The Blindfolded Robot : A Bayesian Approach to Planning with Contact Feedback

Brad Saund[1], Sanjiban Choudhury[2], Siddhartha Srinivasa[2], and Dmitry Berenson[1]

[1] University of Michigan, {bsaund, dmitryb}@umich.edu,
[2] University of Washington, {sanjibac, siddh}@cs.uw.edu

Abstract. We address the problem of robot motion planning under uncertainty where the only observations are through contact with the environment. Such problems are typically solved by planning optimistically assuming unknown space is free, moving along the planned path and re-planning if the robot collides. However this approach can lead to many unnecessary collisions and movements. We propose a new formulation, the Blindfolded Traveler's Problem (BTP), for planning on a graph containing edges with unknown validity, with true validity observed only through attempted traversal by the robot. We prove that BTP is NP-complete and present a number of approximation-based policies. In particular, we analyze the case of a robot arm where it is challenging to construct a reasonable prior over obstacles. We examine a number of belief approximation techniques and finally propose a policy-belief combination. For the policy we propose graph search with edges weights augmented by the probability of collision. For the belief representation we propose a weighted Mixture of Experts of Collision Hypothesis Sets and a Manifold Particle Filter. Empirical evaluation in simulation and on a real robot arm shows that our proposed approach vastly outperforms several baselines as well as a previous approach that does not employ the BTP framework.

1 Introduction

We examine the problem of robot motion planning in partially-known environments where obstacles are sensed only through contact. This problem occurs quite frequently in manipulation tasks with sensing limitations such as a narrow field of view, occlusions in the environment, lack of ambient light, or insufficient sensor precision. For example, a robot may reach into dark confined areas during maintenance and assembly (e.g. inspecting the insides of aircraft [1]) or during everyday household tasks (e.g. reaching deep into a cabinet or behind a box [2]). Here, the goal is to minimize the total time it takes for the robot to move around obstacles sensed on-the-fly and reach a target configuration.

Consider the scenario where a robot arm is tasked with reaching into a box whose location is uncertain (Fig. 1). This could be framed as a POMDP, where the belief over

This research was funded in part by NSF under grant IIS-1750489 and by Toyota Research Institute (TRI). This article solely reflects the opinions of its authors and not TRI or any other Toyota entity. This work was also (partially) funded by the National Institute of Health R01 (#R01EB019335), National Science Foundation CPS (#1544797), National Science Foundation NRI (#1637748), the Office of Naval Research, the RCTA, Amazon, and Honda Research Institute USA.

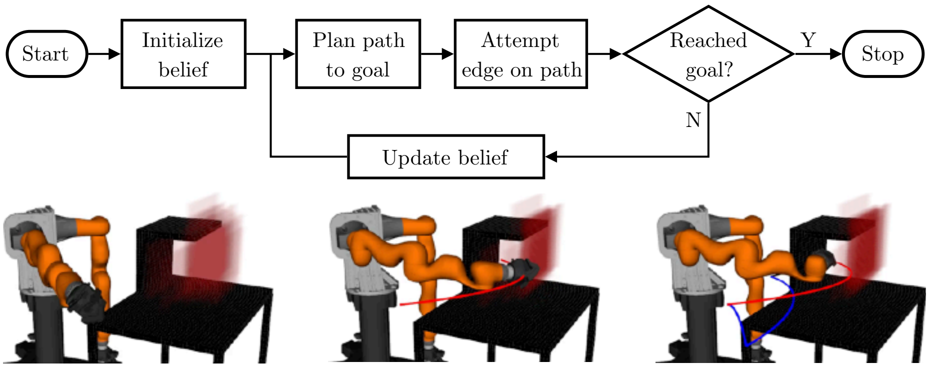


Fig. 1: Overview of the BTP framework for planning with contact feedback. The robot is uncertain about location of the back wall. As it attempts to traverse edges, it partially localizes the wall and eventually finds its way to the goal.

occupancy is obtained through noisy collision measurements. However the possible states of the POMDP include all possible arrangements of obstacles, and the action space includes all possible motions. The general POMDP is thus intractably large.

Instead, such planning problems may be solved by constructing a graph [3], where vertices represent robot configurations and edges represent potentially-valid movements of the robot between these configurations. Here, the validity of edges is unknown *a priori*. A natural strategy is *Optimism in the Face of Uncertainty* (OFU) [4] — assume untraversed edges are valid, plan the shortest path and execute it. If the shortest path is indeed valid, the robot reaches the goal optimally. Otherwise, it removes the invalid edge from the graph and replans. OFU is effective in less-cluttered environments, where the robot finds a path to the goal after a few collisions. However, on problems with narrow passages such as Fig. 1, OFU can lead the robot down a “rabbit hole” trying paths that are not likely to be valid.

Our key insight is that *the validity of edges in the graph is correlated*. There are two main reasons for this correlation. First, edges overlap in swept workspace volume. Second, objects in the world occupy multiple workspace cells. Given a prior on edges, a robot can exploit such correlations to infer edge validities and reach the goal quickly (Fig. 1). We address the following research question:

> How should a robot navigate on a graph with unknown edge validites to minimize the expected traversal cost?

We refer to this broader problem as the *Blindfolded Traveler’s Problem* (BTP). We show that this problem is NP-Complete and discuss a set of approximation-based policies. We also propose a new policy, Collision Measure, that is both efficient to compute and has theoretical guarantees.

We formulate robot arm planning with contact feedback as a BTP. We face an additional challenge for realistic scenarios – *the initial belief is approximate and can be misleading*. With a good initialization we show a particle filter that updates hypothesis worlds from contact observations suffices. Without a good initialization, we show an algorithm that starts with free-space and builds up a world model consistent with observations is effective. Since both scenarios occur in practice, we propose a Mixture of Experts framework for mixing these two belief update strategies.

In summary, this paper makes the following contributions:

- Formulate the *Blindfolded Traveler's Problem*. (Section 3)
- Map the planning with contact feedback task to a BTP. Since the posterior is not specified, we propose a set of belief approximation strategies. (Section 4)
- Propose a set of approximation strategies to solve the BTP. (Section 5)
- Provide empirical evaluation of different strategies and belief approximations on simulated and real robot arm BTP instances. (Section 6)

We evaluate all strategies on two 7 DOF robot arm planning scenarios in simulation, each with three varying levels of difficulty (by adding error in prior). We also evaluate strategies with practical computation times on a real robot arm. We find that the Collision Measure strategy using a Mixture of Experts belief tends to outperform all other baselines by planning consistently low cost paths with consistently low computation time. Furthermore, we find using the BTP framework significantly outperforms a baseline strategy used in planning with contact feedback.

2 Related Work

We examine planning under contact sensing uncertainty which leads to a number of challenges. While some approaches consider tactile skin [5], with only torque feedback contact observations cannot precisely localize collision points. One approach is to use non-parametric particle filters, however, they encounter problems with contact measurements [6]. The Manifold Particle Filter overcomes this by sampling from different proposal distributions depending if contact/no contact [7], though this method requires an accurate prior over obstacles. Without a prior over obstacles we use the Collision Hypothesis Set belief, which we have previously employed in search using RRT [8].

Our problem is closely related to that of real-time motion planning on roadmaps [3]. Roadmaps, which are graphs in configuration space, are efficient because they can be reused across planning iterations. In robot motion planning, edge evaluation dominates computational complexity [9], therefore the key to minimizing search times is laziness [10, 11]. LazySP [12], shown to be optimally lazy [13], optimistically plans the shortest path and checks edges sequentially till an infeasible edge is encountered. Priors on edge validities can be further exploited to minimize edge evaluation [14–16]. These problems can be further mapped to Bayesian active learning [17–19] to compute policies that actively choose edges to evaluate to minimize uncertainty about which path is feasible [20, 21]. An alternate formulation is online shortest path routing [22–24] which is a particular instance of combinatorial bandits [25]. However, unlike our problem, these methods have full flexibility to telelport to and evaluate any edge.

Our work falls under the domain of planning under sensing uncertainty. D* [4] and variants [26, 27] typically replan optimistically and re-using the search graph. An alternative is to cast the problem in a Bayesian paradigm using an occupancy map [28].

However, such methods usually plan to short horizons. Since this problem arises from the mobile robot community, the focus is primarily robot safety [29]. For our problem, the robot is able to collide safely and we seek to minimize the travel cost.

The BTP problem is closely related to the Canadian Traveler's Problem (CTP) [30] where neighboring edge costs are revealed when an agent visits a vertex. DAGs can be solved exactly via DP [31] but the general problem is PSPACE-complete [32]. Typically CTPs are solved using heuristics [33] adopted from probabilistic planning [34] or using Monte-carlo Tree Search [35, 36]. CTP can also be cast in a Bayesian framework [37] and solved near-optimally using informative path planning techniques [38, 39]. While we evaluate some of these strategies for our robot arm planning, others are prohibitively expensive due to expensive collision checking and posterior update. We therefore adapt the Collision Measure [14] as a computationally efficient strategy for the CTP/BTP.

3 Problem Statement

We propose the Blindfolded Traveler's Problem as a graph search problem to model the contact feedback planning problem. In a BTP the traveler traverses a graph attempting to reach a goal. While traversing an edge the traveler may encounter a blockage and be forced to retrace back to the previous node and plan an alternate route. While the traveler only directly senses the validity of the attempted edge, blockages may be correlated, thus providing implicit information about the validity of other edges in the graph.

3.1 Blindfolded Traveler's Problem

Let $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathcal{W})$ be an explicit directed graph where $\mathcal{V}$ denotes the set of vertices, $\mathcal{E}$ denotes the set of edges and $\mathcal{W} : \mathcal{E} \rightarrow \mathbb{R}_{\geq 0}$ denotes the weight of each edge. For each edge $e \in \mathcal{E}$, let $x(e) = \{0, 1\}$ denote if the edge is invalid (0) or valid (1). Note that $x(e)$ is *latent*. Additionally, let $\eta(e) \in [0, 1]$ be the *latent* blockage of an edge. The blockage is the fraction of an edge that can be traversed before encountering an obstruction.

A traveler located at vertex v_1 may attempt to traverse any edge $e_{1,2}$ connecting a neighboring vertex v_2. An attempt $(v_1, e_{1,2})$ is mapped to a resultant vertex and traversal cost specified by the following function:

$$\Gamma(v_1, e_{1,2}, x, \eta) = \begin{cases} (v_2, w(e_{1,2})) & x(e) = 1 \\ (v_1, 2\eta(e_{1,2})w(e_{1,2})) & x(e) = 0 \end{cases} \tag{1}$$

Traversing a valid edge moves the traveler to the new vertex v_2 with a traversal cost equal to the weight of the edge $w_{e_{1,2}}$. Traversing an invalid edge returns the traveler to the original vertex v_1 with a traversal cost equal to the distance travelled to the blocked point and back, $2\eta(e_{1,2})w(e_{1,2})$.

