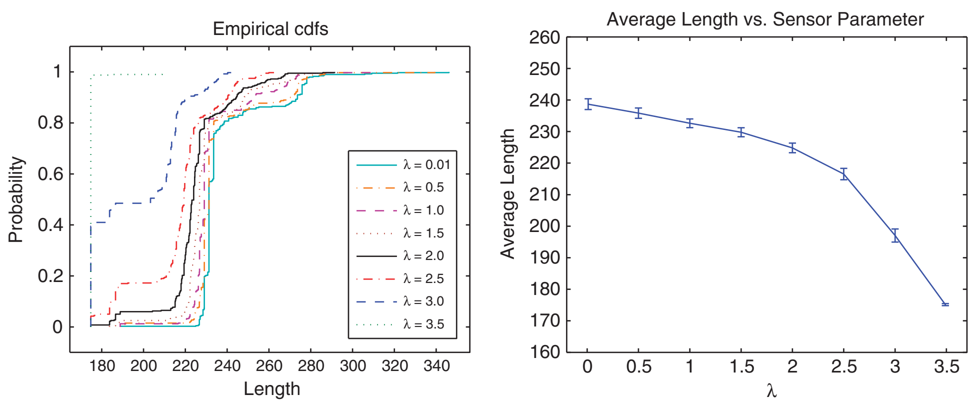

Figure 4 For the particular $\mathscr{D}_N$ and $\mathscr{D}_T$ visualized in Figure 2, we obtained 400 realizations of $\ell\Gamma_{\mathscr{D}_N,\mathscr{D}_T}(\lambda_j)$ for each of j=1, 2, ..., 8; in the left panel we present the respective empirical cdfs $\hat{F}_{\ell\Gamma\mathscr{D}_N,\mathscr{D}_T(\lambda_j)}$ for each λ_j, and in the right panel we present the respective sample means and 95% confidence intervals for $\mathrm{E}\ell\Gamma_{\mathscr{D}_N,\mathscr{D}_T(\lambda_j)}$ plotted against λ_j.

on $\mathscr{D}_N$ and $\mathscr{D}_T$. Note that if strong monotonicity holds whenever conditioning on any particular $\mathscr{D}_N$ and $\mathscr{D}_T$ then, indeed, strong monotonicity holds unconditionally. Finally, even though strong monotonicity implies weak monotonicity, we separately provide statistical evidence of weak monotonicity as the magnitude of the weak monotonicity is of interest.

This monotonicity is illustrated via Monte Carlo simulations. For each of the values of λ from among λ_1:=0.01, λ_2:=0.5, λ_3:=1.0, λ_4:=1.5, λ_5:=2.0, λ_6:=2.5, λ_7:=3.0, λ_8:=3.49 we obtained 2500 realizations of $\ell\Gamma(\lambda_j)$. For each of these λ_j, the respective empirical cumulative distribution function based on the 2500 realizations of $\ell\Gamma(\lambda_j)$ is shown in Figure 3; these empirical cdfs will be denoted $\hat{F}_{\ell\Gamma(\lambda_j)}$. By inspection it is seen that, for all j, $\hat{F}_{\ell\Gamma(\lambda_j)} \leqslant \hat{F}_{\ell\Gamma(\lambda_{j+1})}$, which indicates that $\ell\Gamma(\lambda_j) \geqslant_{sto} \ell\Gamma(\lambda_{j+1})$. To quantify this, we performed the Kolmogorov–Smirnoff test for $H_0 : \ell\Gamma(\lambda_j) {=}_{sto} \ell\Gamma(\lambda_{j+1})$ *versus* $H_a : \ell\Gamma[(\lambda_j) {>}_{sto} \ell\Gamma(\lambda_{j+1})$ for each of $j = 1, 2, \ldots, 7$; the respective p-values are on the order of 10^{-5}—except for $j = 6, 7$, for which the p-values were 0.0002 and 0.0383—which convincingly suggests strong monotonicity here. In Figure 3, we also plot, for $j = 1, 2, \ldots, 8$, the respective sample means (for each sample of 2500 realizations)—including 95% confidence intervals for $\mathrm{E}\ell\Gamma(\lambda_j)$—against λ_j; weak monotonicity is clearly suggested.

