

# Adaptation of the Simulated Risk Disambiguation Protocol to a Discrete Setting \*

Al Aksakalli, Donniell E. Fishkind, and Carey E. Priebe

Department of Applied Mathematics and Statistics  
Whiting School of Engineering, Johns Hopkins University  
Baltimore, Maryland 21218–2682  
{ala,fishkind,cep}@jhu.edu

## Abstract

Suppose a spatial arrangement of possibly hazardous regions needs to be speedily and safely traversed, and there is a dynamic capability of discovering the true nature of each hazard when in close proximity of it; the traversal may enter the associated region only if it is revealed to be nonhazardous. The problem of identifying an optimal policy for where and when to execute disambiguations so as to minimize the expected length of the traversal can be cast both as a completely observable Markov decision process (MDP) and a partially observable Markov decision process (POMDP) and has been proven intractable in many broad settings. In this manuscript, we adapt the basic strategy of a policy called the *simulated risk disambiguation protocol* of Fishkind *et al.* (2006) to a different, discretized setting (a Canadian Traveller Problem with dependent edge probabilities), and we compare the performance of this adapted policy against the performance of the optimal policy—on a class of instances that are small enough for the optimal policy to be computed. On random such instances, the adapted simulated risk disambiguation protocol performed nearly as well as the optimal protocol, and used significantly less computational resources.

## Overview

Suppose there is a set of (possibly overlapping) *hazard regions*  $H_i \subseteq \mathbf{R}^2$ , for  $i = 1, 2, \dots, N$ , each region marked with the probability  $\rho_i$  that  $H_i$  is a *true hazard* (as opposed to a *false hazard*), and assume that hazard regions are true hazards independently of each other. Now, suppose a *starting point*  $s$  and *destination point*  $d$  are given in  $\mathbf{R}^2$ , and the decision maker's objective is to traverse a shortest continuous curve from  $s$  to  $d$  avoiding all true hazards, i.e. the curve is constrained to  $(\bigcup_{i \in \mathcal{I}} H_i)^C$ , where  $\mathcal{I} \subseteq \{1, 2, \dots, N\}$  is the set of indices of true hazards. While  $\mathcal{I}$  is unknown to the decision maker at the outset (in particular, the probability distribution of  $\mathcal{I}$  is specified by the marks  $\rho_i$ ), when the curve is on the boundary  $\partial H_i$  of any hazard region the

decision maker has the dynamic ability to *disambiguate* the region to discover if  $H_i$  is a false or true hazard and, accordingly, the curve may or may not proceed through  $H_i$ . Each execution of a disambiguation adds a fixed cost  $c \geq 0$  to the length of the  $s$ - $d$  curve traversed, and it may be useful to sometimes assume that there are a fixed limit of  $K \geq 0$  disambiguations that may be performed. How to optimally perform the disambiguations so as to minimize the expected length of the traversal is the *random disambiguation path* (RDP) problem in (Priebe *et al.* 2005).

The RDP problem is a minor modification of the Stochastic Obstacle Scene Problem (SOSP) of (Papadimitriou & Yannakakis 1991), who also describe a discrete version of SOSP that they call the Canadian Traveller Problem (CTP). In CTP, the goal is to minimize the expected traversal length from a starting vertex to a destination vertex in a finite graph whose edges are marked with their respective probabilities of being traversable, and every edge's actual status can be dynamically discovered only when encountered. Papadimitriou and Yannakakis prove the intractability of several variants of SOSP and CTP. (For more on CTP see (Bar-Noy & Schieber 1991)).

CTP is also a special case of the Stochastic Shortest Paths with Recourse (SPR) problem of (Andreatta & Romeo 1988), who present a stochastic dynamic programming formulation for SPR and note its intractability. (Polychronopoulos & Tsitsiklis 1996) also present a stochastic dynamic programming formulation for SPR and they prove the intractability of several variants. (Provan 2003) proves that SPR is intractable even if the underlying graph is directed and acyclic.

In (Fishkind *et al.* 2006), we proposed a class of policies, called *simulated risk disambiguation protocols*, for RDP. In this manuscript, we adapt that basic approach for use on a discretized version of RDP which, in effect, is a Canadian Traveller Problem with dependent arc probabilities. We compare the performance of our adapted policy to the performance of the optimal policy, which we can obtain exactly for relatively small—but nontrivial—instances.