The traveler has a prior $\mathcal{P}$ on the joint probability $P(x, \eta)$. When attempting to traverse edge e, the traveler receives the observation $o = (x(e), \eta(e))$. The traveler maintains a history of all observations, i.e. $\psi_T = \{o_t\}_{t=1}^{T}$. The Blindfolded Traveler's Problem can be fully specified by the tuple $\langle \mathcal{G}, \mathcal{P}, v_s, v_g \rangle$ where $v_s, v_g \in \mathcal{V}$ are the initial and goal vertices.

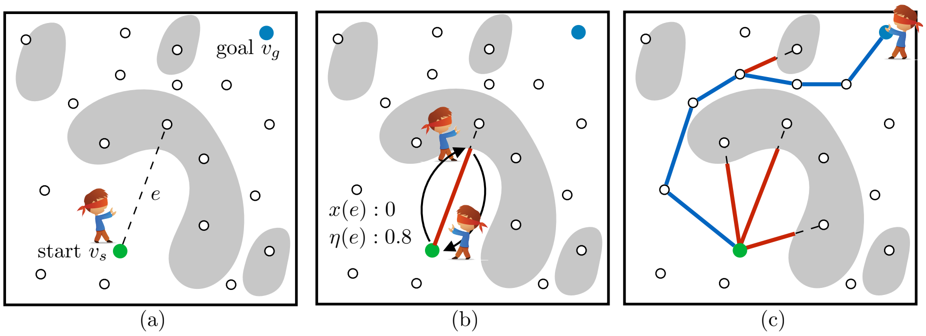


Fig. 2: Blindfolded Traveler's Problem

The solution to the BTP can be represented as a policy tree π, where the nodes specify an edge e that the traveler attempts to traverse. Each branch is labelled by an observation o_e. The root node of the tree is an edge emanating from start vertex v_s. To follow the policy, the traveler attempts to traverse the edge e and takes the branch matching the observation o_e to go to a next edge e'. The procedure repeats until the traveler reaches a terminal node which is always the edge e_{v_g,v_g}, i.e., a loop at the goal vertex.

The cost of a policy for a given (x, η), $c(\pi(x, \eta))$ is the sum of traversal costs. The goal of the traveler is to minimize the expected cost

$$\min_{\pi} \mathbb{E}_{(x,\eta)\sim\mathcal{P}} \left[c(\pi(x, \eta))\right] \tag{2}$$

We show that BTP is NP-complete. We do so by constructing a mapping between any Optimal Decision Tree (ODT) problem, where the goal is to find a hypothesis with minimum tests, to an equivalent BTP. Since ODT is NP-Complete, so is BTP. For the proof and further details refer to supplementary materials [40].

3.2 Contact-based Planning Problem as an instance of BTP

We now examine the problem of a robot arm planning with unknown workspace obstacles sensed only through contact and map this problem to an instance of BTP.

The robot's configuration space $\mathcal{C}$ is composed of free space $\mathcal{C}_{free}$ and obstacles $\mathcal{C}_{obs} = \mathcal{C} \setminus \mathcal{C}_{free}$. The robot operates in a workspace W containing workspace obstacles W_{obs}. A robot configuration $q \in \mathcal{C}$ occupies a workspace volume $\mathcal{R}(q) \subset W$. We say q is *in collision* if $\mathcal{R}(q) \cap W_{obs} \neq \emptyset$.

The graph $\mathcal{G}$ is a roadmap where vertices $\mathcal{V}$ are configurations and edges $\mathcal{E} : [0, 1] \to \mathcal{C}$ are paths through $\mathcal{C}$ connecting vertices, with $w(e) = ||e(0) - e(1)||$. An edge therefore represents the swept volume $W_e = \cup_{d\in[0,1]} \mathcal{R}(e(d))$. The prior $\mathcal{P}$ is a probability density over W_{obs}. This is mapped to $\mathcal{C}$ via $\mathcal{R}(\cdot)$ thus inducing a joint probability $P(x, \eta)$.

We consider a robot that senses obstacles indirectly though collision using measured joint torque $\tau^{meas} \in \mathbb{R}^J$, where J is the number of robot joints. Using a mass model of the robot the expected joint torque due to gravity and dynamics τ^{exp} is calculated and used to estimate the external joint torque $\tau^{ext} = \tau^{meas} - \tau^{exp}$. A noise threshold τ^{th} is set for each joint and τ^{ext} triggers a collision observation at q_{col} whenever any joint exceeds its threshold. A successful edge traversal results in $o = (1, 1)$, while a collision yields $o = (0, \eta)$ where $e(\eta) = q_{col}$.

Furthermore, as a slight augmentation of BTP, a collision yields additional information. Joint i exceeding τ_i^{th} implies an external (contact) force on a link after joint i on the kinematic chain. A set of links $\mathcal{L}_{contact}$ that must contain a contact is constructed by first finding the largest i where $\tau_i^{ext} > \tau_i^{th}$, then adding all links downstream from joint i to $\mathcal{L}_{contact}$. Define $\mathcal{R}(q, \mathcal{L}) \subseteq \mathcal{R}(q)$ as the workspace occupancy for only links $\mathcal{L}$. A traveler may use the knowledge that an object must be in contact with $\mathcal{R}(q, \mathcal{L}_{contact})$, as opposed to anywhere on $\mathcal{R}(q)$.

The BTP for contact planning has a few defining characteristics that warrant attention. First, the edges of this BTP are highly correlated, because a single workspace obstacle can block multiple C-space edges. Hence even an independent prior over workspace occupancy translates to correlation amongst edges. The robot exploits this to gain information about untraversed edges. Second, it's unclear how one obtains priors. A uniform random distribution is certainly not realistic. A finite dataset of worlds has realizability issues on account of continuous observations. Designing parametric distributions that capture all likely worlds is difficult. Finally, a manually-specified prior might be inaccurate. How should the robot detect and compensate for this in a principled manner? We propose solutions that deal with these issues in the next section.

4 Belief Representations for Contact-based Planning

An agent maintains a belief over workspace occupancy W_{obs}, which we refer to as a world $\phi \in \Phi$ and represent it using a voxel grid. The belief at timestep t is represented as $b_t(\phi)$. Since each voxel can be either occupied or free, the set of worlds is $\Phi = \{0, 1\}^N$ where N is the number of voxels, thus explicitly enumerating all possible worlds is infeasible. We follow two approaches for maintaining the belief. The first is a non-parametric particle filter where a set of candidate hypotheses are maintained and possibly ruled out. The second is an approach that adds new hypotheses that are consistent with measurements. We also motivate and discuss mixing these methods.

Approach 1: Manifold Particle Filter (MPF) A particle filter is a non-parametric Bayes filter that represents belief $b_t(\phi)$ as a finite set of possible candidate worlds $\Phi_t = \{\phi_t^1, \phi_t^2, \ldots\}$ with associated weights $\{\mu_t^1, \mu_t^2, \ldots\}$. In this paper, the particles model objects with known geometry but with varying positions. Since in the BTP objects are stationary, the process model is static, and particles are only updated due to the measurement model, thus we only update the particle weights and do not resample.

A known issue with particle filters is poor performance when the proposal distribution does not match the target distribution. A conventional particle filter performs measurement updates via importance sampling: sampling from $\phi_{t-1}^i \sim b_{t-1}$ and weighing by $\mu_t^i = P(o_t | \phi_t^i)$. In the case of a highly discriminative measurement such as a con-

Algorithm 1: Manifold Particle Filter

```
input  : particles $\Phi_{t-1}, e, o_t = (x_t, \eta_t), \mathcal{L}_{contact}$
output: particles $\Phi_t$
1  $\Phi_t \leftarrow \emptyset$
2  for $\phi^i_{t-1} \in \Phi_{t-1}$ do
3      for $d \in [0, \eta_t)$ do
4          $q = e(d)$
5          $\phi^i_t \leftarrow \phi^i_{t-1}$
6          $\mu^i_t \leftarrow P(\mathcal{R}(q) \cap W_{obs} = \emptyset | \phi_t) \mu^i_{t-1}$
7      if $x_t = 0$ then
8          $W_{CM} \leftarrow \mathcal{R}(e(\eta_t), \mathcal{L}_{contact})$)
9          $\phi^i_t \leftarrow$ PROJECT ($\phi^i_{t-1}, W_{CM}$)
10         $\mu^i_t \leftarrow$ KERNELDENSITYESTIMATE ($\Phi_{t-1}, \phi^i_t$)
```

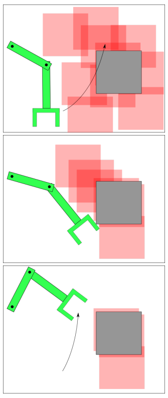


Fig. 3: Manifold Particle Filter: The initial particles Φ_0 model configurations of the true obstacle before the robot moves (top). A collision during a motion causes particles to be resampled on the contact manifold (middle). Subsequent free space motions sweep through and eliminate some particles (bottom).

tact, the target distribution represents a thin manifold of possible object configurations which does not match the proposal b_{t-1}, causing particle starvation.

We therefore adopt the strategy used in the Manifold Particle Filter (MPF) [7], depicted in Fig. 3 and detailed in Algorithm 1. For robot motions through free space where no collision is observed the MPF updates using importance sampling as in a conventional particle filter (Line 6). With our static process model this is equivalent to eliminating particles inconsistent with the new known free space.

When a collision is observed the MPF instead uses the contact manifold as the proposal distribution, sampling particles from obstacle configurations in contact with the robot arm (Line 10). The importance weights are then calculated using $P(\phi^i_t | b^i_{t-1})$. b^i_{t-1} is approximated by applying a Gaussian kernel to Φ_{t-1}, called a Kernel Density Estimate. We implement the Implicit Manifold Particle Filter [7] which approximates the proposal distribution by projecting the prior particles onto the contact manifold. Though computationally efficient, this projection does introduce significant bias, as the previous estimate appears both in the sampling and the re-weighting. In our implementation we translate each particle the minimum distance so that it overlaps with the robot in the collision configuration. This choice of projection can generate new particles that are inconsistent with past contact observations. While a more sophisticated projection operation is of interest, it is beyond the scope of this work.

MPF performs well when given an accurate initialization b_0, but for robots in the real world it is often unrealistic to assume the distribution over obstacles is known accurately. One such instance is when b_0 clusters the correct object far from the correct position. Another common and more difficult instance is when the particles model the incorrect object geometry, so no particle is capable of representing the true world.

```
Algorithm 2: Collision Hypothesis Set
    input : CHSs $\mathcal{K}$, Known Freespace $W_F$,
            $e, o_t = (x_t, \eta_t), \mathcal{L}_{contact}$
    output: $\mathcal{K}, W_F$
1   for $d \in [0, \eta_t)$ do
2       $q = e(d)$
3       $W_F \leftarrow W_F \cup \mathcal{R}(q)$
4   if $x_t = 0$ then
5       $\mathcal{K}$.append($\mathcal{R}(e(\eta)), \mathcal{L}_{contact}$))
6   for $\kappa_i \in \mathcal{K}$ do
7       $\kappa_i \leftarrow \kappa_i \setminus W_F$
```

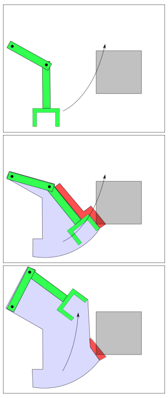


Fig. 4: CHS: The robot initially plans a motion optimistic about unknown space (top). A motion sweeps out free space (blue) and a collision generates a CHS (middle). Future free space motion sweeps out more free space, potentially shrinking CHSs (bottom).

