Journal Title
XX(X):1–23
©The Author(s) 2021
Reprints and permission:
sagepub.co.uk/journalsPermissions.nav
DOI: 10.1177/ToBeAssigned
www.sagepub.com/
SAGE

The Blindfolded Traveler's Problem: A Search Framework for Motion Planning with Contact Estimates

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

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 be very inefficient, leading to many unnecessary collisions and unproductive motion. 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. The solution to a BTP is a policy indicating the next edge to attempt given previous observations and an initial belief. We prove that BTP is NP-complete and show that exact modeling of the belief is intractable, therefore we present several approximation-based policies and beliefs. For the policy we propose graph search with edge 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 through contact. This problem occurs 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 (Siegel et al. 1998)) or during everyday household tasks (e.g. reaching deep into a cabinet or behind a box (Park et al. 2014)). 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 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 PRM Kavraki et al. (1996), a graph where vertices represent robot configurations and edges represent potentially-valid movements of the robot between these configurations. In a typical PRM, edges are collision-checked using the known environment geometry, thus a planned collision-free path is guaranteed to be executed successfully. In our problem the environment geometry is unknown, thus the robot may collide during execution and be forced to replan. Existing approaches apply *Optimism in the Face of Uncertainty* (OFU) (Stentz 1994) — assume untraversed edges are valid, plan the shortest path and execute it. If the shortest path is indeed valid, the robot reaches the goal on the first attempt. Otherwise, it removes the invalid edge from the graph and replans. OFU is effective

[1] Univeristy of Michigan
[2] University of Washington

Corresponding author:
Brad Saund, University of Michigan, Ann Arbor, MI

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.

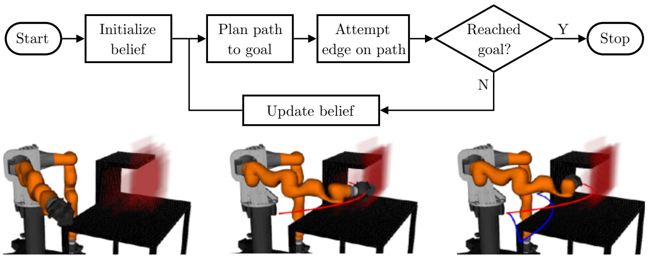


Figure 1. Overview of the BTP framework for planning with contact feedback. The robot is uncertain about location of the back wall. As it attempts to traverse edges, it partially localizes the wall and eventually finds its way to the goal.

in less-cluttered environments, where the robot finds a path to the goal after at most few collisions. However, on problems with narrow passages such as Fig. 1, OFU can lead the robot down a "rabbit hole" trying many 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 occupy significant regions of workspace, coupling even non-overlapping edges. Given a prior on edges, a robot can exploit such correlations to infer edge validities from a few measurements and reach the goal quickly. 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). A traveler has to optimally move from start to goal on a graph. They are blindfolded and can only know the validity of an edge by attempting to traverse it. Solving a BTP involves a belief model and a policy. The policy uses the belief to select the next edge for the traveler to attempt. The attempt yields an observation which in turn updates the belief. This cycle is repeated until the goal is reached Fig. 1. In this paper we show that BTP is NP-Complete and discuss a set of approximation-based policies.

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)
- Prove BTP is NP-complete and relate BTP to POMDPs and other existing problems (Section 4)
- Map the planning with contact feedback problem to a BTP. (Section 5)
- Adapt techniques for belief approximation for this planning with contact feedback task based on a particle filter (MPF), Collision Hypothesis Set (CHS), and a Mixture of Experts (MoE). (Section 6)
- Develop a set of approximation strategies to solve the BTP, and propose the Collision Measure (CM) policy. (Section 7)
- Provide empirical evaluation of different strategies and belief approximations on simulated and real robot arm BTP instances. (Section 8)

We evaluate all strategies and belief representations on a 7 DOF robot arm in multiple simulated and real scenarios. 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 formulating and solving the planning with contact feedback problem within the BTP framework significantly outperforms a baseline strategy that does not use BTP. This paper extends the work presented in Saund et al. (2019). Specifically we expand the BTP NP-completeness proof and connections to other problems, add analysis of strategies including regret analysis for the repeated version of BTP, provide more detailed descriptions of the belief methods, add an additional experiment, and expand the discussion on future improvements.

