On the Mark Information Monotonicity in Random Disambiguation Paths

Xugang Ye¹*, Carey E. Priebe¹, Donniell E. Fishkind¹, Lowell Abrams²

¹Department of Applied Mathematics and Statistics, Whiting School of Engineering, Johns Hopkins University, Baltimore, Maryland, 21218
²Department of Mathematics, George Washington University, Washington, DC, 20052

Abstract

We consider the problem of navigating an agent to safely and swiftly traverse a nondeterministic network along a random disambiguation path (RDP). Other than assuming that the probability of existence of any nondeterministic is known as a priori, we assume that a sensor characterized by two conditional distribution functions is available to generate a maker for each nondetermimistic arc after an observation. We study whether the improvement on the sensor quality incurs the reduction of the total cost (traveling cost plus disambiguation cost) the agent pays to reach the target given a navigation protocol and in the same terrain. We introduce the concepts of stochastic ordering of sensors and mark information monotonicity (MIM) of protocols. We propose two types of concrete navigation protocols: threshold protocol and penalty protocol and introduce some preliminary analytical results. We also introduce some Monte Carlo simulation results in minefield application.

Keywords: Random Disambiguation Paths; Ordering of Sensors; Mark Information Monotonicity; Navigation Protocol

1. Introduction

The problem of navigating an agent to traverse a nondeterministic graph has been studied for a long time. Although the ultimate goals look quite common (e.g., the agent should swiftly reach its target, or the agent should pay relatively small), there exist many different formulations concerning different application areas. Two well-known formulations can be summarized as planning in partially known environment and planning in probabilistic world. The formulation as planning in partially known environment assumes a given graph topology. The uncertainty is the cost function of the graph. Although the graph is uncertain before the agent starts to move, during the travel, any local uncertainty can be removed. Many replanning strategies for this formulation have been proposed and studied [1-6]. The main issues are the conditional optimality and computational efficiency [5, 6]. The formulation as planning in probabilistic world also assumes a given graph topology, and the uncertainty also comes from the cost function of the graph. The difference is that this formulation assumes a probability distribution of the cost function. For this reason, this formulation is also called stochastic shortest path model in literature. Some representative work on stochastic shortest path model can be found in [7-11]. The concept of random disambiguation path (RDP) was introduced in [12-14], where the addressed problem is similar to the second formulation. The difference is that when the agent encounters an uncertain arc/edge and disambiguates, a cost is incurred. Hence the agent may not disambiguate all the encountered uncertain arcs/edges. The objective function to be minimized is the total traveling cost plus the total disambiguation cost. In the statement of the problem, the underlying distribution is Bernoulli. Like many other work, the parameters of the distribution are assumed to be known.

The subject of this paper is still RDP, but a distinguished feature of our work is that we
don’t assume the availability of the parameters of the underlying distribution. Based on the practices of some real-world navigation systems (e.g., planet rover, submarine, ground armored vehicle etc.), we model the random disambiguation path problem with an important concept: mark information. We assume that a global sensor makes observation on the graph and generates markers for the arcs/edges as the guiding information for the agent. We claim that there exists some traveling protocols, or strategies, that can do better if the makers more precisely capture the true state of the graph. We call this relation the Mark Information Monotonicity (MIM). The rest of the paper is organized as follows. In Section 2, we introduce notations and terminologies and develop the concept of MIM. In Section 3, we propose two types of navigation protocols. In Section 4, we introduce some preliminary analytical MIM results. In section 5, we introduce some Monte Carlo simulation results on the MIM in minefield application. Finally, in section 6, we give a summary as well as some suggestions for further research.

2. Notations, Terminologies, and Definitions

Let \( G = (V, \mathcal{A} \cup \mathcal{B}, \; \vec{L}, \; \vec{c}, \; \vec{\rho}) \) be a finite graph, where \( V \) is set of nodes that contains a specified starting node \( s \) and a specified terminal node \( t \), \( \mathcal{A} \) is set of deterministic arcs, \( \mathcal{B} \) is set of nondeterministic arcs, \( \vec{L} \) is arc length vector with component \( L_a \) representing the nonnegative finite length of the arc \( a \in \mathcal{A} \cup \mathcal{B}, \; \vec{c} \) is disambiguation cost vector with component \( c_a \) representing the nonnegative finite disambiguation cost of the arc \( \tilde{a} \in B \), and \( \vec{\rho} \) is a vector with component \( \rho_{\tilde{a}} \) representing the probability that the arc \( \tilde{a} \in B \) is not traversable. For convenience we define \( \vec{X} \) as an indicator vector with component \( X_{\tilde{a}} \) being independently Bernoulli(\( \rho_{\tilde{a}} \))- distributed: \( P(X_{\tilde{a}} = 1) = \rho_{\tilde{a}} \) and \( P(X_{\tilde{a}} = 0) = 1 - \rho_{\tilde{a}} \).

If \( \vec{\rho} \) is known, then we have a context for classical RDP problem [12]. Here, we are interested in the case that \( \vec{\rho} \) is unknown but there exists a realization of a random vector, say \( \vec{Y} \), with component \( Y_{\tilde{a}} \) representing the marker of the arc \( \tilde{a} \in B \). For each \( \tilde{a} \in B \), \( Y_{\tilde{a}} \) is conditionally distributed as \( Y_{\tilde{a}} \mid X_{\tilde{a}} = 0 \sim F_0 \) and \( Y_{\tilde{a}} \mid X_{\tilde{a}} = 1 \sim F_1 \), where \( F_0 : [0,1] \to [0,1] \) and \( F_1 : [0,1] \to [0,1] \) are two distribution functions. We use the notation \( S = (F_0, \; F_1) \) to denote the sensor and use the notation \( \vec{Y} \sim S \) to denote that \( \vec{Y} \) is generated by \( S \) for \( G \) from either the distribution \( F_0 \) or the distribution \( F_1 \) conditioning on \( \vec{X} \).