Approach 2: Collision Hypothesis Sets (CHS) To overcome the reliance on an accurate prior we can adopt the Collision Hypothesis Set (CHS) [8] belief. A single CHS $\kappa_i \in W$ is the complete set of voxels that could explain observed collision i. The CHS belief builds up a set $\mathcal{K} = \{\kappa_1, \kappa_2, \dots\}$ to explain all measurements.

Fig. 4 depicts the CHS update described in Algorithm 2. As the robot moves without collision, the swept volume of the motion is marked as known free space in the voxel grid (Line 3). When a collision is encountered during robot motion a CHS is added containing voxels of the links possibly in collision (Line 5). The known free space is then removed from all CHSs (Line 7).

$\mathcal{K}$ induce a belief $P(x)$ as follows:

$$P(x(e) = 0|\kappa_i) = \frac{|W_e \cap \kappa_i|}{|\kappa_i|} \qquad \text{effect of single CHS} \qquad (3)$$

$$P(x(e) = 1|\mathcal{K}) = \prod_i 1 - P(x(e) = 0|\kappa_i) \qquad \text{effect of all CHSs} \qquad (4)$$

where (3) captures the optimistic assumption that each κ generates exactly one occupied voxel, and (4) comes from the assumption that each κ is independent. Note that the CHS method never mark a valid edge as invalid. $P(x(e) = 1) = 0$ (i.e. e is marked invalid) only if W_e completely contains a κ. By construction a κ must contain an occupied voxel. Additionally note that when an invalid edge is attempted, the new κ created will cause $P(x(e) = 1) = 0$.

The CHS method is optimistic about free space. Sampling $\phi \sim \mathcal{K}$ yields worlds with only a few occupied voxels, not representative of realistic scenarios, though as a single voxel still blocks an edge the edge validities x may still match realistic scenarios. However, while a particle filter with good initialization begins with a good estimate of $P(x)$, it may take many collisions to build up $\mathcal{K}$ sufficiently.

Approach 3: Mixture of Experts We would like to benefit from an MPF prior, but also recover in the case of a bad initialization. In real world examples, it is unknown if an initial b_0 for the MPF is accurate *a priori*. Intuitively, online adaptation can be achieved by comparing particles Φ_t to Φ_0. If measurement updates cause particles to congregate in regions predicted by particles Φ_0 then the prior likely provides a reasonable model of the world. If instead particles update to unlikely regions or disappear entirely the prior was likely not accurate, and we would like to fall back to the CHS belief.

To achieve this behavior we mix the CHS belief b_t^{CHS} and MPF belief b_t^{MPF} using weights $\beta_t = (\beta_t^{MPF}, \beta_t^{CHS})$ to get the following:

$$b_t(\phi) = \frac{\beta_t^{MPF} b_t^{MPF}(\phi) + \beta_t^{CHS} b_t^{CHS}(\phi)}{\beta_t^{MPF} + \beta_t^{CHS}} \tag{5}$$

To set β_t^{MPF}, we consider three terms of interest: Φ_t is the current set of particles in the MPF, b_0^{MPF} is the initial MPF belief before any observations, and b^U is a uniform belief over a support set of volume V. The weights are set as:

$$\beta_t^{CHS} = 1 \tag{6}$$

$$\beta_t^{MPF} = \mathbb{E}_{\phi \sim b_t^{MPF}} \left[\frac{P(\phi | b_0^{MPF})}{P(\phi | b^U)} \right] \tag{7}$$

$$= \sum_{\phi_t^i \in \Phi_t} \mu_t^i \frac{P(\phi_t^i | b_0^{MPF})}{P(\phi_t^i | b^U)} = \sum_{\phi_t^i \in \Phi_t} \mu_t^i \frac{b_0^{MPF}(\phi_t^i)}{1/V} = V \sum_{\phi_t^i \in \Phi_t} \mu_t^i b_0^{MPF}(\phi_t^i) \tag{8}$$

where V is a tuning parameter. In other words, we set the weight of the MPF belief β_t^{MPF} by iterating over all particles and doing a weighted sum of the likelihood of the particle *under the original MPF belief* b_0^{MPF}. The weight β_t^{CHS} is set to be constant.

The rationale for setting β_t^{MPF} in this way is to measure how much the current MPF belief b_t^{MPF} has deviated from the original belief b_0^{MPF}. A large deviation indicates that the prior was not a good estimate and we should instead trust CHS. When the MPF prior b_0^{MPF} is accurate, there are at least some particles that have both a high weight μ_t^i and high likelihood under the original prior $b_0^{MPF}(\phi_t^i)$. Hence β_t^{MPF} is high. The deviation w.r.t b_0^{MPF} is measured relative to a uniform distribution with volume V.

When the prior is inaccurate, particles may still have a high weight μ_t^i. However $b_0^{MPF}(\phi_t^i)$ will be small since the particles have moved significantly, thus resulting in a small β_t^{MPF}.

5 Strategies for Solving the BTP

Since we established that BTP is NP-complete [40], we explore a number of efficient approximation strategies to solve the problem, by drawing from heuristics used in the related Canadian Traveler's Problem (CTP) [33] (Section 5.2). We also propose a new heuristic (Section 5.1) that (to the best of our knowledge) has not been applied to a CTP.

5.1 Collision Measure (CM)

This heuristic balances exploration (assuming unexplored edges are free) with exploitation (penalizing edges with low validity likelihoods). The agent is at a vertex v_t and decides which edge e_t from the set of outgoing edges $\mathcal{N}(v_t)$ to traverse as follows:

$$\begin{aligned}\widehat{\mathcal{G}} &= (\mathcal{V}, \mathcal{E}, w(e) - \alpha \log P(x(e) = 1|\psi_t)) \\ e_t &= \left\{ e \in \mathcal{N}(v_t) \;\middle|\; e \in \textsc{ShortestPath}(\widehat{\mathcal{G}}, v_t, v_g) \right\}\end{aligned} \tag{9}$$

Here $\widehat{\mathcal{G}}$ is an optimistic graph created by removing all edges that are invalid with probability 1 given observation history ψ_t. Further, the weights are penalized by log-probability. Log-probability is chosen because for a path ξ, the log-probability is additive over edges assuming independence, i.e., $\log P(x(\xi)) = \sum_{e \in \xi} \log P(x(e))$. A known blocked edge $(P(x(e) = 1|\psi) = 0)$ yields a weight of ∞, and a known free edge $(P(x(e) = 1|\psi) = 1)$ yields $w(e)$. At each iteration the CM strategy finds the shortest path over $\widehat{\mathcal{G}}$ and attempts the first edge.

We provide the outline of theoretical justification for using this heuristic, (see supplementary material [40] for a detailed discussion). We first map BTP to a Bayesian search [41]. In Bayesian search, an agent repeatedly inspects a series of n boxes until an item is found. The goal is to minimize the expected cost of searching the box. A greedy policy selects the box with the largest ratio of probability of containing an object over the cost of searching the box $\frac{p_i}{c_i}$. Dor et al. [42](Theorem 4.1) proved that a greedy policy has cost at most 4 times optimal cost.

We modify BTP as follows - the agent picks a path, travels along it till an obstacle is encountered, backtracks to the start and tries another path. This is a Bayesian search problem. A greedy policy is equivalent to a more general notion of the collision measure policy that can solve the following optimization

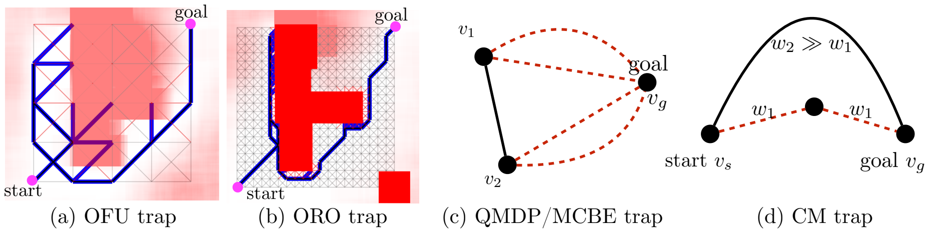
$$e_t = \left\{ e \in \mathcal{N}(v_t) \;\middle|\; e \in \operatorname*{arg\,min}_{\xi} \frac{w(\xi)}{P(x(\xi) = 1|\psi_t)} \right\} \tag{10}$$

This has a bound of 4 w.r.t the optimal policy in the modified problem, and a bound of 8 w.r.t the original BTP problem.

The optimization in (10) is intractable as $P(x(\xi) = 1)$ is not additive. However we can instead solve $\arg\min_{\xi} w(\xi) - \alpha \log P(x(\xi) = 1|\psi_t)$ where the cost function is additive (as log-probabilities are additive) and decomposes nicely. We show in supplementary material [40] that this is a suitable approximation of the near-optimal policy. Furthermore, Collision Measure is complete on the modified BTP even when using the CHS approximation of the belief. Using CHS there are finite ξ, each attempt either reaches a goal or marks an edge as invalid, and no valid edge will ever be eliminated.

5.2 Baselines

To benchmark our proposed Collision Measure strategy we consider three categories of strategies commonly used in POMDPs – approaches that approximate the optimal



(a) OFU trap (b) ORO trap (c) QMDP/MCBE trap (d) CM trap

Fig. 5: Pitfalls for various strategies for a 2D BTP problems.

expected cost-to-go of an action, also referred to as Q-value, with heuristics, approaches that use simulation to evaluate actions, and approaches that plan to gather information. For more details, we refer the reader to [40].

Optimism in the Face of Uncertainty (OFU) [43]: Find the shortest path on the optimistic graph and move along the edge on it.

Thompson Sampling (TS) [44]: Sample a world from the current belief, find the shortest path in that world, and move along the edge on it.

QMDP [44]: Given current belief, move along the edge with the least expected cost-to-go assuming the world is revealed at the next timestep.

Most Common Best Edge (MCBE): Given the current belief, move along the edge that has the highest probability of belonging to a shortest path.

Optimistic Rollout (ORO) [33]: Sample a world from the current belief, simulate moving along an edge and rollout with an optimistic policy. Move along the edge with best Q-value.

Upper Confidence Tree (UCT) [35]: Conduct a Monte-Carlo Tree Search [45] where nodes are belief states and actions are edges to move along. The value of each belief is averaged over successors. To select actions for expansion during search, Upper Confidence Bound (UCB) is used.