The rest of this manuscript is organized as follows. We next describe a discretization of RDP and mention how it can be cast as a partially or completely observable Markov decision process. Then we adapt the simulated risk disambiguation protocol for use in the discrete setting. Finally,

\*The authors are grateful to the Acheson J. Duncan Fund for the Advancement of Research in Statistics (grant number 06-11) for supporting presentation of this manuscript at ICAPS 2006.

Copyright © 2006, American Association for Artificial Intelligence (www.aaai.org). All rights reserved.

we compare the performance of the adapted simulated risk disambiguation protocol against the optimal policy obtained by a standard implementation of value iteration algorithm on the Markov decision process formulation.

## A Discrete Version of RDP

Because of its continuous setting, the simulated risk disambiguation protocol in (Fishkind *et al.* 2006) is not comparable to the existing heuristics for CTP and SPR, nor are optimal policies readily computable for all but the most trivial instances. With this in mind, we adapt the simulated risk disambiguation protocol in (Fishkind *et al.* 2006) to a related, discrete setting in which the protocol adaptation’s performance may be compared to the optimal policy for relatively small but nontrivial instances.

The discretization of RDP we will consider here is, for simplicity and convenience, a subgraph of the integer lattice  $\mathbf{Z}^2$ . Specifically, it is the graph  $G$  whose vertices are all of the pairs of integers  $x, y$  such that  $1 \leq x \leq x_{\max}$  and  $1 \leq y \leq y_{\max}$ , where  $x_{\max}$  and  $y_{\max}$  are given integers. There are edges between all pairs of vertices of the form  $x, y$  and  $x + 1, y$ , and there are edges between all pairs of vertices of the form  $x, y$  and  $x, y + 1$ . The hazard regions  $H_i$ , for  $i = 1, 2, \dots, N$ , are open discs in  $\mathbf{R}^2$ . (See Figure 1 and Figure 2.) One vertex in  $G$  is designated as the starting point  $s$ , another vertex in  $G$  is designated as the destination point  $d$ , and the decision maker is to walk from  $s$  to  $d$  in  $G$ , only traversing edges that do not intersect any true hazards. If an edge intersects any ambiguous hazard region, then a disambiguation of the hazard region may be performed from either of the edge’s endpoints that is outside of the hazard region. As before, the goal is to develop a policy for traversing and performing disambiguations that minimizes the expected length of the traversal. We call our problem *discrete RDP* (DRDP).

## Markov Decision Process Formulation of DRDP

We next describe how the DRDP problem can be modelled as a partially or completely observable Markov decision process, in a manner similar to that which is done in (Blei & Kaelbling 1999).

An *information vector*  $I \in \{\text{“a”}, \text{“t”}, \text{“f”}\}^N$  keeps track of the decision maker’s current knowledge of the hazard regions’ status; specifically, for all  $i = 1, 2, \dots, N$ , the  $i$ th entry of  $I$  is “a”, “t”, or “f” according as  $H_i$  is currently ambiguous, true, or a false hazard. Let  $V$  denote the set of endpoints of edges in  $G$  that intersect any boundary of any hazard region; in particular, these are the vertices of  $G$  at which the decision maker may execute disambiguations. Now, the set of states is  $(V \times \{\text{“a”}, \text{“t”}, \text{“f”}\}^N) \cup \{s, d\}$ , which represent possible locations on the lattice at which the decision maker may be at a particular moment in time, coupled with the possible information vectors that may describe the decision maker’s knowledge at that moment. The set of actions is the set of ordered pairs  $(v, i)$  such that vertex  $v \in V$  is the endpoint of an edge which intersects  $\partial H_i$ ; this pair represents where the next disambiguation is to occur, and which

hazard region will be disambiguated.

The reward for any appropriate action at any particular state is the negative of the shortest path distance (avoiding all ambiguous and true hazards indicated by the information vector of the state) between the vertex identified in the state and the vertex identified in the action; also subtract the disambiguation cost  $c$  if the vertex identified in the action is not  $d$ . The destination  $d$  is an absorptive state for which there is a one-time and very large reward for entering. Given a state and action, the state transitioned into is the vertex identified in the action and the information vector of the previous state, updated to indicate that the hazard  $H_i$  identified in the action is true or false with respective probabilities  $\rho_i$  and  $1 - \rho_i$ .