A sensor \( S \) is said to be valid if \( F_0(y) \geq F_1(y) \) for any \( 0 \leq y \leq 1 \). A valid sensor is said to be discerning if \( F_0(1/2) \geq 1/2 \) and \( F_1(1/2) \leq 1/2 \). Consider two sensors \( S^{(1)} = (F_0^{(1)}, \; F_1^{(1)}) \) and \( S^{(2)} = (F_0^{(2)}, \; F_1^{(2)}) \). For graph \( G \), suppose \( Y^{(1)} \sim S^{(1)} \) and \( Y^{(2)} \sim S^{(2)} \). We say that \( S^{(1)} \) is stochastically at least as good as \( S^{(2)} \) and write \( Y^{(1)} \geq Y^{(2)} \), if for any \( 0 \leq y \leq 1 \), \( F_0^{(1)}(y) \geq F_0^{(2)}(y) \) and \( F_1^{(1)}(y) \leq F_1^{(2)}(y) \).

A disambiguation problem instance (DPI) is denoted as a tuple (\( G, \; \mathcal{A}, \; \mathcal{B}, \; \nu, \; t, \; \vec{L}, \; \vec{c}, \; \vec{\rho}, \; \vec{Y} \)), where \( \nu \) is the node that the agent is currently located at and \( \vec{Y} \) is a realization of \( \vec{Y} \). A navigation protocol, denoted as \( \varphi \), is a function that for each DPI, \( \varphi \) returns empty set \( \phi \) when \( \nu = t \) or otherwise returns an arc \( a \in \mathcal{A} \cup \mathcal{B} \). Starting from the initial DPI in
which \( v = s \) and given a realization \( \tilde{x} \) of \( \tilde{X} \), the agent keeps asking an arc from \( \mathcal{P} \). If \( \mathcal{P} \) does not return any arc, then the agent reaches the target \( t \) and stop; otherwise, the agent faces an arc \( a \in A \cup B \). If \( a \in A \), then the agent will decide whether to traverse \( a \) or to ask another arc from \( \mathcal{P} \). If the agent traverses \( a \), then \( v \) is updated, the agent enters next DPI. If \( a \in B \), then the agent will decide whether to disambiguate \( a \) or to ask another arc from \( \mathcal{P} \). If the agent disambiguates \( a \), the agent will update \( A \) and \( B \) according to the disambiguation result. If \( x_a = 0 \), then the agent actually faces a deterministic arc and will decide whether to traverse it or ask another arc from \( \mathcal{P} \). If \( x_a = 1 \), then the agent must ask another arc from \( \mathcal{P} \). In either case, the agent will enter next DPI. When the agent travels, the agent actually goes through a sequence of DPIs, in which the sequence of locations of the agent defines an \( s-t \) path that may contain repeated nodes (note that here the concept of RDP is different from the definition of path in Graph Theory). The total length plus the total disambiguation cost defines the total cost of this path.

The agent is said to be \textit{balk-free} if it acts on the arc \( a \) it asks from \( \mathcal{P} \) in the following way. If \( a \in A \), then the agent traverses it. If \( a \in B \), then the agent disambiguates it. If \( x_a = 0 \), then the agent traverses it and \( A, B \), and \( v \) are updated; otherwise, \( B \) is updated and the agent will ask the next arc from \( \mathcal{P} \). In this paper, we focus on balk-free agent.

We use \( C = C \left( G, s, t, \tilde{X}, \tilde{Y}, \mathcal{P} \right) \) to denote the cost (traveling cost plus disambiguation cost) the agent pays to travel from \( s \) to \( t \). \( C \) is a random variable since both \( \tilde{X} \) and \( \tilde{Y} \) are random. Moreover, there may also be some randomness in \( \mathcal{P} \) if \( \mathcal{P} \) contains some randomized algorithm. To measure the utility of \( \mathcal{P} \), we consider the distribution function \( H_C(x) = \mathbb{P} \left( C \leq x \right) \), \( x \geq 0 \), the leftmost \( P_0 \)-quantile \( C_0 = \inf_x \{ x: H_C(x) \geq P_0 \} \), and the mean \( \mathbb{E}(C) \).

\begin{definition}
Let \( C^{(1)} = C(G, s, t, X, Y^{(1)}, \mathcal{P}) \) and \( C^{(2)} = C(G, s, t, X, Y^{(2)}, \mathcal{P}) \), where \( Y^{(1)} \sim S^{(1)} \) and \( Y^{(2)} \sim S^{(2)} \). We say \( \mathcal{P} \) is \textit{strongly monotone} with respect to markers if \( Y^{(1)} \succeq Y^{(2)} \) implies \( H_{C^{(1)}}(x) \geq H_{C^{(2)}}(x) \) for any \( x \geq 0 \).
\end{definition}

\begin{definition}
Let \( C^{(1)} = C(G, s, t, X, Y^{(1)}, \mathcal{P}) \) and \( C^{(2)} = C(G, s, t, X, Y^{(2)}, \mathcal{P}) \), where \( Y^{(1)} \sim S^{(1)} \) and \( Y^{(2)} \sim S^{(2)} \). We say \( \mathcal{P} \) is \textit{\( P_0 \)-monotone} with respect to markers for some \( 0 \leq P_0 \leq 1 \) if \( Y^{(1)} \succeq Y^{(2)} \) implies \( C_0^{(1)} \leq C_0^{(2)} \), where \( C_0^{(1)} = \inf_x \{ x: H_{C^{(1)}}(x) \geq P_0 \} \) and \( C_0^{(2)} = \inf_x \{ x: H_{C^{(2)}}(x) \geq P_0 \} \).
\end{definition}

\begin{definition}
Let \( C^{(1)} = C(G, s, t, X, Y^{(1)}, \mathcal{P}) \) and \( C^{(2)} = C(G, s, t, X, Y^{(2)}, \mathcal{P}) \), where \( Y^{(1)} \sim S^{(1)} \) and \( Y^{(2)} \sim S^{(2)} \) and assume the finiteness of expectation. We say \( \mathcal{P} \) is \textit{weakly monotone} with respect to markers if \( Y^{(1)} \succeq Y^{(2)} \) implies \( \mathbb{E}(C^{(1)}) \leq \mathbb{E}(C^{(2)}) \).
\end{definition}

It can be easily shown (in [15]) that strong monotonicity implies \( P_0 \)-monotonicity and weak monotonicity. Moreover, if \( \mathcal{P} \) is \( P_0 \)-monotone with respect to markers for any \( 0 \leq P_0 \leq 1 \), then \( \mathcal{P} \) is strongly monotone with respect to markers.
It’s desirable for a protocol $P$ to be strongly monotone with respective to markers. However, it’s generally difficult to determine whether $P$ has strong monotonicity property since one must exhaustively explore the structure of the cost distribution. It is intuitively clear that it may be much easier to determine whether $P$ has weak monotonicity property because evaluating expectation averages out the distribution structure. Also as we will show later on, recursive patterns in the calculation of mean can sometimes be easily identified.