Interleaving Planning and Control [8]: Alternate between a global RRT planner and greedy local controller to plan a path to the goal through $\mathcal{C}$ with the least probability of collision. Note this is a strategy for the planning with contact feedback problem, but does not directly map to a BTP.

5.3 Pitfalls for Heuristic Strategies

Since all strategies considered are heuristics, it is important to recognize the pitfalls that they face. We illustrate these in Fig. 5. OFU is easily tricked into exploring cul-de-sacs that do not lead to the goal (Fig. 5(a)). A Bayes-aware heuristic would be able to predict the cul-de-saac and backtrack earlier. ORO offers significant improvement over OFU as it simulates executing OFU. However simply increasing the density of the grid yields a BTP where all neighbors of v_s fall into a cul-de-sac (Fig. 5(b)). ORO is not able to discover the non-myopic sequence of actions.

QMDP and MCBE avoid such optimistic pitfalls. However they rely on uncertainty disappearing after performing the first action. This can lead to infinite loops as shown in Fig. 5(c). The belief is such that the solid edge is known to be feasible while only

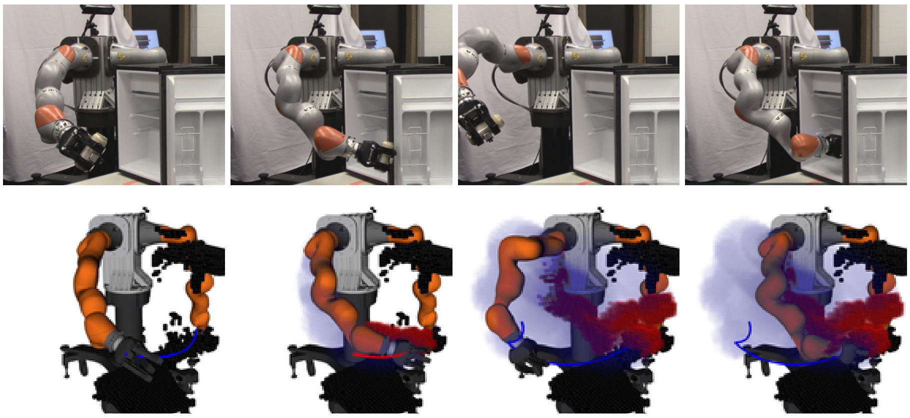


Fig. 6: `Refrigerator` - Victor moving to place an object inside a refrigerator.

one of dotted edges is feasible. When the agent is at v_1, it wishes to move to v_2 and vice-versa.

CM is also susceptible to pitfalls because it treats $P(x)$ independently. Fig. 5(d) shows an example where the solid edge is feasible while only one of the dotted edges is feasible. The only feasible path is the longer path with weight w_2. CM will choose the lower path as long as $2w_1 - \alpha \log 0.5 < w_2$.

However, of the four traps, the CM trap is the least concerning. In Fig. 5(d), the suboptimality of CM is at most $\frac{4w_1+w_2}{w_2}$ which is small as $w_2 \gg w_1$. Moreover, an appropriate α would lead to the optimal answer. This suggest a sweep over α parameter in practice would help prevent such pitfalls.

6 Experiments

We performed experiments on simulated and real worlds for the "Victor" robot's right arm, a KUKA iiwa 7DOF arm that provides joint torque feedback.

Implementation Details: W is represented by a 200x200x200 voxel grid implemented on the GPU using GPUVoxels [46]. Computing $P(x(e)|\psi)$ involves the expensive computation of swept volumes W_e, approximated by discretizing the configurations with a distance of 0.02 rad. For efficiency we lazily compute and cache W_e.

We constructed $\mathcal{G}$ in the $\mathbb{R}^7$ configuration space corresponding to the right arm of the Victor robot with 10000 vertices generated from the 7D Halton sequence and with edges connecting vertices within 1.8 rad, yielding $|\mathcal{E}| = 259146$. All strategies considered in Section 5 involve repeated shortest path queries over subgraphs of $\mathcal{G}$ with modified edge weights. Although any best-first search method is sufficient, we performed all shortest path queries using LazySP [12] to minimize the number of expensive edge-evaluation operations. All trials were conducted on an i7-7700K with a NVidia-1080Ti GPU.

Scenarios We considered 2 real robot scenarios - `Refrigerator` and `RealTable`. In `Refrigerator`, Victor must reach into a refrigerator from behind (Fig. 6). In

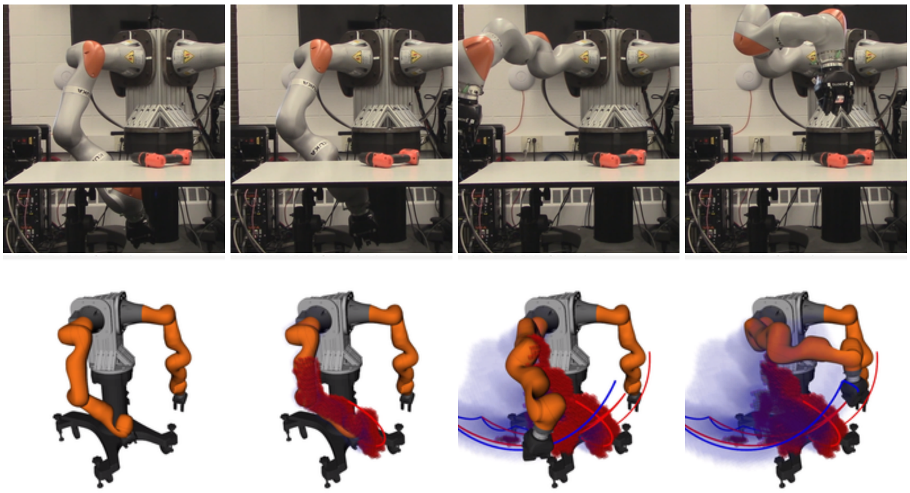


Fig. 7: RealTable - Victor moving from below to above a table.

RealTable, Victor must move from below the table to above (Fig. 7). We also consider 2 simulated robot scenarios (Fig. 8) - Bookshelf and Box. In Box, Victor must reach into a box on a table where the back of the box unknown (which is a typical scenario due to sensor occlusion). In Bookshelf, Victor must reach into a bookshelf at a height above it.

We consider CHS, MPF with 100 particles, and MoE models of the belief. The MPF requires an initial belief b_0^{MPF}, which can have drastic effects on the behavior of strategies.

We consider three levels of difficulties based on how the prior b_0^{MPF} is chosen.

- Easy: true unknown obstacles with offset $\sim \mathcal{N}(0, 0.1)$
- Medium: true unknown obstacles with offset $\sim \mathcal{N}(0.1, 0.4)$
- Hard: a chair in the corner, with no knowledge of the relevant obstacles

In the real robot scenarios the Easy and Medium particle priors were manually generated, approximated the shape of the true obstacle. In the Refrigerator scenario W_{obs} is populated using a Kinect sensor mounted on Victor's head. In the RealTable scenario Victor is wearing a blindfold.

We compare across the three beliefs proposed in Section 4 and all strategies from Section 5, except UCT which was not tested due to excessive computational time. For the stochastic TS strategy we average across 10 trials. We test our proposed CM with $\alpha = 1$ and $\alpha = 10$. We also compare against the (non-BTP) baseline proposed in [8] which interleaves an RRT with a local controller to find low cost paths through $\mathcal{C}$.

Results: Select results for the Bookshelf scenario are shown in Fig. 9 with full results for all scenarios shown in [40]. For the non-BTP baseline [8] applied to the Bookshelf scenario we observe only 2 out of 20 trials succeeded within a 15 minute time limit.

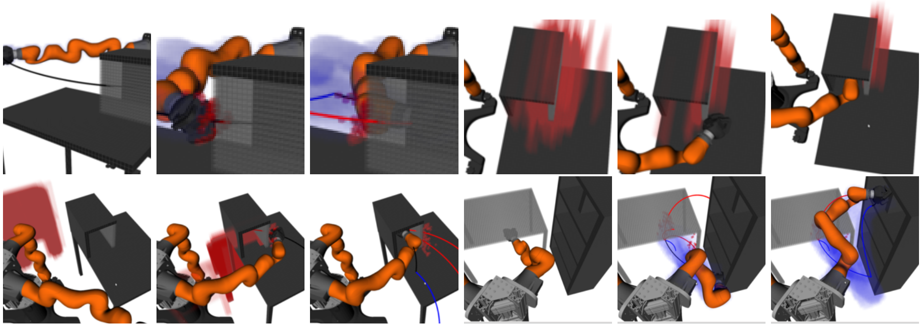


Fig. 8: Simulation scenarios. Here CM is used in all scenarios. Top left: `Easy` setting of `Box` using CHS. Top right: `Easy` setting of `Box` using MPF. Bottom left: `Hard` setting of `Box` using MoE. Bottom right: `Hard` setting of `Bookshelf` using CHS.

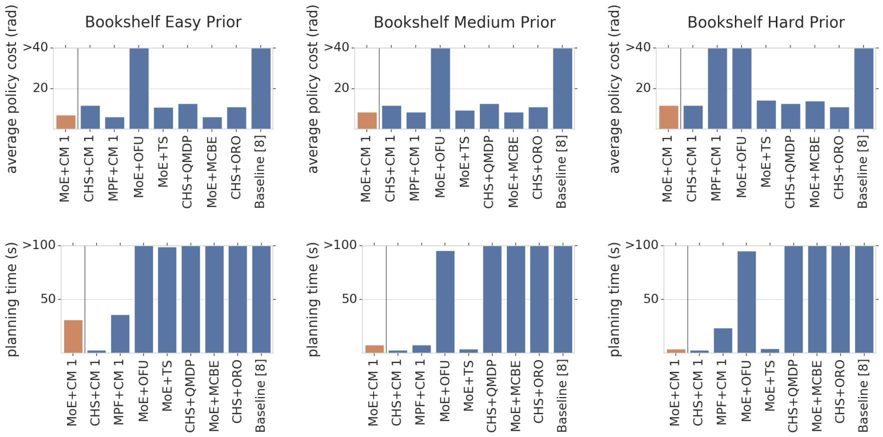


Fig. 9: Results of applying various belief strategies and policies to the `Bookshelf` BTP. Our proposed MoE+CM is consistently fast and solves the BTP with low cost.

Constraining motion to a roadmap yields a manageable action space and depth for the search for the strategies proposed. Furthermore, the roadmap allows reuse of the computationally expensive quantity $P(x(e)|\psi)$ within a single SHORTESTPATH query, and reuse of the edge swept volume W_e between queries. Compared to the previous baseline [8], we observe a significant improvement using the BTP framework.

Furthermore, we observe three key takeaways from the experiments.