Now, conditioning on the particular $\mathscr{D}_N$ and $\mathscr{D}_T$ visualized in Figure 2, we realized 400 values of $\ell\Gamma_{\mathscr{D}_N,\mathscr{D}_T}(\lambda_j)$ for each of $j = 1, 2, \ldots, 8$; the respective empirical cdfs $\hat{F}_{\ell\Gamma_{\mathscr{D}_N,\mathscr{D}_T}(\lambda_j)}$ are depicted in Figure 4. The Kolmogorov—Smirnoff test for $H_0 : \ell\Gamma_{\mathscr{D}_N,\mathscr{D}_T}(\lambda_j) {=}_{sto} \ell\Gamma_{\mathscr{D}_N,\mathscr{D}_T}(\lambda_{j+1})$ *versus* the alternative $H_a : \ell\Gamma_{\mathscr{D}_N,\mathscr{D}_T}(\lambda_j) {>}_{sto} \ell\Gamma_{\mathscr{D}_N,\mathscr{D}_T}(\lambda_{j+1})$, for each of $j = 1, 2, \ldots, 7$, yields a p-value on the order of 10^{-10}, which convincingly suggests strong monotonicity when

Table 1 We randomly (in the sense of Section 3.1) selected $(\mathscr{D}_N^1, \mathscr{D}_T^1)$, $(\mathscr{D}_N^2, \mathscr{D}_T^2)$, $\ldots$, $(\mathscr{D}_N^{50}, \mathscr{D}_T^{50})$, and for each we realized 100 values of $\ell\Gamma_{\mathscr{D}_N^i,\mathscr{D}_T^i}(\lambda_j)$ for each of $j = 3, 6$. We used the empirical cdfs to perform the Kolmogorov–Smirnoff test for $H_0 : \ell\Gamma_{\mathscr{D}_N^i,\mathscr{D}_T^i}(\lambda_3) =_{sto} \ell\Gamma_{\mathscr{D}_N^i,\mathscr{D}_T^i}(\lambda_6)$ *versus* left-, right-, and two-tailed alternatives; the p-values are listed in the left side of this table. We also performed the t-test for $H_0 : \mathrm{E}\ell\Gamma_{\mathscr{D}_N^i,\mathscr{D}_T^i}(\lambda_3) = \mathrm{E}\ell\Gamma_{\mathscr{D}_N^i,\mathscr{D}_T^i}(\lambda_6)$ *versus* left-, right-, and two-tailed alternatives; the p-values are listed in the right side of this table