A disadvantage of the concept of weak monotonicity is that it requires finite expectation. Practically, this may not always be satisfied. For instance if there is a tiny probability that $\mathbb{C} = +\infty$. Then the mean is not a proper measure of the utility of the protocol $P$. However, the leftmost $P_0$-quantile for some $0 \leq P_0 \leq 1$ still works as a utility measure. Like strong monotonicity, to determine whether $P$ is $P_0$-monotone with respect to markers depends on the knowledge of the cost distribution. But this job is much easier to do since we only need to do one comparison.

Due to the difficulty of analytically determining the cost distribution function, the numerical approach based on frequency statistics of Monte Carlo simulation results is more favorable as a nonparametric method to assess the leftmost $P_0$-quantile.

3. Protocols and Agent

This section concerns specific protocols, which include specific mechanism to return an arc and specific balk-free agent, which has a specific way of updating the DPIs. We propose two types of protocols. One is called threshold protocol. The other is called penalty protocol.

The basic idea of the threshold protocol is to screen out those arcs that are not likely to be traversable. The screening is based on the markers. In the threshold protocol, a threshold vector $\tilde{\alpha}$ is predetermined, with its component $\alpha_{\tilde{a}}$ representing the threshold of the arc $\tilde{a} \in B$. Given a DPI, the threshold protocol defines the arc cost function in this way: for each $a \in A$, $W(a) = L_a$; for each $\tilde{a} \in B$, $W(\tilde{a}) = c_{\tilde{a}} + L_{\tilde{a}}$. Upon the request from the agent, the threshold protocol finds a shortest $v$-$t$ path in $G$ with respect to the cost function $W$. When finding the shortest path, any arc $\tilde{a} \in B$ that satisfies $c_{\tilde{a}} \geq \alpha_{\tilde{a}}$ is not used. Also when finding the shortest path, tie breaking favors deterministic arcs. The first arc along the shortest $v$-$t$ path is delivered to the agent.

The basic idea of the penalty protocol is to put a penalty on any arc that seems to be untraversable. In the penalty protocol, a penalty function vector $\tilde{l}$ is predetermined, with its component $l_{\tilde{a}} = \tilde{l}_{\tilde{a}}(c_{\tilde{a}}, L_{\tilde{a}}, y_{\tilde{a}})$ being the penalty function associated with the arc $\tilde{a} \in B$. For each $\tilde{a} \in B$, $\tilde{l}_{\tilde{a}} \geq 0$ is a continuous, monotonically increasing function of the realization $y_{\tilde{a}}$ of the marker $Y_{\tilde{a}}$ and $\tilde{l}_{\tilde{a}}(c_{\tilde{a}}, L_{\tilde{a}}, y_{\tilde{a}}) \rightarrow c_{\tilde{a}} + L_{\tilde{a}}$ as $y_{\tilde{a}} \rightarrow 0$; $\tilde{l}_{\tilde{a}}(c_{\tilde{a}}, L_{\tilde{a}}, y_{\tilde{a}}) \rightarrow +\infty$ as $y_{\tilde{a}} \rightarrow 1$. Given a DPI, the penalty protocol defines the arc cost function in this way: for each $a \in A$, $W(a) = L_a$; for each $\tilde{a} \in B$, $W(\tilde{a}) = \tilde{l}_{\tilde{a}}(c_{\tilde{a}}, L_{\tilde{a}}, y_{\tilde{a}})$. As same as the threshold protocol, upon the request from the agent, the penalty protocol finds a shortest $v$-$t$ path in $G$ with respect to the cost function $W$ and when finding the shortest path, tie breaking favors deterministic arcs. The first arc along the shortest $v$-$t$...
path is delivered to the agent.

As mentioned before, we consider balk-free agent. Furthermore, to facilitate the theoretical analysis, we assume any DPI is updated in simple version. Firstly, $v$ is updated as long as the agent traverses an arc. Secondly, when an arc $\tilde{a} \in B$ is disambiguated, if $x_{\tilde{a}} = 0$, then $A$ is updated into $A \cup \{a\}$ and $B$ is updated into $B \setminus \{a\}$; if $x_{\tilde{a}} = 1$, then $B$ is updated into $B \setminus \{a\}$.

The interaction between the agent and either the threshold protocol or the penalty protocol can be efficiently implemented. Suppose the protocol finds a shortest $v$-$t$ path in $G$. The agent can actually keeps requesting arc from this path and moves along this path until the agent encounters the first nondeterministic arc on this path. This means each time after the agent traverses an arc, the protocol only needs to recalculate path when the next arc the agent encounters is not traversable.

Since different choice of the threshold vector $\tilde{a}$ results in different version of threshold protocol, we denote $\mathcal{P}_t(\tilde{a})$ as the threshold protocol with threshold vector $\tilde{a}$. Practically, there is no way to know what the best threshold vector is. But theoretically, the best $\tilde{a}$, denoted as $\tilde{a}^*$, can be defined under some protocol utility measures introduced in section 2. Parameterized by $\tilde{a}$, Let $H_C(x; \tilde{a})$, $x \geq 0$ be the distribution function of $C = C(G, s, t, X, Y, \mathcal{P}_t(\tilde{a}))$. We can define $\tilde{a}^{*2}(P_0) = \arg \min \inf_{\tilde{a}} \{x: H_C(x; \tilde{a}) \geq P_0\}$ for some $0 \leq P_0 \leq 1$ as the best $\tilde{a}$ under the leftmost $P_0$-quantile measure. We can also define $\tilde{a}^{*3} = \arg \min \mathbb{E}[C(G, s, t, X, Y, \mathcal{P}_t(\tilde{a}))]$ as the best $\tilde{a}$ under the mean measure.