1. CM performs well. CM consistently outperforms OFU, providing a lower cost policy in 19/24 experiments across scenarios, beliefs, and prior hardness. For our proposed MoE belief, CM outperforms OFU in 11/11 experiments, on average yielding 37% the cost. Compared to MCBE, CM yields a lower cost in 17/26 trials. In ad-

dition, averaged across all trials the planning time of CM is 15s, while MCBE is 217s.

2. MPF with a good prior performs well but breaks down when poorly initialized. MPF with the `Easy` prior outperforms CHS in 21/22 trials across all strategies and scenarios. MPF with the `Hard` prior only outperforms CHS in 1/22 trials, causing strategies to fail in half of trials.
3. MoE costs are approximately the minimum of MPF and CHS when using CM.

7 Conclusions

We proposed the Blindfolded Traveler's Problem as a class of problems in planning under uncertainty. We showed that contact-based planning is an instance of BTP. We examined various strategies for approximating the belief over the workspace obstacles based on contact feedback and argue for a Mixture of Experts that work well with and without correct initialization. We also examined various policies for approximately solving the BTP and propose a new policy, Collision Measure, that is both efficient and has theoretical guarantees.

There are several possibilities for future work. As currently modeled the traveler is constrained to $\mathcal{G}$ which, for high dimensional $\mathcal{C}$, possess long edges for a practically sized $|\mathcal{V}|$. Long edges may result in significant backtracking after a collision. Potential alternatives would be to dynamically alter $\mathcal{G}$ by adding vertices and edges after a collision to avoid such backtracking. Another direction is to examine alternate schemes for setting β^{MPF} based on $f-$divergence between b_t^{MPF} and the original belief b_0^{MPF}.

Bibliography

[1] M. Siegel, P. Gunatilake, and G. Podnar. Robotic assistants for aircraft inspectors. *IEEE Instrumentation & Measurement*, 1998.

[2] D. Park, A. Kapusta, J. Hawke, and C. C. Kemp. Interleaving planning and control for efficient haptically-guided reaching in unknown environments. In *Humanoids*, 2014.

[3] L.E. Kavraki, P. Svestka, J.C. Latombe, and M.H. Overmars. Probabilistic roadmaps for path planning in high-dimensional configuration spaces. *Robotics and Automation, IEEE Transactions on*, 1996.

[4] Anthony Stentz. Optimal and efficient path planning for partially known environments. In *Intelligent Unmanned Ground Vehicles*. 1997.

[5] T. Bhattacharjee, P. Grice, A. Kapusta, M. Killpack, D. Park, and C. Kemp. A robotic system for reaching in dense clutter that integrates model predictive control, learning, haptic mapping, and planning. *IROS*, 2014.

[6] B. Saund, S. Chen, and R. Simmons. Touch based localization of parts for high precision manufacturing. In *ICRA*, 2017.

[7] M. Klingensmith, M. Koval, S. Srinivasa, N. Pollard, and M. Kaess. The manifold particle filter for state estimation on high-dimensional implicit manifolds, April 2016.

[8] B. Saund and D. Berenson. Motion planning for manipulators in unknown environments with contact sensing uncertainty. In *ISER*, 2018.

[9] K Hauser. Lazy collision checking in asymptotically-optimal motion planning. 2015.

[10] Robert Bohlin and Lydia E Kavraki. Path planning using lazy PRM. In *ICRA*, 2000.

[11] Benjamin Cohen, Mike Phillips, and Maxim Likhachev. Planning single-arm manipulations with n-arm robots. In *Eigth Annual Symposium on Combinatorial Search*, 2015.

[12] C. M Dellin and S. Srinivasa. A unifying formalism for shortest path problems with expensive edge evaluations via lazy best-first search over paths with edge selectors. In *ICAPS*, 2016.

[13] N. Haghtalab, S. Mackenzie, A. Procaccia, O. Salzman, and S. Srinivasa. The provable virtue of laziness in motion planning. In *ICAPS*, 2018.

[14] S. Choudhury, C. Dellin, and S. Srinivasa. Pareto-optimal search over configuration space beliefs for anytime motion planning. In *IROS*, 2016.

[15] Aditya Mandalika, Sanjiban Choudhury, Oren Salzman, and Siddhartha Srinivasa. Generalized lazy search for robot motion planning: Interleaving search and edge evaluation via event-based toggles. 2019.

[16] V. Narayanan and M. Likhachev. Heuristic search on graphs with existence priors for expensive-to-evaluate edges. In *ICAPS*, 2017.

[17] Simon Tong and Daphne Koller. Support vector machine active learning with applications to text classification. *Journal of machine learning research*, 2001.

[18] D. Golovin, A. Krause, and D. Ray. Near-optimal bayesian active learning with noisy observations. In *NIPS*, 2010.

[19] Yuxin Chen, Shervin Javdani, Amin Karbasi, Drew Bagnell, Siddhartha Srinivasa, and Andreas Krause. Submodular surrogates for value of information. In *AAAI*, 2015.

[20] S. Choudhury, S.S. Srinivasa, and S. Scherer. Bayesian active edge evaluation on expensive graphs. In *IJCAI*, 2018.

[21] Sanjiban Choudhury, Shervin Javdani, Siddhartha Srinivasa, and Sebastian Scherer. Near-optimal edge evaluation in explicit generalized binomial graphs. In *Advances in Neural Information Processing Systems*, 2017.

[22] B. Awerbuch and R. Kleinberg. Adaptive routing with end-to-end feedback: Distributed learning and geometric approaches. In *ACM symposium on Theory of computing*, 2004.

[23] András György, Tamás Linder, Gábor Lugosi, and György Ottucsák. The on-line shortest path problem under partial monitoring. *Journal of Machine Learning Research*, 2007.

[24] Mohammad Sadegh Talebi, Zhenhua Zou, Richard Combes, Alexandre Proutiere, and Mikael Johansson. Stochastic online shortest path routing: The value of feedback. *IEEE Transactions on Automatic Control*, 2017.

[25] Nicolo Cesa-Bianchi and Gábor Lugosi. Combinatorial bandits. *Journal of Computer and System Sciences*, 2012.

[26] Sven Koenig and Maxim Likhachev. D* lite. In *AAAI*, 2002.

[27] Dave Ferguson and Anthony Stentz. Field d*: An interpolation-based path planner and replanner. In *Robotics research*. 2007.

[28] Charles Richter, William Vega-Brown, and Nicholas Roy. Bayesian learning for safe high-speed navigation in unknown environments. In *Robotics Research*. 2018.

[29] L. Janson, T. Hu, and M. Pavone. Safe motion planning in unknown environments: Optimality benchmarks and tractable policies. *arXiv preprint arXiv:1804.05804*, 2018.

[30] Christos H Papadimitriou and Mihalis Yannakakis. Shortest paths without a map. *Theoretical Computer Science*, 1991.

[31] Evdokia Nikolova and David R Karger. Route planning under uncertainty: The canadian traveller problem. In *AAAI*, 2008.

[32] Dror Fried, Solomon Eyal Shimony, Amit Benbassat, and Cenny Wenner. Complexity of canadian traveler problem variants. *Theoretical Computer Science*, 2013.

[33] P. Eyerich, T. Keller, and M. Helmert. High-quality policies for the canadian traveler's problem. In *AAAI*, 2010.

[34] Sung Wook Yoon, Alan Fern, Robert Givan, and Subbarao Kambhampati. Probabilistic planning via determinization in hindsight. In *AAAI*, 2008.

[35] Sylvain Gelly and David Silver. Combining online and offline knowledge in uct. In *ICML*, 2007.

[36] A. Guez, D. Silver, and P. Dayan. Efficient bayes-adaptive reinforcement learning using sample-based search. In *Advances in neural information processing systems*, 2012.

[37] Zhan Wei Lim, David Hsu, and Wee Sun Lee. Shortest path under uncertainty: Exploration versus exploitation. In *UAI*, 2017.

[38] Zhan Wei Lim, David Hsu, and Wee Sun Lee. Adaptive informative path planning in metric spaces. *IJRR*, 2016.

[39] Zhan Wei Lim, David Hsu, and Wee Sun Lee. Adaptive stochastic optimization: From sets to paths. In *Advances in Neural Information Processing Systems*, 2015.

[40] Brad Saund, Sanjiban Choudhury, Siddharth Srinivasa, and Dmitry Berenson. The blindfolded robot : A bayesian approach to planning with contact feedback. 2019. URL `https://github.com/bsaund/BTP_Appendix`.

[41] S. Ross. *Introduction to stochastic dynamic programming*. Academic press, 2014.

[42] Avner Dor, Eitan Greenshtein, and Ephraim Korach. Optimal and myopic search in a binary random vector. *Journal of applied probability*, 1998.

[43] Ronen I Brafman and Moshe Tennenholtz. R-max-a general polynomial time algorithm for near-optimal reinforcement learning. *Journal of Machine Learning Research*, 2002.

[44] Michael L Littman, Anthony R Cassandra, and Leslie Pack Kaelbling. Learning policies for partially observable environments: Scaling up. In *Machine Learning Proceedings 1995*. 1995.

[45] Levente Kocsis and Csaba Szepesvári. Bandit based monte-carlo planning. In *European conference on machine learning*, 2006.

[46] A. Hermann, F. Drews, J. Bauer, S. Klemm, A. Roennau, and R. Dillmann. Unified GPU voxel collision detection for mobile manipulation planning. In *IROS*, 2014.

[47] H. Laurent and R. Rivest. Constructing optimal binary decision trees is np-complete. *Information processing letters*, 1976.

[48] C. Papadimitriou and M. Yannakakis. Shortest paths without a map. *Theoretical Computer Science*, 1991.

[49] Z. Bnaya, A. Felner, and S. Shimony. Canadian traveler problem with remote sensing. In *IJCAI*, 2009.

[50] Aditya Mandalika, Oren Salzman, and Siddhartha Srinivasa. Lazy Receding Horizon A* for Efficient Path Planning in Graphs with Expensive-to-Evaluate Edges. In *icaps*, 2018.

[51] William R Thompson. On the likelihood that one unknown probability exceeds another in view of the evidence of two samples. *Biometrika*, 1933.

[52] Olivier Chapelle and Lihong Li. An empirical evaluation of thompson sampling. In *NIPS*, 2011.

[53] S. Agrawal and N. Goyal. Further optimal regret bounds for thompson sampling. In *Artificial intelligence and statistics*, 2013.

[54] I. Osband, D. Russo, and B. Van Roy. (more) efficient reinforcement learning via posterior sampling. In *NIPS*, 2013.

[55] D. Russo, B. Van Roy, et al. A tutorial on thompson sampling. *Foundations and Trends in Machine Learning*, 2018.

[56] D. Silver, A. Huang, C. Maddison, et al. Mastering the game of go with deep neural networks and tree search. *nature*, 2016.

[57] David Silver and Joel Veness. Monte-carlo planning in large POMDPs. In *NIPS*, 2010.