	KS-test for $H_0 : \ell\Gamma_{\mathscr{D}_N^i,\mathscr{D}_T^i}(\lambda_3) =_{sto} \ell\Gamma_{\mathscr{D}_N^i,\mathscr{D}_T^i}(\lambda_6)$ *vs*			*t-test for* $H_0 : \mathrm{E}\ell\Gamma_{\mathscr{D}_N^i,\mathscr{D}_T^i}(\lambda_3) = \mathrm{E}\ell\Gamma_{\mathscr{D}_N^i,\mathscr{D}_T^i}(\lambda_6)$ *vs*		
i of $(\mathscr{D}_N^i, \mathscr{D}_T^i)$	$H_a : \neq_{sto}$	$H_a : >_{sto}$	$H_a : <_{sto}$	$H_a : \neq$	$H_a : >$	$H_a : <$
1	0	0	1	0	0	1
2	0	0	1	1.54E−05	7.70E−06	1
3	0.9921	0.688	1	0.0991	0.0495	0.9505
4	2.85E−06	1.42E−06	1	1.15E−07	5.73E−08	1
5	3.70E−09	1.85E−09	1	7.41E−10	3.70E−10	1
6	1	0.9897	1	0.3197	0.1599	0.8401
7	0.0205	0.0102	1	0.0528	0.0264	0.9736
8	0	0	0.9593	0	0	1
9	1	0.9593	1	0.1583	0.0792	0.9208
10	0.0994	0.0497	1	0.0255	0.0128	0.9872
11	0	0	1	0	0	1
12	0	0	1	0	0	1
13	0	0	1	8.78E−09	4.39E−09	1
14	0	0	1	0	0	1
15	0	0	1	0	0	1
16	0	0	1	0	0	1
17	0	0	1	0	0	1
18	0	0	1	0	0	1
19	0	0	1	0	0	1
20	0	0	1	0	0	1
21	0	0	1	0	0	1
22	0.9921	0.688	1	0.0136	0.0068	0.9932
23	0.003	0.0015	1	2.26E−07	1.13E−07	1
24	0	0	1	0	0	1
25	0.8938	0.5144	1	0.0042	0.0021	0.998
26	0	0	1	0	0	1
27	1.47E−09	7.33E−10	0.9897	1.36E−10	0	1
28	0	0	1	0	0	1
29	1	0.8469	1	0.0459	0.023	0.977
30	0	0	1	0	0	1
31	5.22E−08	2.61E−08	1	1.95E−07	9.77E−08	1
32	0	0	1	0	0	1
33	0	0	1	0	0	1
34	0	0	1	0	0	1
35	0	0	1	0	0	1
36	0	0	1	0	0	1
37	0	0	1	0	0	1
38	0.4431	0.2241	1	0.0004	0.0002	0.9998
39	0	0	1	0	0	1
40	0	0	1	3.29E−09	1.65E−09	1
41	0.4431	0.2241	1	0.0127	0.0063	0.9937
42	9.12E−09	4.56E−09	1	7.18E−08	3.59E−08	1
43	0	0	1	0	0	1
44	9.12E−09	4.56E−09	1	4.54E−08	2.27E−08	1
45	0	0	1	0	0	1
46	0	0	1	0	0	1
47	0.14	0.07	1	0.0002	7.69E−05	0.9999
48	0	0	1	0	0	1
49	0.0314	0.0157	1	2.76E−05	1.38E−05	1
50	0	0	1	0	0	1

Table 2 Coordinates and marks for the 39 COBRA detections. The 12 true mines are in bold

1st coordinates	*2nd coordinates*	ρ	*1st coordinates*	*2nd coordinates*	ρ	*1st coordinates*	*2nd coordinates*	ρ
321.17	158.27	0.40983	54.23	201.12	0.45822	158.17	516.48	0.56475
215.13	**428.31**	**0.38110**	**−145.67**	**703.06**	**0.38286**	**−151.01**	**572.15**	**0.43924**
221.12	**557.31**	**0.35953**	−166.36	299.42	0.50827	296.16	163.31	0.88351
163.31	186.14	0.34364	28.31	205.03	0.84731	−79.26	709.99	0.43915
100.40	**376.47**	**0.48513**	−105.75	262.20	0.74252	185.31	182.18	0.34734
116.39	110.84	0.55876	−128.60	274.12	0.37999	−61.19	345.12	0.82817
−91.27	664.45	0.83325	−82.87	248.29	0.41692	**105.47**	**509.80**	**0.14853**
−19.93	568.04	0.40063	−310.23	402.92	0.34572	−320.73	532.23	0.66908
−35.11	242.61	0.89670	−169.99	438.90	0.35837	**95.39**	**248.12**	**0.81132**
−78.75	396.14	0.92690	−245.28	372.05	0.47846	**−166.45**	**180.33**	**0.38918**
−134.53	769.27	0.80614	**−258.45**	**641.03**	**0.34330**	**111.60**	**640.10**	**0.43471**
−219.32	313.68	0.42551	−455.72	742.57	0.36013	**−157.10**	**441.96**	**0.35556**
−242.22	321.51	0.34345	−237.86	546.19	0.86207	**−269.98**	**379.65**	**0.47198**