It’s also obvious that different penalty function leads to different version of penalty protocol. For any $\tilde{a} \in B$, several penalty functions could be proposed as follows.

1) $\tilde{I}_{a,1}(c_{\tilde{a}}, L_{\tilde{a}}, Y_{\tilde{a}}) = c_{\tilde{a}} + L_{\tilde{a}}/(1 - Y_{\tilde{a}})$, \hspace{1cm} (3.1)
2) $\tilde{I}_{a,2}(c_{\tilde{a}}, L_{\tilde{a}}, Y_{\tilde{a}}) = L_{\tilde{a}} + c_{\tilde{a}}/(1 - Y_{\tilde{a}})$, \hspace{1cm} (3.2)
3) $\tilde{I}_{a,3}(c_{\tilde{a}}, L_{\tilde{a}}, Y_{\tilde{a}}) = c_{\tilde{a}} + L_{\tilde{a}} + Y_{\tilde{a}}/(1 - Y_{\tilde{a}})$, \hspace{1cm} (3.3)
4) $\tilde{I}_{a,4}(c_{\tilde{a}}, L_{\tilde{a}}, Y_{\tilde{a}}) = c_{\tilde{a}} + L_{\tilde{a}} + 1 - \log(1 - Y_{\tilde{a}})$, \hspace{1cm} (3.4)
5) $\tilde{I}_{a,5}(c_{\tilde{a}}, L_{\tilde{a}}, Y_{\tilde{a}}) = c_{\tilde{a}} + L_{\tilde{a}} + Y_{\tilde{a}} \cdot [1 - \log(1 - Y_{\tilde{a}})]$. \hspace{1cm} (3.5)

Of course, there can also be many other penalty functions besides (3.1-3.5). It’s hard to say which penalty function is the best since different combination of penalty functions also leads to different version of penalty protocol. Let $\mathcal{P}_2(\tilde{I})$ denote the penalty protocol defined by penalty function vector $\tilde{I}$. To find the best penalty protocol under the leftmost $P_0$-quantile measure for some $0 \leq P_0 \leq 1$ is to find a penalty function vector $\tilde{I}^{*2}$ that minimizes $\inf_x \{x: H_C(x; \tilde{I}) \geq P_0\}$, where $H_C(x; \tilde{I})$, $x \geq 0$ is the distribution function of $C = C(G, s, t, X, Y, \mathcal{P}_2(\tilde{I}))$. To find the best penalty protocol under the mean measure is to find a penalty function vector $\tilde{I}^{*3}$ that minimizes $\mathbb{E}[C(G, s, t, X, Y, \mathcal{P}_2(\tilde{I}))]$. 


4. Analytical MIM Results

In this section, we introduce some preliminary analytical results we obtained so far. We first show that that both the threshold protocol and the penalty protocol can guarantee that the agent is able to reach the target $t$ as long as the graph $G$ is convergent with respect to $t$. We then show that the threshold protocol has weak monotonicity property for parallel graph. We also show that there exists some relation between the threshold protocol and the penalty protocol.

**Definition 4.** Graph $G$ is called convergent with respective to $t$ if for any $v \in V$, $v \neq t$, there is a $v$-$t$ path that contains only the arcs in $A$.

**Theorem 5.** If $G$ is convergent with respect to $t$, the agent is balk-free, and the DPI updates are simple, then the expected cost the agent pays to travel from any $v \in V$ to $t$ is finite under any version of threshold protocol or penalty protocol.

**Proof.** Note that the finiteness of $G$ implies there are only finite possibilities. Suppose (for contradiction) that there exists a positive probability that the agent pays infinitely large cost to reach $t$ (or, the agent is never able to reach $t$). By finiteness of $G$, the agent at most pays finite disambiguation cost. Hence it must be stuck in some cycle. Note that both protocols define nonnegative cost functions and plan shortest path, by convergence of $G$, the cycle will eventually be avoided, a contradiction! □

A special case of $G$ that is convergent with respect to $t$ is called parallel graph. In this case, $V = \{s, t\}$ and $|A| \geq 1$. We next show that the threshold protocol has weak monotonicity property in a parallel graph.

**Theorem 6.** If $G$ is parallel graph, the agent is balk-free, and the DPI updates are simple, then for any threshold vector $\alpha = (\alpha_r, P_1(\alpha_r))$ is weakly monotone with respect to markers. That is, if $Y(1) \preceq S(1)$ and $Y(2) \preceq S(2)$, then $Y(1) \preceq Y(2)$ implies $E(C(1)) \leq E(C(2))$, where $C(1) = C(G, s, t, X, Y(1), P_1(\alpha_r))$ and $C(2) = C(G, s, t, X, Y(2), P_1(\alpha_r))$.

**Proof.** At first, without losing generality, we can assume that $|A| = 1$ since only the shortest deterministic arc may affect the cost. Suppose $A = \{a\}$. Note that for any $\tilde{a} \in B$, if $L_{\tilde{a}} \leq c_{\tilde{a}} + L_{\tilde{a}}$, then $\tilde{a}$ can be removed without affecting the cost, hence it’s also without losing generality that we can assume $L_{\tilde{a}} > c_{\tilde{a}} + L_{\tilde{a}}$ for any $\tilde{a} \in B$.