The above set of states, actions, rewards, and transition distributions comprise a Markov decision process with  $K$  stages (or  $N$  stages if there is no limit  $K$  on the number of allowed disambiguations). In the rest of this paper, we will compute optimal policies in this formulation using the standard stochastic dynamic programming technique of value iteration for relatively small but nontrivial instances. Let  $p^*$  denote the  $s, d$ -curve traversed by the optimal policy; since the trajectory under the optimal policy is still random,  $p^*$  is an  $(s, d)$ -walk-valued random variable, and we denote its expected length  $E(p^*)$ .

Also, DRDP may be cast as a partially observable Markov decision process by trimming the set of information vectors to be just  $\{\text{“t”}, \text{“f”}\}^N$ , and by folding the ambiguity of hazards into ambiguity of the information vector, hence the partial observability of the state.

## Adapting the Simulated Risk Disambiguation Protocol

We next introduce the adaptation of the simulated risk disambiguation protocol. In our framework, the traversal never enters hazard regions while they are still ambiguous or are known to be true hazards. The paradigm of the simulated risk disambiguation protocol is—for the sole purpose of deciding where to disambiguate next—to temporarily pretend (*simulate*) that the ambiguous hazards are riskily traversable.

Under this simulation of risk, for any  $s, d$  walk  $W$ , the Euclidean length of  $W$ ,  $\ell^{\mathcal{E}}(W)$ , is the number of edges in  $W$  (since each edge in our lattice clearly has Euclidean length 1). The *risk length* of  $W$  is defined as

$$\ell^{\mathcal{R}}(W) := -\log \prod_{H_i: H_i \cap W \neq \emptyset} (1 - \rho_i).$$

This negative logarithm of the probability that  $W$  is permissibly traversable is a measure of the risk in traversing  $W$ —if you were willing to take on risk. An *undesirability function* is any function  $g : \mathbf{R}_{\geq 0} \times \mathbf{R}_{\geq 0} \rightarrow \mathbf{R}$  which is monotonically nondecreasing in its arguments; that is to say, for all  $r_1, r_2, t_1, t_2 \in \mathbf{R}_{\geq 0}$  such that  $r_1 \leq r_2$  and  $t_1 \leq t_2$ , it holds that  $g(r_1, t_1) \leq g(r_2, t_2)$ . The number  $g(\ell^{\mathcal{E}}(W), \ell^{\mathcal{R}}(W))$  is thought of as a measure of the undesirability of  $W$  in the sense that, if you were required to traverse from  $s$  to  $d$  in  $G$  under the simulation of risk and without a disambiguation capability, you would select the walk

$$\phi_g := \arg \min_{s, d \text{ walks } W} g(\ell^{\mathcal{E}}(W), \ell^{\mathcal{R}}(W)).$$

The *adapted simulated risk disambiguation protocol* associated with  $g$  would have the decision maker traverse  $\phi_g$  from  $s$  until its first ambiguous edge, say,  $e$  is encountered at, say, vertex  $v$ , and say  $e$  intersects region  $H_i$ . At this point (since we may not take on risk in actuality) disambiguate the hazard region  $H_i$  and repeat this whole procedure using  $v$  as the new  $s$ , and then either removing  $H_i$  or setting  $\rho_i := 1$  according as  $H_i$  was just discovered to be a false or true hazard. (If at some point the limit  $K$  on the number of disambiguations has been reached, then the shortest unambiguously permissible path is then taken to  $d$ .)

The simplest undesirability functions are the linear ones, where  $g(r, t) := r + \alpha \cdot t$  for some given constant  $\alpha > 0$ , and it is to these undesirability functions that we restrict our attention in the remainder of this manuscript. To find  $\phi_g$  in this particular case, we just need to find a deterministic shortest  $s, d$  path in  $G$  (via Dijkstra's algorithm, say) where each edge in  $G$  is weighted<sup>1</sup> with

$$w(e) := 1 - \alpha \cdot \frac{1}{2} \sum_{i=1}^N \# \text{comp}(e \setminus H_i) \cdot \mathbf{1}_{e \cap H_i \neq \emptyset} \cdot \log(1 - \rho_i),$$

where  $\mathbf{1}$  is the indicator function (taking value 1 or 0 according as its subscripted expression is true or false) and  $\text{comp}(\cdot)$  is the number of connected components of its argument. See illustration in Figure 1 with  $N = 4$ . Corresponding edge weights are shown in Table 1.