conditioning on this $\mathscr{D}_N$ and $\mathscr{D}_T$. We also plotted in Figure 4, for $j = 1, 2, \ldots, 8$, the respective sample means (for each sample of 400 realizations)—including 95% confidence intervals for $E\ell\Gamma_{\mathscr{D}_N,\mathscr{D}_T}(\lambda_j)$—plotted against λ_j; weak monotonicity is clearly suggested here. But, is this typical of general $\mathscr{D}_N$ and $\mathscr{D}_T$ that arise from the process of Section 3.1? To address this, we randomly selected $(\mathscr{D}_N^1, \mathscr{D}_T^1)$, $(\mathscr{D}_N^2, \mathscr{D}_T^2)$, $\ldots$, $(\mathscr{D}_N^{50}, \mathscr{D}_T^{50})$ in the manner described in Section 3.1. For each $i = 1, 2, \ldots, 50$ we realized 100 values of $\ell\Gamma_{D_N^i,D_T^i}(\lambda_j)$ for each of $j = 3, 6$. We used the empirical cdfs to conduct the Kolmogorov–Smirnoff test for $H_0 : \ell\Gamma_{\mathscr{D}_N^i,\mathscr{D}_T^i}(\lambda_3) =_{sto} \ell\Gamma_{\mathscr{D}_N^i,\mathscr{D}_T^i}(\lambda_6)$ *versus* left-, right-, and two-tailed alternatives; the p-values are in Table 1. We also performed t-tests for $H_0 : \mathrm{E}\ell\Gamma_{\mathscr{D}_N^i,\mathscr{D}_T^i}(\lambda_3) = \mathrm{E}\ell\Gamma_{\mathscr{D}_N^i,\mathscr{D}_T^i}(\lambda_6)$ *versus* left-, right-, and two-tailed alternatives; the p-values are also in Table 1. Indeed, these results suggest that weak and strong monotonicity hold when conditioning on general $(\mathscr{D}_N, \mathscr{D}_T)$.

3.3. *COBRA data*

Table 2 contains the two-dimensional Euclidean coordinates of an actual pattern of potential-mine detections from the Coastal Battlefield Reconnaissance and Analysis (COBRA) Program for Minefield Detection (Witherspoon *et al* (1995)); this data set appears and is analysed in Smith (1995), Aksakalli *et al* (2008), Fishkind *et al* (2007), Olson *et al* (2002), Priebe *et al* (1997), and Priebe *et al* (2005). The column of Table 2 with the ρ's gives the marks as rendered by the classification rule in Olson *et al* (2002), (see also Priebe *et al* (1999), Piatko *et al* (2001), Priebe *et al* (2001), Piatko *et al* (2002)); they are estimates of the probabilities that the respective detections are not mines, that is, are traversable. We know which of these are actual mines and which are false detections—see Table 2—but the protocol does not make use of this 'truth' information in this experiment. We create a disk of radius 50 for each detection (centred at the respective two-dimensional coordinates in Table 2); denote the disks $D_1, D_2, \ldots, D_{39}$ (they are illustrated in Figure 5), and denote their respective marks $\rho_1, \rho_2, \ldots, \rho_{39}$. We still use the integer lattice $\mathbb{Z}^2$ as our underlying graph in the manner described in Section 3.2, and we use $s = (-300, 250)$ for the start vertex and $t = (300, 600)$ for the destination vertex. The cost of disambiguating a detection is taken as $c = 50$.