2 Related Work

2.1 Contact Sensing

The information gained from sensing a contact can vary drastically depending on the sensors and conditions involved. The most sensitive contact sensors such as SynTouch (Wettels et al. 2008) mimic fingers and provide force, vibration, and temperature. GelSight (Yuan et al. 2017) and the Soft-bubble grippers (Kuppuswamy et al. 2020) implicitly sense surface deformation from which shear and slip can be inferred. Such sensors provide rich feedback, but are expensive and bulky so while they are appropriate for an end-effector, it is currently impractical to coat an entire arm with such detailed sensors. Tactile skin for robots has been designed (Bhattacharjee et al. 2014), but since this adds cost, complexity, and more components with points of failure, no commercially produced robot arm has such skin. Furthermore, even if robot had such skin, the objects they grasp would not, and thus contacts between these objects and obstacles would be ambiguous.

It is instead appealing to infer contact from robot proprioception and joint torques. With sufficiently accurate torque sensing a Kuka iiwa or Franka robot can localize the point of contact during a dynamic collision (Haddadin et al. 2017). Such accurate localization requires robots with expensive highly-accurate torque sensors and precisely known masses, and that the contact torques and accelerations are sufficiently higher than the noise of the sensors. Furthermore, the contact point may be ambiguous when inferred from joint torques even when assuming no noise (Pang et al. 2021).

In this work, we assume a simple, low-information contact model with minimal requirements applicable to many robots. We follow the contact model from the Manifold Particle Filter (Klingensmith et al. 2016) and previous work on Collision Hypothesis Sets (Saund and Berenson 2018), where contact is a binary measurement with the additional information of which links may potentially be in contact. Note that while our problem formulation assumes static obstacles, movable obstacles are of great interesting in robotics. The belief over moving object poses can be modeled with the Manifold Particle Filter (though we do not in this work), other particle filter methods tailored to contact (Páll et al. 2018; Wirnshofer et al. 2019), or by scoring and selecting hypotheses over object poses with MCTS (Mitash et al. 2018).

2.2 Graph Search

Our problem is closely related to that of real-time motion planning on roadmaps (Kavraki et al. 1996). Roadmaps are graphs in configuration space. In robot motion planning, edge evaluation dominates computational complexity (Hauser 2015), therefore the key to minimizing search times is laziness (Bohlin and Kavraki 2000; Cohen et al. 2015). LazySP (Dellin and Srinivasa 2016), shown to be optimally lazy (Haghtalab et al. 2018), optimistically plans the shortest path and checks edges sequentially until an infeasible edge is encountered. Priors on edge validities can be further exploited to minimize edge evaluation (Choudhury et al. 2016; Mandalika et al. 2019; Narayanan and Likhachev 2017). These problems can be further mapped to Bayesian active learning (Tong and Koller 2001; Golovin et al. 2010; Chen et al. 2015) to compute policies that actively choose edges to evaluate to minimize uncertainty about which path is feasible (Choudhury et al. 2018, 2017). An alternate formulation is online shortest path routing (Awerbuch and Kleinberg 2004; György et al. 2007; Talebi et al. 2017), which is a particular instance of combinatorial bandits (Cesa-Bianchi and Lugosi 2012). However, unlike our problem, these methods have the ability to query an oracle to evaluate *any edge*. In BTP the agent must move to and then attempt an edge to discover the validity.