The proof is by induction on $|B|$. The base case is $|B| = 0$. In based case, $C = L_{\tilde{a}}$, which is constant, hence the weak monotonicity trivially holds. Suppose (inductive hypothesis) that weak monotonicity holds for $|B| = k \geq 0$, we then consider the case $|B| = k + 1$. Consider $\tilde{a} = \arg \min_{\tilde{a} \in B} W(\tilde{a}) = c_{\tilde{a}} + L_{\tilde{a}}$. Suppose $0 \leq \alpha_{\tilde{a}} \leq 1$ is the threshold for $\tilde{a}$. For convenience, denote $\gamma = E(C | Y_{\tilde{a}} < \alpha_{\tilde{a}}, X_{\tilde{a}} = 0)$, $\xi = E(C | Y_{\tilde{a}} < \alpha_{\tilde{a}}, X_{\tilde{a}} = 1)$, and $\eta = E(C | Y_{\tilde{a}} \geq \alpha_{\tilde{a}})$. Obviously,

$$\gamma = c_{\tilde{a}} + L_{\tilde{a}} \leq \eta$$

$$\leq \eta + c_{\tilde{a}}$$

$$= \xi < +\infty.$$
The last strict inequality is due to Theorem 5. We can compute (via conditioning)
\[
P(Y_a < \alpha_a, X_a = 0) = (1 - \rho_a) \cdot F_0(\alpha_a),
\]
\[
P(Y_a < \alpha_a, X_a = 1) = \rho_a \cdot F_1(\alpha_a),
\]
and
\[
P(Y_a \geq \alpha_a) = (1 - \rho_a) \cdot [1 - F_0(\alpha_a)] + \rho_a \cdot [1 - F_1(\alpha_a)].
\]
Hence
\[
E(C) = \gamma(1 - \rho_a) \cdot F_0(\alpha_a) + \xi \cdot \rho_a \cdot F_1(\alpha_a) + \eta(1 - \rho_a) \cdot [1 - F_0(\alpha_a)] + \eta \cdot \rho_a \cdot [1 - F_1(\alpha_a)].
\]

Now, consider \(C^{(1)} = C(G, s, t, X, Y^{(1)}, \mathcal{P}_1(\alpha))\) and \(C^{(2)} = C(G, s, t, X, Y^{(2)}, \mathcal{P}_1(\alpha))\), where \(Y^{(1)} \sim S^{(1)} = (F_0^{(1)}, F_1^{(1)})\) and \(Y^{(2)} \sim S^{(2)} = (F_0^{(2)}, F_1^{(2)})\). By the expression of \(E(C)\) and inductive hypothesis, we have
\[
E(C^{(1)}) - E(C^{(2)}) \leq (\gamma^{(2)} - \eta^{(2)}) \cdot \rho_a \cdot \eta \cdot \rho_a \cdot \xi \cdot \rho_a \cdot \eta(1 - \rho_a) \cdot [1 - F_0(\alpha_a)] + \eta \cdot \rho_a \cdot [1 - F_1(\alpha_a)].
\]
By \(\gamma^{(2)} \leq \eta^{(2)} \leq \xi^{(2)}\) and \(Y^{(1)} \geq Y^{(2)}\), we have \(E(C^{(1)}) - E(C^{(2)}) \leq 0. \) □

**Theorem 7.** If \(G\) is parallel graph, the agent is balk-free, and the DPI updates are simple, then \(Y^{(1)} \geq Y^{(2)}\) implies \(E[C(G, s, t, X, Y^{(1)}, \mathcal{P}_1(\alpha^{*3}))] \leq E[C(G, s, t, X, Y^{(2)}, \mathcal{P}_1(\alpha^{*3})), \mathcal{P}_1(\alpha))]\), where \(\alpha^{*3} = \arg\min_{\alpha} E[C(G, s, t, X, Y^{(1)}, \mathcal{P}_1(\alpha))]\) and \(\alpha^{*3} = \arg\min_{\alpha} E[C(G, s, t, X, Y^{(2)}, \mathcal{P}_1(\alpha))]\).

**Proof.** By definition of \(\alpha^{*3}\) and Theorem 6, we have
\[
E[C(G, s, t, X, Y^{(1)}, \mathcal{P}_1(\alpha^{*3}))] \leq E[C(G, s, t, X, Y^{(2)}, \mathcal{P}_1(\alpha^{*3}))] \leq E[C(G, s, t, X, Y^{(2)}, \mathcal{P}_1(\alpha^{*3}))]. \) □

Another special case of \(G\) that is convergent with respect to \(t\) is that \(|B| = 1\). In this case, we will show that the penalty protocol can be reduced to threshold protocol and furthermore, both protocols have the strong monotonicity property.

**Theorem 8.** If \(G\) is convergent with respect to \(t\), \(|B| = 1\), the agent is balk-free, and the DPI updates are simple, then both the threshold protocol and the penalty protocol are strongly monotone with respect to markers.

**Proof.** Suppose \(B = \{ \tilde{\alpha} = (u, v) \}\). We first consider the threshold protocol with some \(0 < \alpha_a < 1\) as the threshold for \(\tilde{\alpha}\). Suppose \(P_1\), with length \(L_1\), is a shortest \(s-t\) path that does not pass \(\tilde{\alpha}\); \(P_2\), with length \(L_2\), is a shortest \(s-u\) path; \(P_3\), with length \(L_3\), is a shortest \(v-t\) path; and \(P_4\), with length \(L_4\), is a shortest \(u-t\) path that does not pass \(\tilde{\alpha}\). Since \(G\) is convergent with respect to \(t\), the three paths \(P_1\), \(P_3\), and \(P_4\) must exist. If \(P_2\) does not exist, then \(C = L_1\), strong monotonicity trivially holds. We consider the case that \(P_2\) exists as Figure 4.1 shows.
which implies

\[ L_1 \leq L_2 + L_4 \leq c_{\bar{a}} + L_2 + L_4. \] If \( L_1 \leq c_{\bar{a}} + L_{\bar{a}} + L_2 + L_3 \), then \( C = L_1 \), strong monotonicity trivially holds too. We now consider the nontrivial case that \( L_1 > c_{\bar{a}} + L_{\bar{a}} + L_2 + L_3 \). It’s easy to find that this cost distribution function actually implies strong monotonicity. We consider the nontrivial case that \( L_1 > c_{\bar{a}} + L_{\bar{a}} + L_2 + L_3 \).