For each λ such that $0 \leqslant \lambda \leqslant 1$ and for each $i = 1, 2, \ldots, 39$, we define $\rho_{i,\lambda}$ to be $\lambda \cdot 1 + (1-\lambda) \cdot \rho_i$ if D_i is traversable, and $\lambda \cdot 0 + (1-\lambda) \cdot \rho_i$ if D_i is not traversable. Of course, as λ goes from 0 to 1 the marks $\rho_{i,\lambda}$, for all $i = 1, 2, \ldots, 39$, better reflect actual respective traversabilities; when $\lambda = 0$ these marks are as given in Table 2, and when $\lambda = 1$ the marks definitively indicate respective traversabilities. In the last box of Figure 5 we plot—against the respective values for the sensor improvement parameter λ of $0.0, 0.1, 0.2, \ldots, 1.0$—the length of the path taken under the Reset protocol based on the marks $\rho_{i,\lambda}$; the other three boxes in Figure 5 illustrate the actual traversals taken under each of $\lambda = 0.0, 0.4, 0.5$. Note the improvement in traversal length as the sensor parameter λ goes from 0 to 1, and indeed the improvement is monotone—up to the limit of the resolution used here.

4. Discussion

We consider the problem of swiftly traversing a marked traversal-medium where the marks represent sensor estimates of the probabilities that associated local regions are traversable, further supposing that the traverser is equipped with a dynamic capability to disambiguate these regions en route. In this disambiguation problem setting, a superior sensor ought to yield superior traversal performance; any protocol that does not have this Sensor Information Monotonicity property is suspect. Furthermore, for any given protocol, the degree of superiority in traversal performance obtained through the use of one sensor *versus* another is an essential quantity in any cost/benefit analyses considering whether the superior (and presumably more expensive) sensor is worth the additional cost.