2.3 Planning Under Uncertainty

Our work falls under the domain of planning under sensing uncertainty. D* (Stentz 1994) and variants (Koenig and Likhachev 2002; Ferguson and Stentz 2007) typically replan optimistically and re-using the search graph. An alternative is to cast the problem in a Bayesian paradigm using an occupancy map (Richter et al. 2015). However, such methods usually plan to short horizons. Since this problem arises from the mobile robotics community, the focus is primarily robot safety (Janson et al. 2018). For our problem, the robot is able to collide safely and we seek to minimize the travel cost.

Many problems involving belief space planning, including ours, define an MDP with unknown initial state. For tractability, we must limit the dimensionality of the belief space (Ong et al. 2009), so while uncertainty can be caused by unknown robot motion (Lee et al. 2013), we consider only unknown obstacles. Given a belief, some approaches seek to find a single path with sufficient probability of success (Kimmel et al. 2019; Platt et al. 2010), though our BTP considers the full cost involving the replanned cost if the initial path fails.

The BTP problem is closely related to the Canadian Traveler's Problem (CTP) (Papadimitriou and Yannakakis 1991*a*) where neighboring edge costs are revealed when an agent visits a vertex. CTPs that have a directed acyclic graph structure can be solved exactly via dynamic programming (Nikolova and Karger 2008) but the general problem is PSPACE-complete (Fried et al. 2013). Typically CTPs are solved using heuristics (Eyerich et al. 2010) adopted from probabilistic planning (Yoon et al. 2008) or using Monte-carlo Tree Search (Gelly and Silver 2007; Guez et al. 2012). CTP can also be cast in a Bayesian framework (Lim et al. 2017) and solved near-optimally using informative path planning techniques (Lim et al. 2015, 2016), and which has been examined in the robotics context of 2D terrain navigation (Guzzi et al. 2019). While we evaluate some of these strategies for our robot arm planning problem, others are prohibitively expensive due to expensive collision checking and posterior updates. We therefore adapt the Collision Measure (Choudhury et al. 2016) as a computationally efficient strategy for the CTP/BTP.

3 Problem Statement

We propose the Blindfolded Traveler's Problem as a graph traversal 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*

The pure BTP is defined from a graph $\mathcal{G}$ with start and goal nodes, and a belief over edge blockage (x, η).

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 $x(e) = \{\text{BLOCKED}, \text{FREE}\}$ denote if the edge is invalid or valid. Note that $x(e)$ is *latent*, as the traveler is initially unaware of the validity of edges. 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:

$$\Gamma(v_1, e_{1,2}, x, \eta) = \tag{1}$$

$$\begin{cases} (v_2, \mathcal{W}(e_{1,2})) & x(e) = \text{FREE} \\ (v_1, 2\eta(e_{1,2})\mathcal{W}(e_{1,2})) & x(e) = \text{BLOCKED} \end{cases} \tag{2}$$

That is to say, traversing a valid edge moves the traveler to the new vertex v_2 with a traversal cost equal to the weight of the edge $\mathcal{W}(e_{1,2})$. 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, $2\eta(e_{1,2})\mathcal{W}(e_{1,2})$.

The traveler has a prior $\mathcal{P}$ on the joint probability over all edge validities and blockages $P(x, \eta)$. When attempting to traverse edge e, the traveler learns the true validity as well as the location of the blockage (if applicable) via the observation $o = (x(e), \eta(e))$. The prior combined with the observations can be used to inform the updated belief over edge validities and blockages. 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.

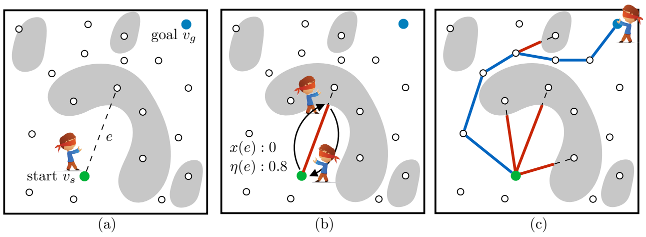


Figure 2. Blindfolded Traveler's Problem

The solution to the BTP is a policy π that defines the next action of the traveler, dependent on the prior and all previous observations. π can be defined as a policy tree, where nodes of this policy tree specify action (i.e. an edge on the BTP graph to attempt), and edges of this policy tree correspond to the observation the received by the traveler. In practice this tree may be represented implicitly.