\[
C = \begin{cases}
  L_1 & \text{if } Y_{\bar{a}} \geq \alpha_{\bar{a}}; \\
  c_{\bar{a}} + L_{\bar{a}} + L_2 + L_3 & \text{if } Y_{\bar{a}} < \alpha_{\bar{a}} \text{ and } X_{\bar{a}} = 0; \\
  c_{\bar{a}} + L_2 + L_4 & \text{if } Y_{\bar{a}} < \alpha_{\bar{a}} \text{ and } X_{\bar{a}} = 1,
\end{cases}
\] (4.1)

which implies

\[
H_C(c_{\bar{a}} + L_2 + L_4) = P(C \leq c_{\bar{a}} + L_2 + L_4) = 1,
\]

\[
H_C(L_1) = P(C \leq L_1) = (1 - \rho_{\bar{a}}) [1 - F_0(\alpha_{\bar{a}})] + \rho_{\bar{a}} [1 - F_1(\alpha_{\bar{a}})] + (1 - \rho_{\bar{a}}) \cdot F_0(\alpha_{\bar{a}})
\]

\[
= 1 - \rho_{\bar{a}} \cdot F_1(\alpha_{\bar{a}}), \text{ and}
\]

\[
H_C(c_{\bar{a}} + L_{\bar{a}} + L_2 + L_3) = P(C \leq c_{\bar{a}} + L_{\bar{a}} + L_2 + L_3) = (1 - \rho_{\bar{a}}) \cdot F_0(\alpha_{\bar{a}}).
\]

It’s easy to find that this cost distribution function actually implies \( P_0 \)-monotonicity for any \( 0 \leq P_0 \leq 1 \), hence strong monotonicity.

We then consider the penalty protocol. Actually, given a penalty function \( \tilde{\tilde{I}}_{\bar{a}}(c_{\bar{a}}, L_{\bar{a}}, Y_{\bar{a}}) \), it’s crucial to compare \( L_1 \) with \( L_2 + \tilde{\tilde{I}}_{\bar{a}}(c_{\bar{a}}, L_{\bar{a}}, Y_{\bar{a}}) + L_3 \) because

\[
C = \begin{cases}
  L_1 & \text{if } L_1 \leq L_2 + \tilde{\tilde{I}}_{\bar{a}}(c_{\bar{a}}, L_{\bar{a}}, Y_{\bar{a}}) + L_3; \\
  c_{\bar{a}} + L_{\bar{a}} + L_2 + L_3 & \text{if } L_1 \geq L_2 + \tilde{\tilde{I}}_{\bar{a}}(c_{\bar{a}}, L_{\bar{a}}, Y_{\bar{a}}) + L_3 \text{ and } X_{\bar{a}} = 0; \\
  c_{\bar{a}} + L_2 + L_4 & \text{if } L_1 \geq L_2 + \tilde{\tilde{I}}_{\bar{a}}(c_{\bar{a}}, L_{\bar{a}}, Y_{\bar{a}}) + L_3 \text{ and } X_{\bar{a}} = 1,
\end{cases}
\] (4.2)

By the definition of \( \tilde{\tilde{I}}_{\bar{a}} \), \( \tilde{\tilde{I}}_{\bar{a}}(c_{\bar{a}}, L_{\bar{a}}, Y_{\bar{a}}) \geq c_{\bar{a}} + L_{\bar{a}} \). If \( L_1 \leq c_{\bar{a}} + L_{\bar{a}} + L_2 + L_3 \), then \( L_1 \leq L_2 + \tilde{\tilde{I}}_{\bar{a}}(c_{\bar{a}}, L_{\bar{a}}, Y_{\bar{a}}) + L_3 \), hence \( C = L_1 \), which implies strong monotonicity. We consider the nontrivial case that \( L_1 > c_{\bar{a}} + L_{\bar{a}} + L_2 + L_3 \). Under fixed \( c_{\bar{a}} \) and \( L_{\bar{a}} \), let \( \psi(Y_{\bar{a}}) = \tilde{\tilde{I}}_{\bar{a}}(c_{\bar{a}}, L_{\bar{a}}, Y_{\bar{a}}) - (L_1 - L_2 - L_3) \). Note that \( \psi(Y_{\bar{a}}) \) is also a continuous, monotonically increasing function of \( Y_{\bar{a}} \) and \( \psi(Y_{\bar{a}}) < 0 \) as \( Y_{\bar{a}} \to 0 \); \( \psi(Y_{\bar{a}}) \to +\infty \) as \( Y_{\bar{a}} \to 1 \). By intermediate value theorem
in elementary calculus, \( \psi(Y_a) = 0 \) has a unique root, say \( Y_a^* \) in \((0, 1)\). Again, since \( \psi(Y_a) \) is monotonically increasing, we have that \( L_1 \leq L_2 + \tilde{I}_a(c_a, L_a, Y_a) + L_3 \Leftrightarrow \psi(Y_a) \geq 0 \Leftrightarrow Y_a \geq Y_a^* \), which implies that (4.2) has the same form as (4.1) except that \( \alpha^* \) in (4.1) can be arbitrarily chosen from \([0,1]\) and \( Y_a^* \) in (4.2) is specified by the definition of \( \tilde{I}_a \). Hence penalty protocol reduces to threshold protocol and the strong monotonicity of penalty protocol is implied. □

5. Numerical MIM Results in Minefield Application

In this section, we introduce some experimental results from applying a penalty protocol to guide a balk-free agent to traverse a minefield.

A minefield is typically modeled as \( m \) detected risk centers \( d_1, d_2, \ldots, d_m \in S \subseteq \mathbb{R}^2 \), with each \( d_i \) being either a true detection or a false detection and \( S \) denoting a bounded region. For \( i = 1, 2, \ldots, m \), we denote \( X_i \) as the indicator that \( d_i \) is a true detection or not. That is \( X_i = 0 \) if \( d_i \) is a false detection; \( X_i = 1 \) if \( d_i \) is a true detection. For each risk center \( d_i \), there is a marked disk shape risk region, say \( D_i \), that is centered at \( d_i \) and has radius \( r_i > 0 \). If a disambiguation yields \( X_i = 0 \), then \( d_i \) should be removed. But the agent still cannot traverse \( D_i \) if \( D_i \) intersects another marked disk shape risk region, say \( D_j \). If the same disambiguation shows \( X_i = 1 \), then \( d_i \) is confirmed to be a true hazard center. In this case, \( D_i \) becomes a forbidden region that should never be traversed. Now, guided by a global sensor that generates a marker \( 0 < Y_i < 1 \) for each \( d_i \), the agent is supposed to travel from a starting location \( s \in S \subseteq \mathbb{R}^2 \) to a target location \( t \in S \subseteq \mathbb{R}^2 \), traversing the union of all \( D_i \)'s when necessary. The agent should safely and swiftly reach the target \( t \). During the travel, the agent collects the new information through (local) disambiguations and updates the knowledge of the world.