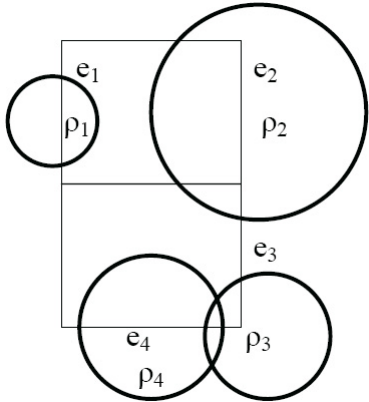


Figure 1: Illustration with  $N = 4$

Edge	Edge weight
$e_1$	$1 - \alpha \log(1 - \rho_1)$
$e_2$	1
$e_3$	$1 - \alpha(1/2)(\log(1 - \rho_2) + \log(1 - \rho_3))$
$e_4$	$1 - \alpha(1/2)(\log(1 - \rho_3) + 2\log(1 - \rho_4))$

Table 1: Weights of edges in Figure 1

<sup>1</sup>We are assuming  $s$  and  $d$  are not inside any hazard region, and that  $\phi_g$  never revisits a hazard region.

For a fixed  $\alpha > 0$ , denote by  $p_\alpha$  the  $s, d$  walk traversed under the adapted simulated risk disambiguation protocol;  $p_\alpha$  is an  $s, d$ -walk-valued random variable, since its realization depends on the outcomes of the dictated disambiguations. We will denote by  $Ep_\alpha$  the expected length of this walk.

## Computational Experiments

In this section, we evaluate the performance of the adapted simulated risk disambiguation protocols against the optimal policy, the latter obtained by a standard implementation of value iteration algorithm on the Markov decision process model described previously.

For all of the experiments in this Section, the lattice used is  $x_{\max} = 40$  by  $y_{\max} = 20$ , with  $s = (20, 20)$  and  $d = (20, 1)$ . Each hazard region is a disc with radius 5.5 units, and the disc's centers are sampled from a uniform distribution on the pairs of integers in  $[7, 34] \times [7, 14]$ ; in particular, this ensures that there is always a permissible path from  $s$  to  $d$ . Probabilities  $\rho_i$  of the hazard regions being true are sampled from a uniform distribution on  $[0, 1]$ . The cost of disambiguation is taken here as  $c = 1.5$ . The environment is illustrated in Figure 2 with  $N = 7, K = 1$ .

For the instance shown in Figure 2, the adapted simulated risk disambiguation protocol using  $\alpha = 1$  calls for traversing to  $(20, 15)$ , at which point the hazard region  $H_4$  centered at  $(19, 9)$  is disambiguated. In case the hazard turns out to be false, the decision maker traverses directly to the destination at a total cost of 20.5. Otherwise, the decision maker traverses to the destination avoiding all the hazard regions, at a total cost of 50.5. (Both walks are illustrated in black on Figure 2.) Here, we had  $\rho_4 = .2230$  thus, in particular, the expected length of the  $s, d$ -traversal is  $(1 - .2230)(20.5) + (.2230)(50.5) = 27.19$ . This protocol turns out to be the overall optimal policy.

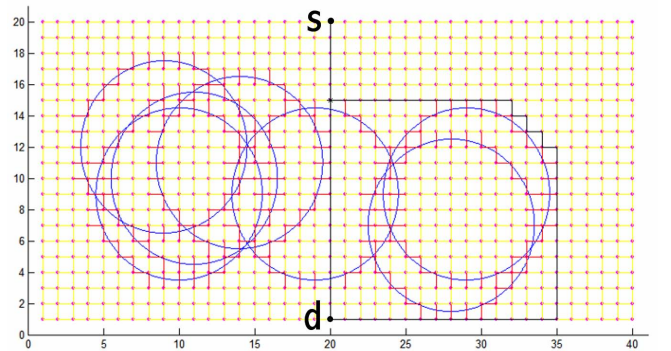


Figure 2: An experimental realization with  $N = 7, K = 1$

Similar to what was done in (Fishkind *et al.* 2006), values of  $\alpha$  minimizing  $Ep_\alpha$  are computed numerically by evaluating  $Ep_\alpha$  for a mesh of values of  $\alpha$ —starting at  $\alpha = 1$ , incrementing  $\alpha$  successively by 5 units until  $\alpha$  is large enough that no disambiguations are performed. We will now denote  $\hat{p}^* := p_{\alpha^*}$ , where  $\alpha^*$  is the value of  $\alpha$  minimizing  $Ep_\alpha$ .