Appendix for "The Blindfolded Robot : A Bayesian Approach to Planning with Contact Feedback"

A Analysis of BTP

A.1 Mapping the Problem to a POMDP

We map BTP problem to a Partially Observable Markov Decision Process (POMDP) specified by the following tuple $\langle \mathcal{S}, \mathcal{A}, T, C, \mathcal{O}, Z \rangle$ which we define as follows.

The state $s \in \mathcal{S}$ is the tuple $s = (v, x, \eta)$ where $v \in \mathcal{V}$ is the current location of the traveler on the graph $\mathcal{G}$, x is the binary vector of edge validities and η is a vector of edge blockages. The state is partially observable, i.e. v is observable but the rest is latent.

Given state $s \in \mathcal{S}$, the action $a \in \mathcal{A}(s)$ is any edge $e \in \mathcal{E}$ that can be traversed, i.e., whose parent is v. Let the result of the attempt be $(v', c) = \Gamma(v, e, x, \eta)$. The transition function $T(s, a, s')$ is deterministic, i.e. $s' = (v', x, \eta)$. Similarly, the one step cost is $C(s, a) = c$. The observation $o \in \mathcal{O}$ is the tuple $o = (x(e), \eta(e))$. Hence the observation model $Z(s', a, o)$ is deterministic.

Since the state is partially observable, the POMDP is viewed as a MDP over belief b. A POMDP policy $\pi(b)$ maps b to actions. The optimal policy π^* accumulates the minimum cost in expectation. The Q-value of action a in a belief state is the expected total cost of taking a and subsequently following π^*, i.e.

$$Q(b, a) = \mathbb{E}_{s \sim b} \left[C(s, a) \right] + \mathbb{E}_{b' \sim P(.|b,a)} \left[V^{\pi^*}(b') \right] . \tag{11}$$

A.2 Computational Complexity

In BTP, the belief b is over a continuous space $\mathcal{S}$ due to blockages η, i.e. the exact belief is infinite dimensional. This necessitates approximation based approaches that rely on non-parametric sample-based belief representations. For the proofs in this section we consider a discrete/simplified BTP with discrete b by fixing $\eta(e) = 1$. Furthermore, we examine the BTP decision problem instead of the optimally problem.

We follow an analysis parallel to Lim et al. [37] to show that the BTP decision problem is NP-complete by showing it is both in NP and NP-Hard.

We first prove that the BTP decision problem is in NP. For this result we consider an explicit description of the input $\mathcal{P}$, that is $\mathcal{P}$ specifies probability of each possible world. Note that $\mathcal{P}$ could be exponentially larger than $|\mathcal{E}|$. In this case BTP would still be in NP, though $\mathcal{P}$ (part of the input) would be so large as to make this claim of limited use.

We also prove that BTP is NP-hard by reduction from the Optimal Decision Tree (ODT) problem. The ODT problem is as follows. We have a finite set of hypotheses $\mathcal{H} = (h_1, h_2, \dots, h_n)$ and a finite set of tests $\mathcal{T} = (t_1, t_2, \dots, t_m)$. A test t_i leads to an outcome $o_i \in \{0, 1\}$ depending on the latent hypothesis $h^* \in \mathcal{H}$. The objective is to find a policy that identifies h^* with the least number of tests when h^* is uniformly distributed. The policy is a binary decision tree where nodes are tests, edges branch on outcomes and the terminal nodes stores the latent object $h^* \in \mathcal{H}$. The decision version of the problem, which asks if a policy with expected cost of less than or equal to w is NP-complete [47].

Theorem 1. *We define the decision version of Blindfolded Traveler Problem as the question of whether there is a policy with expected cost less than or equal w. The decision version of discrete/simplified BTP is NP-complete.*

Proof. The solution of BTP can be represented as a policy tree. Note that nodes and edges in this policy tree are distinct from nodes and edges in the graph $\mathcal{G}$ of the BTP. Nodes of this policy tree represent testing an unevaluated edge in $\mathcal{G}$. A node in the policy tree may even represent traversing several known edges in $\mathcal{G}$ to reach the unknown edge in $\mathcal{G}$. Each edge of the policy tree is an observation o received upon performing an edge. A BTP is solved by traversing the policy tree till the leaf node is reached, i.e. evaluating unknown edges, receiveing observations until the goal is reached.

The optimal policy tree is polynomial size in the input of BTP. Consider that each edge in the policy tree corresponds to an action (or actions) in the BTP that will determine the validity of one edge in $\mathcal{G}$, thus the policy tree can be at most $|\mathcal{E}|$ deep. Furthermore, there can be at most one unique path through the policy tree for each hypothesis world in $\mathcal{P}$. Since we assume each hypothesis world is explicitly represented in $\mathcal{P}$, the optimal policy tree is polynomial in $|\mathcal{G}|$ and $|\mathcal{P}|$.

Finally, computing the expected cost of a policy is simply a weighted sum for all paths through the policy tree. Hence the BTP decision problem is in NP.

We now show that ODT is polynomial time reducible to BTP and thus BTP is NP-hard. Given an instance of ODT$(\mathcal{H}, \mathcal{T})$, we consider an instance of BTP $\langle \mathcal{G}, \mathcal{P}, v_s, v_g \rangle$ as follows. Consider the BTP problem shown in Fig. 10. The cluster of edges $\{e_1, \ldots, e_m\}$ correspond to the tests $\{t_1, \ldots, t_m\}$. Note again that the blockages for all tests is fixed at $\eta(e) = 1$. An agent attempting to traverse the edge e_j will either be successful and reach the vertex v_j, or unsuccessful and the agent will return back to v_s. The cluster of edges $\{e_{m+1}, \ldots, e_{m+n}\}$ has only one valid edge correspond to identifying the correct hypothesis from $(h_1, h_2, \ldots, h_n)$. The weights of the left cluster of edges $\{e_1, \ldots, e_m\}$ is 1 and the right cluster of edges is $2m$.

We set up the prior $\mathcal{P}$ to be uniform over a set of candidate vectors x_i, each of which corresponds to a h_i. For the latent hypothesis h_i, we set the edge validities $x(e_j) = o_j$ for $j = \{1, \ldots, m\}$, i.e. the outcome of the tests for h_i. For the other cluster, we set $x(e_{m+i}) = 1$ and all other edges to 0, i.e., $x(e_j) = 0$ for $j = \{m, \ldots, m+n\}, j \neq i$. We now argue that expected cost of ODT instance is less than or equal to some value w iff cost of BTP instance is less than or equal to $2w + 2m$.

First, if the cost of the ODT is $\leq w$ then the agent can traverse the left cluster using the policy tree of ODT and identify the correct hypothesis h^* with cost $\leq 2w$. The agent then goes to v_g using the valid edge incurring $2m$. Hence the total cost of the BTP is $\leq 2w + 2m$.

Next, we prove the converse that if the cost of the BTP is $\leq 2w + 2m$, then the cost of the ODT is $\leq w$. Note that $w > m$ is vacuous because ODT is clearly solved by at

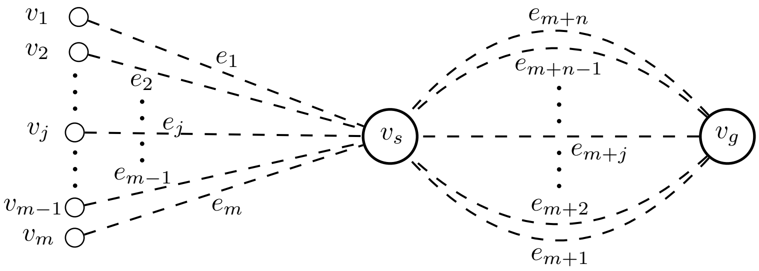


Fig. 10: Reduction from Optimal Decision Tree problem

worst evaluating all tests, which would incur cost m. Thus we consider $w \leq m$ which implies the cost of the BTP is $\leq 4m$. First consider that if an edge to v_g is attempted before identifying the correct hypothesis, there will be at least two equally likely paths with cost $2m$ and so the expected cost of any policy that tries to go directly to the goal is $\geq 4m$. Hence the agent will try to identify the true hypothesis before going to the target. If the agent solves the BTP by identifying the correct hypothesis with cost $\leq 2w + 2m$ then it also has a policy to solve the ODT with cost w.

Thus ODT is reducible to BTP in polynomial time, and since ODT is known to be NP-hard then BTP is also NP-hard. Since we also showed BTP is in NP, BTP is therefore NP-complete. □

Note that if $\mathcal{P}$ is not represented explicitly (e.g. not by a matrix of size $|\mathcal{E}|$ by the number of hypothesis worlds), but with factored representations, then the problem may no longer be in NP. Also if we further consider the location of the contact (η), the size of the hypothesis space is now continuous and this analysis no longer holds.

A.3 Relation to the Bayesian Canadian Traveler's Problem

The BTP is closely related to the Canadian Traveler's Problem (CTP) [48]. In graph search an agent executes a polity to reach a goal with the minimum expected cost. Consider the k-lookahead graph search problem, where an agent only observes the true validity of edges within k steps of its location. The Shortest Path Problem over known graphs in an instance of ∞-lookahead. The CTP is a 1-lookahead instance. For $k \geq 1$ an agent knows the state of adjacent edges and therefore will never attempt an invalid edge. In BTP, with $k = 0$, an agent might attempt invalid edges, which is the reason for the more complicated cost formulation.

In the original CTP $x(e)$ are independent. In the more general Bayesian CTP (BTCP) [37] $x(e)$ are correlated through beliefs of underlying worlds ϕ rather than beliefs directly over x. As defined, the BTP is analogous to the Bayesian CTP.

B Strategies for Solving the BTP

Since we established that BTP is a hard problem (Section A.2), we explore a number of efficient approximation strategies to solve it. We organize these approaches into three categories – approaches that approximate the Q-value with heuristics, approaches that use simulation to evaluate actions and approaches that plan to gather information. Note that while the latter approaches have theoretical guarantees, they come at the cost of computational complexity.

For all of these strategies, we assume that the agent is current at a vertex v_t and must decide which edge e_t from the set of outgoing edges $\mathcal{N}(v_t)$ to traverse. The history of observations is encoded in ψ_t.

B.1 Heuristic Estimates of Q-values

One class of approaches try to approximate optimal Q-value $Q^*(b, a)$ with an estimate $\hat{Q}(b, a)$. These approximations are motivated by different relaxations of the original problem. Since these approximations are myopic, i.e., only consider the instantaneous belief, they do not offer any performance guarantees in general. However, they are efficient to compute and perform quite well in practice.

Optimism in the Face of Uncertainty (OFU) A common approach for planning under uncertainty is to be optimistic [43], i.e., pick a world from the plausible set of worlds that leads to the lowest action value. The rationale is that either the assumption is correct and the agent does the best it can do, or the possibility is eliminated and the search space is reduced. This heuristic is commonly used in navigation [4, 26] as well as for solving CTP [49].