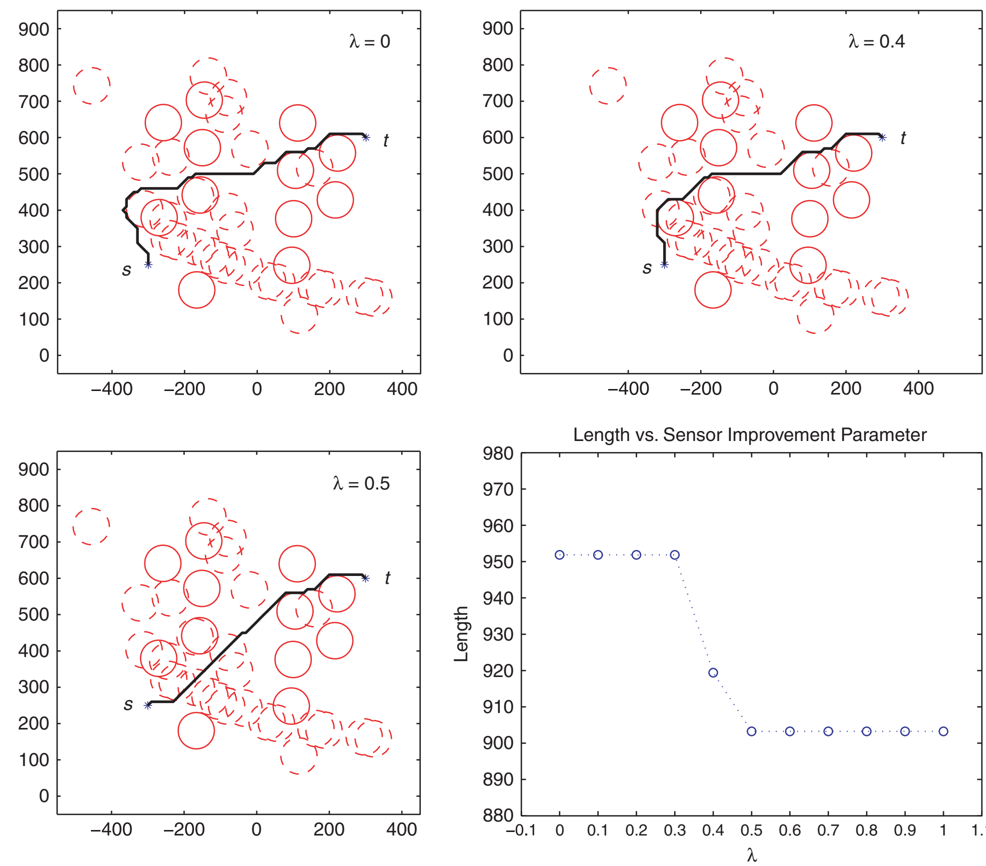

Figure 5 The first three boxes show the COBRA data (actual mines in solid line, false detections in dashed line); these boxes contain traversals for different values of the sensor parameter λ. The last box plots the length of the traversals against sensor improvement parameter values $\lambda = 0.0, 0.1, 0.2, \ldots, 1.0$.

We have demonstrated Sensor Information Monotonicity properties for some classes of protocols through theory, simulation, and experiment. These results lay the groundwork for the theory and practice of quantitatively comparing multiple sensors in disambiguation problems.

Finally, while making the argument rigorous is a challenging open problem, we go beyond the results presented here and conjecture that strong Sensor Information Monotonicity in the general graph context holds for the Reset Protocol investigated herein and in Aksakalli *et al* (2008).

Acknowledgements—XY, DEF, and CEP were supported in part by Office of Naval Research grant N000140610013. LA was supported in part by The Acheson J Duncan Fund for the Advancement of Research in Statistics grant 08–17. We thank the Johns Hopkins University Center for Imaging Science for providing substantial computing resources.

References

Ahuja RK, Magnanti TL and Orlin JB (1993). *Network flows: Theory, algorithms, and applications*. Prentice Hall: Upper Saddle River, NJ.

Aksakalli V, Fishkind DE, Priebe CE and Ye X (2008). *The CR Disambiguation Protocol*, submitted for publication.

Andreatta G and Romeo L (1988). Stochastic shortest paths with recourse. *Networks* **18**: 193–204.

Baglietto M, Battistelli G, Vitali F and Zoppoli R (2003). Shortest path problems on stochastic graphs: A neuro dynamic programming approach. *Proceedings of the 42nd IEEE Conference on Decision and Control*, pp 6187–6193.

Bar-Noy A and Schieber B (1991). The Canadian traveller problem. *SODA'91: Proceedings of the Second Annual ACM-SIAM Symposium on Discrete Algorithms*, pp 261–270.

Blei DM and Kaelbling LP (1999). *Shortest paths in a dynamic uncertain domain*. IJCAI Workshop on Adaptive Spatial Representations of Dynamic Environments.

Fishkind DE, Priebe CE, Giles K, Smith LN and Aksakalli V (2007). Disambiguation protocols based on risk simulation. *IEEE Trans Syst Man & Cybern Part A* **37**: 814–823.

Olson T, Pang JS and Priebe CE (2002). A likelihood-MPEC approach to target classification. *Math Program Ser A* **96**: 1–31.

Papadimitriou CH and Yannakakis M (1991). Shortest paths without a map. *Theor Comput Sci* **84**: 127–150.

Piatko CD, Priebe CE, Cowen LJ, Wang IJ and McNamee P (2001). Path planning for mine countermeasures command and control. *Proceedings of the SPIE* **4394**: 836–843.

Piatko CD, Diehl CP, McNamee P, Resch C and Wang IJ (2002). Stochastic search and graph techniques for MCM path planning. *Proceedings of the SPIE* **4742**: 583–592.

Polychronopoulos GH and Tsitsiklis JN (1996). Stochastic shortest path problems with recourse. *Networks* **27**: 133–143.

Priebe CE, Olson TE and Healy Jr DM (1997). Exploiting stochastic partitions for minefield detection. *Proceedings of the SPIE* **3079**: 508–518.

Priebe CE, Pang JS and Olson T (1999). Optimizing sensor fusion for classification performance. *Proceedings of the CISST '99 International Conference*, Las Vegas, Nevada, June 1999, pp 397–403.