Table 2 compares  $E\hat{p}^*$  (the expected length of the best adapted simulated risk disambiguation protocol) to  $Ep^*$  (the

expected length of the overall optimal policy) for 50 experiment realizations under each  $N, K$  combination listed. The second column shows the percentage of simulations where  $E\hat{p}^* = Ep^*$ , in which case the family of adapted simulated risk disambiguation protocols contains the overall optimal policy. The next column shows the percentage of the simulations where the optimal policy was to perform no disambiguations. The next column shows the mean of the relative errors  $\frac{E\hat{p}^* - Ep^*}{Ep^*}$  for the 50 experiment realizations in each  $N, K$ .

	% where $E\hat{p}^* = Ep^*$	% where $p^*$ didn't disambig.	mean relative errors $\frac{E\hat{p}^* - Ep^*}{Ep^*}$	VI exec. time
$K = 1 :$				
$N = 4$	86	28	.0097	12 sec
$N = 5$	86	30	.0076	15 sec
$N = 6$	72	22	.0103	20 sec
$N = 7$	66	22	.0114	33 sec
$N = 8$	66	36	.0159	48 sec
$K = 2 :$				
$N = 4$	78	30	.0058	4 min
$N = 5$	66	22	.0065	6 min

Table 2: Comparison of  $\hat{p}^*$  to  $p^*$ ; for  $K = 1, N = 4$  the overall optimal policy was indeed an adapted simulated risk disambiguation protocol in 86% of the experiments.

As Table 2 indicates, solutions found by the simulated risk protocol are quite comparable to the optimal solutions. Among all the simulations we performed, the simulated risk protocol found the optimal solution 74.3% of the time and the mean relative error of these simulations was .0096.

The last column in Table 2 shows the execution time for a single instance of the value iteration algorithm used to identify an overall optimal policy and to compute  $Ep^*$ , on a 3.5 gigahertz personal computer with 1 gigabyte memory. We observed that value iteration required significant computational resources even for small instances. In fact, value iteration execution time was over an hour for  $N = 7, K = 3$  and, for  $N = 10, K = 1$ , value iteration did not run at all due to insufficient memory. This was in sharp contrast to the identification of  $\hat{p}^*$  and the computation of  $E\hat{p}^*$ , which took a negligible amount of time in all of these experiments. Computing  $E\hat{p}^*$  continued to take a negligible amount of time for much, much larger values of  $N$  and  $K$ .

## Summary and Conclusions

In this manuscript, we adapted the simulated risk disambiguation protocol from the RDP setting to the discrete RDP setting, which is essentially a Canadian Traveller Problem with dependent edge probabilities. We cast the problem as a Markov decision process and, via value iteration, obtained optimal policies for relatively small but nontrivial instances. Against these optimal policies, we compared the performance of adapted simulated risk disambiguation protocols and discovered that much of the time the adapted simulated risk disambiguation protocols were indeed optimal and, in general, compared very favorably. Furthermore, negligible computing time was needed—compared to that expended on the computation of the optimal policy.

## References

- Andreatta, G., and Romeo, L. 1988. Stochastic shortest paths with recourse. *Networks* 18(3):193–204.
- Bar-Noy, A., and Schieber, B. 1991. The canadian traveller problem. *SODA '91: Proceedings of the Second Annual ACM-SIAM Symposium on Discrete algorithms*.
- Blei, D. M., and Kaelbling, L. 1999. Shortest paths in a dynamic uncertain domain. *IJCAI Workshop on Adaptive Spatial Representations of Dynamic Environments*.
- Fishkind, D.; Priebe, C.; Giles, K.; Smith, L.; and Aksakalli, V. 2006. Disambiguation protocols based on risk simulation. *IEEE Transactions on Systems, Man, and Cybernetics Part A*:to appear.
- Papadimitriou, C., and Yannakakis, M. 1991. Shortest paths without a map. *Theoretical Computer Science* 84:127–150.
- Polychronopoulos, G. H., and Tsitsiklis, J. N. 1996. Stochastic shortest path problems with recourse. *Networks* 27(2):133–143.
- Priebe, C.; Fishkind, D.; Abrams, L.; and Piatko, C. 2005. Random disambiguation paths for traversing a mapped hazard field. *Naval Research Logistics* 52:285–292.
- Provan, J. S. 2003. A polynomial-time algorithm to find shortest paths with recourse. *Networks* 41(2):115–125.