Formally, the approximation is $\hat{Q}(b, a) \approx \min_{s, b(s)>0} Q(s, a)$. An optimistic policy selects the best action $\pi^{\mathrm{OFU}} = \arg\min_{a} \hat{Q}(b, a)$. Mapping this back to the BTP, the agent chooses edge e_t as follows:

$$\begin{aligned} \widehat{\mathcal{G}} &= \left(\mathcal{V}, \mathcal{E} \setminus \{e \mid P(x(e) = 0|\psi) = 1\}, \mathcal{W}\right) \\ e_t &= \left\{ e \in \mathcal{N}(v_t) \;\middle|\; e \in \textsc{ShortestPath}(\widehat{\mathcal{G}}, v_t, v_g)) \right\} \end{aligned} \tag{12}$$

Here $\widehat{\mathcal{G}}$ is the optimistic graph created by removing all edges that are invalid with probability 1. The agent invokes a search subroutine $\textsc{ShortestPath}(\widehat{\mathcal{G}}, v_t, v_g)$ to compute the shortest path from current vertex v_t to goal v_g. It then looks at which of the outgoing edges $\mathcal{N}(v_t)$ belongs to the shortest path and takes that.

We can bound the sub-optimality of the optimistic policy if we alter it to backtrack whenever the shortest path is in collision. Let this policy be π^{OFU2}. This results in the following iterative policy

1. At iteration i, the agent computes shortest path from start to goal on the optimistic graph, i.e. $\xi_i = \textsc{ShortestPath}(\widehat{\mathcal{G}}_i, v_s, v_g)$
2. It moves along ξ_t till it either reaches the goal or hits a blocked edge $x(e) = 0$.
3. If it hits a blocked edge, it back tracks to start v_s and repeats.

Then the following theorem is true

Theorem 2. *Given a configuration (x, η), let w^* be the length of the shortest feasible path between v_s and v_g, and K be the number of shorter paths that are infeasible. For all such configurations, the cost of the optimistic backtracking policy π^{OFU2} is upper bounded by*

$$c(\pi^{\mathrm{OFU2}}(x, \eta)) \leq 2Kw^* \tag{13}$$

Proof. The optimistic backtracking policy will attempt the shortest path from v_s on $\widehat{\mathcal{G}}$, which must be no longer than the shortest path on $\mathcal{G}$. Each attempted path therefore incurs at most a cost of $2w^*$. Since each attempt either reaches the goal or invalidates a path shorter than w^*, there will be at most K attempts. □

Interestingly, if we changed the problem to the following:

1. The agent has to reach the goal via the shortest path from start
2. The agent is allowed to backtrack for free

this problem becomes equivalent to the shortest path planning problem on expensive graphs [12]. π^{OFU2} is then equivalent to LAZYSP [12] which has been shown to be optimal [50].

Thompson Sampling (TS) This is a commonly used heuristic for Bayesian Multi-armed Bandit (MAB) problem based on the idea of randomized probability matching [51]. At every decision step, it samples a model from a posterior and selects the optimal action for that model. Hence action selection probability is matched to the posterior of actions being optimal. In recent literature, Thompson Sampling has shown to be empirically successful [52], theoretically sound [53] and applicable beyond MAB to RL [54].

Formally, the TS policy is $\pi^{\mathrm{TS}} = \arg\min_a Q^*(s, a)$ where $s \sim b$. Mapping this back to BTP, the agent chooses edge e_t as follows:

$$\begin{aligned} \hat{x} \sim P(x|\psi_t),\ \widehat{\mathcal{G}} = (\mathcal{V}, \mathcal{E} \setminus \{e \mid \hat{x}(e) = 0\}, \mathcal{W}) \\ e_t = \left\{ e \in \mathcal{N}(v_t) \;\middle|\; e \in \textsc{ShortestPath}(\widehat{\mathcal{G}}, v_t, v_g)) \right\} \end{aligned} \tag{14}$$

Here $\widehat{\mathcal{G}}$ is the sampled valid graph from the posterior on which the agent plans the shortest path and takes a step along it. Thompson sampling usually provides a bound for MAB w.r.t Bayesian regret, i.e., the expected regret under the prior [55]. These bounds are meaningful for repeated trials on the same world, which is not the case for BTP.

QMDP This is one of the most commonly used heuristics for POMDPs [44]. It assumes that all uncertainty will disappear at the next timestep. Hence the optimal action is the one with the least expected value based on the current uncertainty.

Formally, the approximation is $\hat{Q}(b,a) \approx \mathbb{E}_{s \sim b}\left[Q^*(s,a)\right]$ and the policy is $\pi^{\text{QMDP}} = \underset{a}{\arg\min}\, \hat{Q}(b,a)$. Mapping this back to BTP, the agent chooses edge e_t as follows:

$$\begin{aligned} e_t &= \underset{e \in \mathcal{N}(v_t)}{\arg\min}\, \mathbb{E}_{(x,\eta) \sim P(\cdot|\psi_t)} \left[c + w(\textsc{ShortestPath}(\mathcal{G}(x), v', v_g))\right] \\ &\text{where } (v', c) = \varGamma(v_t, e, x, \eta) \text{ and } \mathcal{G}(x) = (\mathcal{V}, \mathcal{E} \setminus \{e \mid x(e) = 0\}, \mathcal{W}) \end{aligned} \tag{15}$$

Here we sample a set of worlds $(x,\eta) \sim P(\cdot|\psi_t)$. For each candidate edge $e \in \mathcal{N}(v_t)$, we simulate moving along the edge (which may or may not result in a success) and subsequently plan the shortest path on the revealed world.

It's straightforward to see QMDP lowerbounds the optimal value $\hat{Q}(b,a) \leq Q^*(b,a)$. There are two known drawbacks. Firstly, the policy never acts to gain information because it ignores potential observations. Secondly, and perhaps more relevant to BTP, it's susceptible to a clairvoyance trap.

Most Common Best Edge (MCBE) This is a further relaxation of the QMDP heuristic. Note that QMDP calls $\textsc{ShortestPath}(\cdot)$ a total of kN times, where k is the degree of the graph and N is the number of samples. We can reduce this to N if the agent chooses action based on the current belief, without first simulating an action.

Formally, the policy is $\pi^{\text{MCBE}} = \underset{a}{\arg\max}\, \mathbb{E}_{s \sim b}\left[\mathbb{I}(a \in \underset{a'}{\arg\min}\, Q^*(s,a'))\right]$. Mapping this back to BTP, the agent chooses edge e_t as follows:

$$\begin{aligned} \mathcal{G}(x) &= (\mathcal{V}, \mathcal{E} \setminus \{e \mid x(e) = 0\}, \mathcal{W}) \\ e_t &= \underset{e \in \mathcal{N}(v_t)}{\arg\max}\, \mathbb{E}_{(x,\eta) \sim P(\cdot|\psi_t)} \left[\mathbb{I}(e \in \textsc{ShortestPath}(\mathcal{G}(x), v_t, v_g))\right] \end{aligned} \tag{16}$$

Here we sample a set of worlds $(x,\eta) \sim P(\cdot|\psi_t)$, find the shortest path for each world and store the first edge along the path. The agent moves along the most common edge.

MCBE and QMDP do not necessarily agree on the same actions. One can construct examples where MCBE has a very high QMDP value because the action maybe quite suboptimal for worlds for which it is not on the shortest path. MCBE too is susceptible to the clairvoyance trap.

Collision Measure (CM) A drawback of the OFU policy is that it does not reason about the likelihood of a path to be valid. This can lead to excessive exploration of implausible paths. Augmenting the original $\mathcal{W}$ with a term penalizing small $P(x)$ retains the graph substructure needed for efficient search while hedging against likely blocked edges. We examine weight augmentation using the collision measure proposed in [14] for fast motion planning with C-space beliefs.

This heuristic balances exploration (assuming unexplored edges are free) with exploitation (penalizing edges with low validity likelihoods). The agent is at a vertex v_t and decides which edge e_t from the set of outgoing edges $\mathcal{N}(v_t)$ to traverse as follows:

$$\begin{aligned} \widehat{\mathcal{G}} &= (\mathcal{V}, \mathcal{E}, w(e) - \alpha \log P(x(e) = 1|\psi_t)) \\ e_t &= \left\{ e \in \mathcal{N}(v_t) \;\middle|\; e \in \textsc{ShortestPath}(\widehat{\mathcal{G}}, v_t, v_g)) \right\} \end{aligned} \tag{17}$$

Here $\widehat{\mathcal{G}}$ is an optimistic graph created by removing all edges that are invalid with probability 1 given observation history ψ_t. Further, the weights are penalized by log-probability. Log-probability is chosen because for a path ξ, the log-probability is additive over edges assuming independence, i.e., $\log P(x(\xi)) = \sum_{e \in \xi} \log P(x(e))$. A known blocked edge $(P(x(e) = 1|\psi) = 0)$ yields a weight of ∞, and a known free edge $(P(x(e) = 1|\psi) = 1)$ yields $w(e)$.

We provide theoertical justification behind such a heuristic. We begin by mapping BTP to a Bayesian Search [41] problem. Let $\Xi = (\xi_1, \xi_2, \ldots, \xi_n)$ be the set of simple paths from v_s to v_g. The probability of edge validity $P(x)$ maps to a joint probability $P((\xi_1, \xi_2, \ldots, \xi_n))$ of paths being valid. For each path ξ_k, we assign a cost twice the length of the path $c_i = 2w(\xi_i)$. We now describe a sequential game of at most n rounds. In each round the agent attempts to traverse a path ξ_k. If the path is valid, it reaches the goal and receives a cost of c_k and the game terminates. Else, it receives a cost of c_k, remains at the start and the game continues.

Let σ be a sequence of attempting paths, i.e. a particular permutation of $\{1, \cdots, n\}$. Let $\mathbb{E}\left[c(\sigma)\right]$ be the expected cost of a sequence. The optimal sequence σ^* has minimal expected cost, i.e. $\mathbb{E}\left[c(\sigma^*)\right] \leq \mathbb{E}\left[c(\sigma)\right]$ for all sequences σ^*.

Let σ^g be a sequence corresponding to a greedy policy that selects the path with the maximum posterior to cost ratio. Formally, this rule is defined as follows.

$$\sigma^g(i+1) = \arg\max_j \frac{P(\xi_j = 1|\xi_{\sigma^g(1)} = 0, \xi_{\sigma^g(2)} = 0, \cdots, \xi_{\sigma^g(i)} = 0)}{c(\xi_j)} \tag{18}$$

where the numerator is the posterior probability of a path given the observations seen thus far and the denominator is cost of the path.