The problem of robot motion planning has been extensively studied for a long time. Although the problem of traversing minefield cannot be completely fit into any existing framework, some well-developed techniques are quite useful. The first one is to represent the 2-D world by grid. One advantage of using grid is that the computational complexity can be controlled by adjusting the resolution. Compared with grid, visibility graph [12] is a more precise map representation. However, visibility graph is only practical when the number of obstacles is small and the shape of the obstacles is simple. In minefield application, it’s very common that the number of detections will exceed 100. To construct visibility graph will be computationally inefficient. While representing the world by grid can lead to much less computational efforts when the size of the cell is adjusted to a proper level. In our experiments, we use the 8-connected grid illustrated in Figure 5.1, where \( \delta \) denotes the cell size.

Since in the minefield model, the uncertainty is represented by regions, the penalty terms in the protocol won’t just simply be those proposed as any one of (3.1-3.5). In our experiments, the formation of the penalty terms is more complicated. For any grid node \( v \) and any risk center \( d_i \), define \( Y_{v,i} = Y_i \) if \( v \in D_i \); \( Y_{v,i} = 0 \) otherwise. Hence \( Y_{v,i} \) is the “marker” of \( v \) associated with \( d_i \). For any two adjacent grid nodes \( u \) and \( v \), let \( u\overrightarrow{v} \) denote the line segment that connects \( u \) and \( v \). We define the cost function of \( u\overrightarrow{v} \) as...
Figure 5.1. Illustration of 8-connected grid world

Figure 5.2. Probability density functions of Beta distribution ($s = 1, 2$)

Figure 5.3. A collection of $m = 100$ detections independently generated from uniform $([10, 90] \times [10, 90])$

(lef: risk centers; right: disk shape risk regions)
where, $L(\overline{uv})$ is the Euclidean distance between $u$ and $v$, $c(\overline{uv})$ is the disambiguation cost of $\overline{uv}$, and $Y_{uv} = \max\{\max_{i=1,\ldots,m} Y_{i,u}, \max_{i=1,\ldots,m} Y_{i,v}\}$. Obviously, if both end points of $\overline{uv}$ are not covered by any risk region, then the cost of traversing $\overline{uv}$ is just its usual length; otherwise, due to the potential danger, the cost is amplified in the making of plans.

Another well-developed technique is the A* algorithm [16-19] for searching shortest path in a nonnegative weighted graph embedded with heuristic information. In minefield application, the penalty protocol requires extensive shortest path searches in 2-D grid world. Note that the 2-D grid world has a natural heuristic: the Euclidean distance, hence A* algorithm guided by such heuristic is an ideal choice. If the scale of the grid is large, then some dynamic versions of the A* algorithm (e.g., Focused D* algorithm [20] and D* Lite algorithm [21]) will help accelerating the path replanning. In our experiments, we search shortest paths in a 100×100 grid world and the A* algorithm is already very efficient even if there are a lot of replanings.

We simulate the markers from Beta distribution, which has the probability density function

$$f_{\text{Beta}}(x; A, B) = \begin{cases} \frac{\Gamma(A+B)}{\Gamma(A)\Gamma(B)} x^{t-1}(1-x)^{B-1} & \text{if } 0 < x < 1; \\ 0 & \text{otherwise}, \end{cases}$$

where $A > 0$ and $B > 0$ are the two parameters, $\Gamma$ is Gamma function. In our experiments, for each $i = 1, 2, \cdots, m$, we draw $Y_i|X_i = 0 \sim f_{\text{Beta}}(y; 3.5 - s_n, 3.5 + s_n)$ and $Y_i|X_i = 1 \sim f_{\text{Beta}}(y; 3.5 + s_n, 3.5 - s_n)$, where $0 < s_n \leq 3.5$ is the uniparameter. In fact, such parameter setting renders us a discerning (Beta) sensor for each $s_n$. An illustrative plot of the probability density functions of Beta distribution under $s_n = 1, 2$ is shown in Figure 5.2.

We simulate $m = 100$ disk shape risk regions within $[0, 100] \times [0, 100]$. The locations $d_1, d_2, \cdots, d_{100}$ are independently uniformly drawn from $[10, 90] \times [10, 90]$. For $i = 1, 2, \cdots, 100$, we set $r_i = 4$. Based on this risk region size, we choose $\delta = 1$, which means the resolution of the grid is 100×100. We let $s = (0, 0)$ be the starting location and $t = (100, 100)$ be the target location. This setting is illustrated in Figure 5.3.

For convenience, we assume $c(\overline{uv}) = d \cdot L(\overline{uv})$ for any $\overline{uv}$ such that $Y_{uv} > 0$, where $d$ is the disambiguation cost coefficient. In our experiments, we use $d = 0.5$. It should be noticed that the larger the value of $d$, the more likely the penalty protocol plans deterministic shortest path. The reason that we choose $d = 0.5$ is to see the cases in which disambiguation happens. Finally, for $i = 1, 2, \cdots, 100$, we independently draw $X_i \sim \text{Bernoulli}(0.6)$.