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

$$\min_{\pi} \mathbb{E}_{(x,\eta)\sim\mathcal{P}} \left[c(\pi(x, \eta)) \right] \tag{3}$$

4 Analysis of BTP

Before considering solutions, we first analyze the complexity and relationships to existing problems. We show the BTP can be mapped to a POMDP, and thus perhaps it is no surprise that we then show BTP is NP-hard, and NP-complete under some conditions. Finally, we place BTP within the context of the existing Canadian Traveler's Problem as well as the common Shortest Path Problem

4.1 *Mapping the Problem to a POMDP*

The BTP problem maps to a Partially Observable Markov Decision Process (POMDP) specified by the tuple $\langle \mathcal{S}, \mathcal{A}, T, C, \mathcal{O}, Z \rangle$ defined below. This mapping connects concepts from the POMDP literature to the BTP, however the remaining secions of this paper follow the BTP notation and not this POMDP notation.

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 x and η are 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.

$$Q(b, a) = \mathbb{E}_{s\sim b}\left[C(s, a)\right] + \mathbb{E}_{b'\sim P(.|b,a)}\left[V^{\pi^*}(b')\right]. \quad (4)$$

4.2 Computational Complexity

We show BTP is NP-hard, and NP-complete when the prior is explicitly represented. In this analysis we examine the BTP decision problem instead of the optimally problem.

Definition 1. *The Blindfolded Traveler Problem decision problem is the question of whether there is a policy with expected cost less than or equal w.*

We follow an analysis parallel to Lim et al. (2017) 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 we assume a finite number of possible worlds and $\mathcal{P}$ enumerates the probability of each possible world. Note that this explicit representation does not cover common cases such as continuous beliefs over η.

Theorem 1. *The decision version of BTP is in NP when the belief $\mathcal{P}$ is expressed explicitly.*

Proof 1. *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 thus represent traversing several known edges in $\mathcal{G}$ to reach the unknown edge. Each edge of the policy tree corresponds to an observation o received upon traversing an unknown edge. A BTP is solved by traversing the policy tree until the leaf node is reached, i.e. evaluating unknown edges and receiving 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, each hypothesis world in $\mathcal{P}$ yields a unique set of observations and thus a unique path through the policy tree. Since we assume each hypothesis world is explicitly represented in $\mathcal{P}$, the size of 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. Since the solution policy tree can be verified in polynomial time the BTP decision problem is in NP.

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 or parametric representations, then the problem may no longer be in NP. For example, a factored representation may generate exponentially more possible observations than the input size, thus the policy tree could be larger than polynomial size.

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 $\mathcal{H} = (h_1, h_2, \ldots, h_n)$ and a finite set of tests $\mathcal{T} = (t_1, t_2, \ldots, t_m)$. 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 fewest number of tests when h^* is uniformly randomly selected from $\mathcal{H}$. 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 (Laurent and Rivest 1976).

Our reduction maps tests to cheap information-gathering edges (left side $Fig.$ 3) which inform the traveler which one of the many expensive goal-seeking edges (right side $Fig.$ 3) to attempt. For this proof, consider the simplified BTP "sBTP" with discrete b fixed at $\eta(e) = 1$ (i.e. an agent must traverse an entire edge to learn the validity).