Dor et al. [42](Theorem 4.1) proved that greedy has an optimality bound of 4

Theorem 3. *Given the following conditions on the game:*

1. *There exists at least one valid path*
2. *Ratio of costs are bounded* $\sup_{i,j} \frac{c_i}{c_j} < \infty$

The performance of the greedy sequence σ^g *is bounded*

$$\mathbb{E}\left[c(\sigma^g)\right] \leq 4\mathbb{E}\left[c(\sigma^*)\right] \tag{19}$$

Proof. We refer the reader to Theorem 4.1 in Dor et al. [42]. □

We now map this result back to BTP. Note that BTP has an *asymmetric cost* of attempting a path. If traversal is successful, the agent pays half price of $0.5c_i$, else in the worst case pays the full price of c_i for going all the way to goal and returning. Let $\bar{c}(\sigma)$ be the cost of a sequence under these new rules. Note that the greedy policy σ^g remains the same with these new rules. We can transfer the bound from Theorem 3

Corollary 1. *The performance of the greedy sequence* σ^g *is bounded*

$$\mathbb{E}\left[\bar{c}(\sigma^g)\right] \leq 8\mathbb{E}\left[\bar{c}(\bar{\sigma}^*)\right] \tag{20}$$

Proof. Let $\bar{\sigma}*$ be the optimal policy for the new game. Then $\bar{c}(\bar{\sigma}^*) \geq 0.5c(\bar{\sigma}^*)$ where the bound is tight if the optimal policy never encounters a blocked path. It's straightforward to see that

$$\bar{c}(\sigma^g) \leq c(\sigma^g) \leq 4c(\sigma^*) \leq 4c(\bar{\sigma}^*) \leq 8\bar{c}(\bar{\sigma}^*) \tag{21}$$

□

The greedy sequence is equivalent to a more general notion of the collision measure policy that can solve the following optimization

$$\pi^{\text{CM2}} \equiv \left\{ e \in \mathcal{N}(v_t) \;\middle|\; e \in \arg\min_{\xi} \frac{w(\xi)}{P(x(\xi)=1|\psi_t)} \right\} \tag{22}$$

The optimization in (22) is intractable as $\frac{1}{P(x(\xi)=1)}$ is not additive. We choose to approximate this with log-probability. We utilize the following inequality for $p \in (p_{\min}, 1]$ and $\alpha \geq \frac{\frac{1}{p_{\min}}-1}{\log \frac{1}{p_{\min}}}$

$$(1 - \log p) \leq \frac{1}{p} \leq (1 - \alpha \log p) \tag{23}$$

Hence $(1 - \alpha \log p)$ is a good family of approximators to $\frac{1}{p}$ which justifies (17) is an approximation.

B.2 Simulation-based Policies

This class of approaches employ *simulation* to estimate action values. We refer to the policy being simulated as the *rollout* policy $\pi(b)$. Let $V^{\pi(b)}(s)$ be the cumulative cost of the rollout policy initialized with belief b and simulated on the underlying MDP from state s. Note that unlike Section B.1, the simulator only has access to s and not the policy π. The simulator is thus able to provide observations o to the policy which updates the belief used in the rollout. We can then approximate action value as $\hat{Q}(b,a) \approx \mathbb{E}_{s\sim b}\left[c(s,a) + V^{\pi(b')}(s')\right]$, where s', b' is the next state and belief.

The attractive aspect of these approaches is that any policy from Section B.1 can be used as a rollout policy. For any such policy, we have the following upper bound

$$\hat{Q}(b,a) \geq \mathbb{E}_{s\sim b}\left[c(s,a) + V^{\pi^*}(s')\right] \geq Q^*(b,a) \tag{24}$$

If this is close to matching lower bounds from Section B.1, the value can be known exactly. However, the simulator invokes these policies $O(NTk)$ where N is the number of samples and T is the maximum horizon length, and k is the degree of the graph. Each invocation requires at least one belief update and perhaps several calls to SHORTESTPATH. Even with parallelization this is memory and computation heavy.

Optimistic Rollout (ORO) One of the simplest rollout policies is the OFU policy because it involves only one invocation of SHORTESTPATH. Let π^{OFU} be the OFU policy. Let $V^{\pi^{\mathrm{OFU}}(v,\psi)}(x,\eta)$ be the evaluation of the policy starting from vertex v with history ψ on an underlying graph (x,η). The agent chooses edge e_t as follows:

$$\begin{aligned} e_t &= \underset{e \in \mathcal{N}(v_t)}{\arg\min}\ \mathbb{E}_{(x,\eta)\sim P(\cdot|\psi_t)} \left[c + V^{\pi^{\mathrm{OFU}}(v',\psi')}(x,\eta) \right] \\ &\text{where } (v',c) = \Gamma(v_t,e,x,\eta) \text{ and } \psi' = \psi_t \cup (x(e),\eta(e)) \end{aligned} \tag{25}$$

Upper Confidence Tree (UCT) This is a state of the art algorithm from planning under uncertainty [45] which combines the framework Monte-Carlo Tree Search with Upper Confidence Bound (UCB) for action selection. It has successfully been used for solving games [35, 56], POMDPs [57] and Bayesian RL [36]. The idea builds on top of simulation based evaluation but differs on how actions are selected and how estimates are backed up.

Each UCT rollout begins with a belief sate b_0 and grows a tree where each node is a successor b. The value of each action $\hat{Q}(b,a)$ is an average over successors. To expand a given node, the search has to select one of k actions that according to the following rule:

$$\underset{a_i}{\arg\max}\ B\sqrt{\frac{\log N(b,a_i)}{N(b,a_i)}} - \hat{Q}(b,a) \tag{26}$$

Once the search goes off the tree, it uses a roll out policy (such as π^{OFU}) to finish the episode. UCT has been proved to converge to the exact Q-values [33] asymptotically, i.e. $\hat{Q}(b,a) \to Q(b,a)$. However there is no such guarantee on the rate of convergence. Hence, in practice, UCT might have to do a large number of simulations.

B.3 Planning to gather information

The final class of approach we consider is where an agent plans to explicitly gather information. One such approach is the Hedged Shortest Path under Determinization (HSPD) [37] algorithm which was original defined for the Bayesian Canadian Traveler Problem. HSPD determinizes the graph according to the most likely edge (MLE) assumption - each edge is set to valid if the marginal posterior probability is 0.5. The agent at every timestep plans two paths - exploitation and exploration. The exploitation is simply the shortest path to goal. The exploration path is the shortest path that reduces the version space to less than 0.5 fraction. The agent then takes the shorter of these paths and travels till it encounters a blocked edge, following which it returns to the start. This happens only a logarithmic number of times till it finds a path to goal.

This method for the BCTP has a near-optimality guarantee of $4(\log\delta + 1)$ where δ is the minimum prior probability of an underlying world. However, there are two concerns with the approach. Planning in belief space requires several invocations to the Bayes filter which can be expensive. Secondly, for the case of BTP the value of δ can be quite small as the observations are continuous. For these reasons, we chose not to proceed with this method although an efficient implementation for BTP would be of great interest.

C Full Numerical Results

	MoE			CHS	MPF		
	Easy	Med	Hard	CHS	Easy	Med	Hard
CM 1	7.3	11.6	11.6	11.6	7.3	10.1	fail
CM 10	5.0	7.3	7.3	7.3	5.0	11.6	fail
OFU	25.5	25.5	25.5	25.5	7.3	14.2	fail
ORO	-	-	-	13.7	5.0	10.1	-
MCBE	5.0	12.2	12.2	12.2	5.0	13.2	fail
QMDP	5.0	11.9	13.4	11.9	5.0	10.1	fail
TS	7.8	7.3	11.6	21.1	5.0	13.5	fail

(a) Box Policy Cost

MoE			CHS	MPF		
Easy	Med	Hard	CHS	Easy	Med	Hard
8.5	8.9	8.3	4.1	11.7	15.1	fail
14.3	14.3	11.4	6.5	14.8	15.1	fail
2.5	2.6	3.4	2.3	4.7	3.1	fail
-	-	-	1779.9	1648.0	3446.8	-
46.1	186.9	224.9	173.5	47.1	38.2	fail
716.6	1782.2	3040.5	663.3	579.0	1150.9	fail
14.8	6.9	38.7	68.1	3.9	8.0	fail

(b) Box Planning Times

	MoE			CHS	MPF		
	Easy	Med	Hard	CHS	Easy	Med	Hard
CM 1	7.0	8.4	11.7	11.7	6.1	8.4	117.2
CM 10	10.0	12.3	10.1	10.1	10.0	12.1	117.2
OFU	117.4	100.4	100.4	51.8	9.1	14.5	117.2
ORO	-	-	-	11.1	7.4	-	-
MCBE	6.1	8.4	13.9	14.9	6.1	8.4	69.1
QMDP	-	-	-	12.7	6.1	-	-
TS	10.9	9.4	14.3	15.5	11.5	8.4	117.2

(c) Bookshelf Policy Cost

MoE			CHS	MPF		
Easy	Med	Hard	CHS	Easy	Med	Hard
31.2	7.8	4.0	3.0	35.9	7.8	23.7
265.2	43.3	3.8	2.7	221.8	595.5	24.1
675.9	95.6	95.0	15.0	14.8	42.4	23.9
-	-	-	1152.3	4333.4	-	-
994.7	293.3	121.2	131.9	1080.6	280.2	1025.7
-	-	-	475.0	1600.5	-	-
98.8	4.1	4.4	5.3	163.3	10.4	376.7

(d) Bookshelf Planning Times

	MoE			CHS	MPF		
	Easy	Med	Hard	CHS	Easy	Med	Hard
CM 1	7.6	7.1	9.3	9.3	8.2	7.6	14.7
OFU	15.1	15.1	15.3	15.2	6.7	8.2	14.7
MCBE	8.2	11.3	9.3	10.5	fail	15.4	14.7
TS	7.6	7.2	9.7	9.0	5.9	6.7	14.6

(e) RealTable Policy Cost

	MoE			CHS	MPF		
	Easy	Med	Hard	CHS	Easy	Med	Hard
CM 1	34.7	16.2	3.3	2.9	55.4	59.2	2.8
OFU	4.0	6.5	3.4	3.2	3.4	12.7	2.9
MCBE	63.4	158.7	68.5	78.4	fail	138.5	61.1
TS	7.7	2.8	2.0	1.7	22.0	33.7	3.5

(f) RealTable Planning Times

	MoE		CHS	MPF	
	Easy	Hard	CHS	Easy	Hard
CM 1	8.1	6.9	8.1	7.5	fail
OFU	14.8	14.8	14.8	7.5	fail
MCBE	6.9	6.9	8.3	7.5	fail
TS	8.5	12.7	12.7	7.5	fail

(g) Refrigerator Policy Cost

	MoE		CHS	MPF	
	Easy	Hard	CHS	Easy	Hard
CM 1	13.2	11.2	11.6	1.8	fail
OFU	3.3	5.0	2.8	1.7	fail
MCBE	54.6	85.8	88.7	25.0	fail
TS	11.4	5.5	3.9	1.7	fail

(h) Refrigerator Planning Times

Table 1: Results for simulated and real robot arm experiments using different belief models and strategies. "-" indicates the GPU memory was exceeded during the trial. Policy costs are in radians, times are in seconds. "fail" indicates the policy incorrectly believed there was no path to the goal.