Priebe CE, Naiman DQ and Cope L (2001). Importance sampling for spatial scan analysis: Computing scan statistic *p*-values for marked point processes. *Comput Stat Data An* **35**(4): 475–485.

Priebe CE, Fishkind DE, Abrams L and Piatko CD (2005). Random disambiguation paths for traversing a mapped hazard field. *Nav Res Log* **52**: 285–292.

Provan JS (2003). A polynomial-time algorithm to find shortest paths with recourse. *Networks* **42**: 115–125.

Smith DL (1995). Detection technologies for mines and minelike targets. *Proceedings of the SPIE: Detection Technologies for Mines and Minelike Targets*, Orlando, Florida, April 1995, Vol. 2496, pp 404–408.

Washburn A (1999). Mine warfare. In: Wagner DH, Mylander WC and Sanders TJ (eds). *Chapter 10 of Naval Operations Analysis*. Naval Institute Press: Annapolis, MD.

Witherspoon NH, Holloway JH, Davis KS, Miller RW and Dubey AC (1995). The Coastal Battlefield Reconnaissance and Analysis (COBRA) Program for Minefield Detection. *Proceedings of the SPIE: Detection Technologies for Mines and Minelike Targets*, Orlando, Florida, April 1995, Vol. 2496, pp 500–508.

Appendix

Stochastic order and the ordering of sensors by sensitivity

For any random variable Θ, let F_Θ denote the cumulative distribution function, and let F_Θ^{-1} denote the quantile function; of course, if F_Θ is injective then F_Θ^{-1} is the inverse function of F_Θ, otherwise, for all $u \in (0, 1)$, recall that $F_\Theta^{-1}(u) := \inf\{w \in \mathbb{R} : F_\Theta(w) \geqslant u\}$. (So, for example, $F^{-1}(0.25)$, $F^{-1}(0.5)$, and $F^{-1}(0.75)$ are the first quartile, median, and third quartile of the distribution, that is, the probability of being less than or equal to these numbers is 0.25, 0.5, and 0.75, respectively.) The random variable Θ_1 is said to be *stochastically greater than or equal to* the random variable Θ_2, denoted $\Theta_1 \geqslant_{sto} \Theta_2$, if $F_{\Theta_1}(w) \leqslant F_{\Theta_2}(w)$ for all $w \in \mathbb{R}$; straightforward computation confirms that this condition is equivalent to the condition that $F_{\Theta_1}^{-1}(u) \geqslant F_{\Theta_2}^{-1}(u)$ for all $u \in (0, 1)$.

Thus, in the notation from this paper, when we say that sensor (F_T, F_N) is *at least as sensitive as* sensor $(F_{T'}, F_{N'})$ precisely when $F_T \geqslant_{sto} F_{T'}$ and $F_{N'} \geqslant_{sto} F_N$, what is meant is that, conditioning on an edge being traversable, each quantile for the marks generated by the at-least-as-sensitive-sensor (random vector $\vec{\varrho}^{F_T, F_N}$) is greater than or equal to that same quantile for the marks generated by the other sensor and, conditioning on an edge not being traversable, each quantile for the marks generated by the at-least-as-sensitive-sensor is less than or equal to that same quantile for the marks generated by the other sensor. So, very informally, if an edge is actually traversable then a more sensitive sensor can be more optimistic about the edge's traversability than the less sensitive sensor, and if an edge is actually nontraversable then the more sensitive sensor can be more pessimistic about the edge's traversability than the less sensitive sensor.

Also note that $\Theta_1 \geqslant_{sto} \Theta_2$ if and only if it holds that, for all nondecreasing real functions g, $\mathrm{E}[g(\Theta_1)] \geqslant \mathrm{E}[g(\Theta_2)]$. Thus, in the notation of this paper, if a protocol is strongly monotone then it is weakly monotone, but the converse does not necessarily hold.

Received July 2008;
accepted September 2009 after one revision