Theorem 2. *The decision version of sBTP is NP-hard.*

Proof 2. *ODT is polynomial time reducible to sBTP and thus sBTP is NP-hard. Given an instance of ODT$(\mathcal{H}, \mathcal{T})$, we consider a specific instance of sBTP $\langle \mathcal{G}, \mathcal{P}, v_s, v_g \rangle$ as follows. Consider the sBTP problem shown in Fig. 3. The cluster of edges $\{e_1, \ldots, e_m\}$ correspond to the tests $\{t_1, \ldots, t_m\}$. Note again that in sBTP 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 that corresponds to identifying the correct hypothesis from $(h_1, h_2, \ldots, h_n)$. The weight is 1 for each edge in the left*

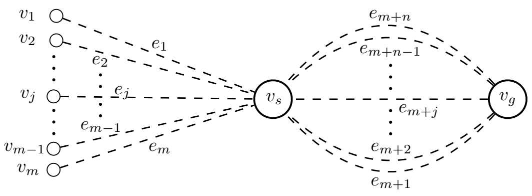


Figure 3. Reduction from Optimal Decision Tree problem

cluster $\{e_1, \dots, e_m\}$ and is $2m$ for each edge in the right cluster.

We set 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 $j = \{1, \dots, m\}$, 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 $j = \{m, \dots, m+n\}, j \neq i$. We now argue that the expected cost of this ODT instance is less than or equal to some value w iff the cost of this sBTP 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 sBTP is $\leq 2w + 2m$.

Next, we prove the converse that if the cost of the sBTP is $\leq 2w + 2m$, then the cost of the ODT is $\leq w$. Note that $w > m$ is impossible because the ODT is clearly solved by, at worst, evaluating all tests, which would incur cost m. Thus we consider $w \leq m$ which implies the cost of the sBTP 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 sBTP 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 sBTP in polynomial time, and since ODT is known to be NP-hard then sBTP is also NP-hard.

Finally, since sBTP is a BTP, the full BTP problem is NP-hard. Since we also showed BTP is in NP (with an explicit $\mathcal{P}$), BTP is therefore NP-complete.

4.3 Relation to the Bayesian Canadian Traveler's Problem

The BTP is closely related to the Canadian Traveler's Problem (CTP) Papadimitriou and Yannakakis (1991*b*). The name "Canadian Traveler" is motivated by roads that have been randomly snowed over and a driver able to peek down roads at intersections to see if they have been plowed. As in BTP, CTP defines a graph with unknown edge validities with a goal-seeking agent traversing this graph. Unlike in BTP, the CTP agent learns the validity of *each* edge adjacent to the current node.

Both BTP and CTP are part of a family of graph traversal problems where an agent executes a policy to reach a goal with the minimum expected cost. We call this family the k-lookahead graph traversal problem, where an agent only observes the true validity of edges within k steps of its location. The Shortest Path Problem over known graphs is an instance of ∞-lookahead, as the agent can query any edge on the graph regardless of distance. 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 necessitates the more complicated cost formulation.

In the original CTP the probability of edge validities $x(e)$ are independent. In the more general Bayesian CTP (BCTP) (Lim et al. 2017) $x(e)$ are correlated through beliefs of underlying worlds ϕ rather than beliefs directly over x. As defined, the BTP is analogous to the Bayesian CTP, and in the contact-feedback instance of BTP the workspace obstacles workspace obstacles create correlations between elements of $x(e)$.

5 Contact-feedback Planning Problem as an instance of BTP

For the remainder of the paper we examine the specific BTP of a robot arm planning with unknown workspace obstacles sensed only through contact. The key to mapping to a BTP is that the graph $\mathcal{G}$ is a roadmap and the belief over edge blockage (x, η) is generated from a belief over obstacles.