Figure 5.4 and Figure 5.5 show two realizations of RDP. In Figure 5.4, the sensor parameter is set as $s_n = 0.5$. The left plot shows the actual path displayed in real terrain.
The solid circles represent the true hazards; while the dotted circles represent the false detections. The right plot shows the same path displayed in one marked map of the same terrain. In the marked map, the darker, the more dangerous; the lighter, the safer. Observe that a bad marker leads the agent to a dead end and the agent finally escapes, but the resulting path is long. In Figure 5.5, the sensor parameter is set as $s_k = 3$. The real terrain is the same as that displayed in Figure 5.4. Obviously, the actual path displayed in Figure 5.5 is much better. We can see that although the disambiguation cost associated with the second path is greater, the total cost of the second path is less. With the same

Figure 5.4. A realization of random disambiguation path in a real terrain (left) and in one of its marked map (right) under the penalty protocol

Sensor parameter $s_k = 0.5$
Total cost: 204.91; Traveling cost: 202.99; and Disambiguation cost: 1.91

Figure 5.5. A realization of random disambiguation path in the same real terrain (left) and in one of its marked map (right) under the penalty protocol

Sensor parameter $s_k = 3.0$
Total cost: 165.58; Traveling cost: 159.92; and Disambiguation cost: 5.66
Figure 5.6. Frequency statistics of costs under \( s_n = 0.05, 0.5, 1, 1.5, 2, 2.5, 3, 3.45, 3.5 \)
(same terrain; 100 simulated marked maps under each of the 9 sensor parameters)

Figure 5.7. Leftmost \( P_0 \)-quantiles (left plot) and sample mean (right plot) under \( s_n = 0.05, 0.5, 1, 1.5, 2, 2.5, 3, 3.45, 3.5 \)
(same terrain; 100 simulated marked maps under each of the 9 sensor parameters)
terrain, for each $s_n = 0.05, 0.5, 1, 1.5, 2, 2.5, 3, 3.45, 3.5$, we simulate 100 marked maps. The bar plots based on calculating cost frequency are displayed in Figure 5.6. An obvious pattern of those bar plots is that the greater the value of $s_n$, the greater the frequency of small costs. Further statistical analysis is based on calculating the leftmost $P_0$-quantiles and the sample mean under each $s_n$. For each $s_n$, we calculate four leftmost $P_0$-quantiles under $P_0 = 0.95, 0.90, 0.80, 0.70$. The results are graphically displayed in Figure 5.7. The left plot shows $P_0$-monotonicity for the four values of $P_0$; the right plot, which is an error bar plot that uses standard deviations of the sample means, shows weak monotonicity. Furthermore, more calculations show that the $P_0$-monotonicity also holds for many other values of $P_0$. This leads us to conjecture that, for this terrain, our penalty protocol is strongly monotone with respect to markers generated by discerning beta sensor.

We further conjecture that the strong monotonicity holds regardless of the realizations of $X_1, X_2, \ldots, X_{100}$ and the realizations of $d_1, d_2, \ldots, d_{100}$. This bold conjecture is supported by more results from the experiments that involve different terrains. We
Figure 5.9. Leftmost $P_0$-quantiles (four plots above) and sample means (below plot) under $s_n = 0.05, 0.5, 1, 1.5, 2, 2.5, 3, 3.45, 3.5$. Each gray curve represents the experimental results from an individual terrain; each black curve represents the average of the corresponding 30 gray curves (30 different terrains; 100 simulated marked maps for each terrain and under each of the 9 sensor parameters).
randomly draw 30 terrains. For each terrain and each $s_n = 0.05, 0.5, 1, 1.5, 2, 2.5, 3, 3.45, 3.5$, we simulate 100 marked maps. For each run, we calculate the relative cost, that is, the ratio of the final cost to the length of the actual shortest path.

The bar plots in Figure 5.8 again display the pattern that the greater the value of $s_n$, the greater the frequency of small costs. In Figure 5.9, the four leftmost $P_0$-quantiles under $P_0 = 0.95, 0.90, 0.80, 0.70$ for all $s_n$'s show the $P_0$-monotonicity. More calculations show that the $P_0$-monotonicity also holds for many other values of $P_0$. Also in Figure 5.9, the average relative costs show the weak monotonicity.

Based on considerably many experimental results, we believe that our penalty protocol, is not only weakly monotone but also strongly monotone.

6. Summary and Suggestions

Based on some real-world applications, we model the random disambiguation path problem as navigation under the guidance of markers. We don’t assume the parameters of the underlying distribution of true/false states of detections a priori. We assume that there is a global sensor that generates a marker for each detection. The greater the value of the maker, the more likely the detection is true. Under the i.i.d. assumption, we measure the sensor via two conditional distribution functions and henceforth establish the concept of stochastic ordering of sensors. Given a navigation protocol that utilizes the mark information, we want to know whether the markers generated by better sensor yield less cost. This is what the concept of MIM stands for. We have proposed three utility measures of protocol: cost distribution function, leftmost $P_0$-quantile, and mean. In terms of these three measures, we establish the concepts of strong monotonicity, $P_0$-monotonicity, and weak monotonicity.

As one contribution of this work, we have proposed two navigation protocols that utilize markers. One is threshold protocol; the other is penalty protocol. Analytically, we have shown that for convergent graph, both protocols can guarantee the agent to reach the goal; we have shown that the threshold protocol is weakly monotone with respect to makers if the convergent graph is parallel graph; and we have shown that both protocols are strongly monotone with respect to markers if the convergent graph contains only one nondeterministic arc. Another contribution is that we designed and applied a specific penalty protocol to minefield application. Substantial Monte Carlo simulations with discerning Beta sensors have shown both $P_0$-monotonicity and weak monotonicity. Based on the good experimental results, we conjecture that the penalty protocol we designed for minefield model has the strong monotonicity property.

There are many things to do in future research. Analytically, we need to generalize Theorem 7 from parallel graph to more general convergent graph; we need to show whether the penalty protocol is monotone with respect to markers in a convergent graph with more than one nondeterministic arc. Numerically, we need to test our protocol in some different application area (e.g., road network); we need to test the sensors that are modeled by distributions different from Beta. These are just several immediate next projects. In the long run, some interesting problems include designing protocols for specific applications, measuring and comparing the utilities of different protocols, studying the validity and discernability of sensors, and comparing different sensors that are built upon different distributions etc.
Acknowledgement. We greatly thank the supports from Office of Naval Research and Defense Advanced Research Projects Agency. We greatly thank the Center for Imaging Science of Johns Hopkins University for providing computing resources for us to perform large-scale simulations.

References