The robot operates in a workspace W containing workspace obstacles W_{obs}. The robot's configuration space $\mathcal{C}$ is composed of free space $\mathcal{C}_{free}$ and obstacles $\mathcal{C}_{obs} = \mathcal{C} \setminus \mathcal{C}_{free}$, defined from these workspace obstacles. The links $\mathcal{L}$ of the robot at configuration $q \in \mathcal{C}$ occupy a workspace volume $\mathcal{R}(q, \mathcal{L}) \subset W$. As an abbreviation, let $\mathcal{R}(q) = \mathcal{R}(q, \mathcal{L}_{all})$, where $\mathcal{L}_{all}$ is the set of all the robot links. 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. In this work we consider the straight line paths between vertices with weighting $\mathcal{W}(e) = ||e(0) - e(1)||$, although other edge-weighting schemes could be substituted with no change in the method. An edge represents the swept volume $W_e = \cup_{d\in[0,1]} \mathcal{R}(e(d))$, which is calculated in practice by discretizing configurations along the edge. 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 i exceeds its threshold τ_i^{th}. A successful edge traversal results in $o = (\text{FREE}, 1)$, while a collision yields $o = (\text{BLOCKED}, \eta)$ where $e(\eta) = q_{col}$.

Furthermore, as a slight augmentation of BTP, a collision yields additional information about which links could possibly be in contact. 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 $\mathcal{L}_{contact}$. Recall that $\mathcal{R}(q, \mathcal{L}) \subseteq \mathcal{R}(q)$ is 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 $\mathcal{R}(q)$.

Note that this model for contact observation provides less information than may be expected. Specifically, a contact observation includes which links may have collided at a specific configuration, but *does not include the contact point*. This formulation models robots without touch-sensitive "skin", and a high collision detection sensitivity with τ^{th} set just above the noise in the torque measurements.

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 a simple prior over workspace occupancy yields correlation amongst edges. The robot can exploit this to gain information about untraversed edges. Second, it is 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, and a manually-specified prior might not model the true robot's world. How should the robot detect and compensate for this in a principled manner? Section 6 addresses construction of priors with good properties and belief updates based on contact observations.

The solution to the contact-feedback BTP is still a policy π, yet it is impractical to represent π with an explicit policy tree. In the solutions we consider, the traveler maintains a belief over world occupancy which induces a belief over edges. We consider strategies in Section 7 that can both calculate the probability of any edge validity and sample worlds using the traveler's belief.

6 Belief Representations for Contact-based Planning

With the BTP defined two challenges remain in instantiating a solution for the robot contact-planning problem. This section addresses beliefs: how to represent a prior over world occupancy, and how to update this belief given a contact measurement. In the next section we examine strategies which use this belief to solve a BTP.

6.1 *Belief and Observation framework*

An agent maintains a belief over workspace occupancy W_{obs}, which we refer to as a world $\phi \in \Phi$. In this work we represent a world as a voxel grid. 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.

The belief at timestep t is represented as $b_t(\phi)$, and is updated by contact observations. As discussed in Section 5, contact observations do not indicate the specific point in contact, but rather that the contact occurred in a region tangent to robot links $\mathcal{L}_{contact}$ at a configuration q.

We follow three approaches for maintaining the belief. The first (MPF) is a non-parametric particle filter where a set of candidate hypotheses are maintained, updated and possibly eliminated. The second approach (CHS) is initially optimistic and adds the minimal new hypotheses needed to explain the contact measurements. Finally, combine these two approaches (MoE).

6.2 *Approach 1: Manifold Particle Filter (MPF)*

A particle filter is a non-parametric Bayes filter that approximates the belief $b_t(\phi)$ as a finite set of possible candidate worlds $\Phi_t = \{\phi_t^1, \phi_t^2, \dots\}$ with associated weights $\{\mu_t^1, \mu_t^2, \dots\}$ (Thrun et al. 2005). Robot actions update each particle according to the process model, mapping states and actions to next states. Each observation updates the belief by reweighting and resampling particles. A conventional particle filter performs measurement updates via importance sampling: weighing each particle by observation likelihood

$\mu^i_t = P(o_t|\phi^i_t)$, and resampling according to these importance weights. In this paper, the each particle models objects with known geometry but with unknown 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. This issue is directly caused by the conventional importance sampling measurement update. In the case of a highly discriminative measurement such as a contact, the target distribution represents a thin manifold of possible object configurations which does not match the proposal b_{t-1}. This leads to situations where the measurement likelihood is near 0 for all particles and also nearly all of the weight is given to a few particles. Resampling repeatedly samples these few particles, causing the particle filter belief to lose variety and become a bad approximation of the true belief. This is known as particle starvation.

We therefore adopt the strategy used in the Manifold Particle Filter (MPF) (Klingensmith et al. 2016), depicted in Fig. 4 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. The importance weights are then calculated using $P(\phi^i_t|b^i_{t-1})$ (Line 10). 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 (Klingensmith et al. 2016) which approximates the proposal distribution by projecting the prior particles onto the contact manifold (Line 9). 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 of projection 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.

```
Algorithm 1: Manifold Particle Filter
input  : Prior particles: $\Phi_{t-1}$
         Edge traversed: $e$
         Observation: $o_t = (x_t, \eta_t)$
         Links possibly in contact: $\mathcal{L}_{contact}$
output: Posterior 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 // discretized
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 = $ BLOCKED 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
            ($\Phi_{t-1}, \phi^i_t$)
```

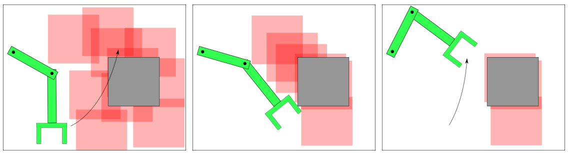


Figure 4. Manifold Particle Filter: The initial particles Φ_0 model configurations of the true obstacle before the robot moves (top). A collision during a motion causes particles to be resampled on the contact manifold (middle). Subsequent free space motions sweep through and eliminate some particles (bottom).

Consider the case where the MPF models the correct object but with a low probability of the correct location. Another common and more difficult instance occurs when the MPF model the incorrect object geometry, so no particle is capable of representing the true world. We address these limitations in our second approach.

6.3 Approach 2: Collision Hypothesis Sets (CHS)

To overcome the reliance on an accurate prior we can adopt the Collision Hypothesis Set (CHS) belief (Saund and Berenson 2018). The CHS method is composed of an initial assumption of free space, retaining the exact information gained from contact (under the model describe in Section 5), and assumptions on calculating the probability of occupancy.

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. 5 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

Algorithm 2: Collision Hypothesis Set

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

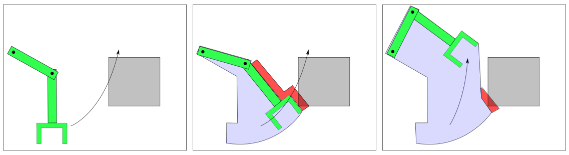


Figure 5. CHS: The robot initially plans a motion optimistic about unknown space (top). A motion sweeps out free space (blue) and a collision generates a CHS (middle). Future free space motion sweeps out more free space, potentially shrinking CHSs (bottom).

possibly in collision (Line 5). The known free space is then removed from all CHSs (Line 7).

With two assumptions, $\mathcal{K}$ induces a belief of validity for an edge $P(x(e))$. First, the optimistic assumption that each κ_i contains exactly one occupied voxel chosen uniformly yields that for a single CHS κ_i the probability of blockage is the fraction of κ_i in the swept volume of the edge W_e:

$$P(x(e) \text{ is BLOCKED}|\kappa_i) = \frac{|W_e \cap \kappa_i|}{|\kappa_i|} \tag{5}$$

The second assumption that each κ_i is independent of other κ_j means the probability of edge validity for an entire $\mathcal{K}$ can be easily computed:

$$\begin{aligned} &P(x(e) \text{ is FREE}|\mathcal{K}) = \\ &\quad \prod_i 1 - P(x(e) \text{ is BLOCKED}|\kappa_i) \end{aligned} \tag{6}$$

Note that CHS method never marks a valid edge as invalid. Additionally note that an attempt of an invalid edge e generates $\kappa_i \subset W_e$, thus the CHS belief correctly marks the edge as invalid: $P(x(e) = \text{FREE}|\kappa) = 0$.

The optimistic (5) and independence (6) assumptions mean that the CHS method is optimistic about free space. Sampling a world $\phi \sim \mathcal{K}$ yields only a few occupied voxels (one for each CHS), not representative of realistic scenarios, though as a single voxel still blocks an edge, the edge validities x may still match realistic scenarios. However, unlike a particle filter with good initialization, it may take many collisions to build up a sufficient $\mathcal{K}$.

6.4 Approach 3: Mixture of Experts (MoE)

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 in the MPF to congregate in regions predicted by prior Φ_0 then the prior likely provides a reasonable model of the world. If instead particles update to regions that were unlikely under the prior, or if particles disappear entirely, then the prior was likely a poor guess about the underlying world, 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:

$$b_t(\phi) = \frac{\beta_t^{MPF} b_t^{MPF}(\phi) + \beta_t^{CHS} b_t^{CHS}(\phi)}{\beta_t^{MPF} + \beta_t^{CHS}} \tag{7}$$

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:

$$\beta_t^{CHS} = 1 \tag{8}$$

$$\beta_t^{MPF} = \mathbb{E}_{\phi \sim b_t^{MPF}} \left[\frac{P(\phi|b_0^{MPF})}{P(\phi|b^U)} \right] \tag{9}$$

$$= \sum_{\phi_t^i \in \Phi_t} \mu_t^i \frac{P(\phi_t^i|b_0^{MPF})}{P(\phi_t^i|b^U)} \tag{10}$$

$$= \sum_{\phi_t^i \in \Phi_t} \mu_t^i \frac{b_0^{MPF}(\phi_t^i)}{1/V} \tag{11}$$

$$= V \sum_{\phi_t^i \in \Phi_t} \mu_t^i b_0^{MPF}(\phi_t^i) \tag{12}$$

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 a good model of the world, there are at least some particles that have both a high weight after updating from measurements μ_t^i and high probability 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, thus CHS and MPF are equally weighted when the current MPF particles are equally probably under the initial b_0^{MPF} as a uniform prior.

Consider MoE applied to the scenario of Fig. 4 and Fig. 5. Initially, MPF particles are sampled from b_0^{MPF}, thus $\beta_0^{MPF} > 1$. After the two robot motions the updated particles Φ_2 (Fig. 4 right pane) still are more likely under b_0^{MPF} than under a uniform prior, thus $\beta_2^{MPF} > 1$, and the particle filter continues to dominate the MoE belief. If, however, the updated particles Φ_2 were, say, on the lower left portion of the pane then Φ_2 would be far more likely under a uniform prior, and the MoE belief would "fall back" to using the CHS belief.

7 Strategies for Solving the BTP

Since BTP is NP-complete (Section 4.2), there is no efficient optimal solution to an arbitrary Blindfolded Traveler's Problem. Instead we seek to understand the conditions of the contact planning instance of BTP to suggest practical approximate solutions.

We explore a number of computationally efficient approximation strategies to solve the problem, by drawing from heuristics used in the related Canadian Traveler's Problem (CTP) (Eyerich et al. 2010) (Section 7.2). We also propose the heuristic of Collision Measure (CM) (Section 7.1) that (to the best of our knowledge) has not been applied to a CTP. Detailed analysis and guarantees of these strategies applied to BTP is provided in Appendix A. Broadly, each strategy employs one or more of the ideas: approximating the cost-to-go, simulating actions in sampled worlds, policy rollouts, and exploration-exploitation tradeoffs.

Each strategy defines a policy at time t for an agent at vertex v_t and must decide which edge e_t from the set of outgoing edges $\mathcal{N}(v_t)$ to traverse. Naturally, the edge chosen will depend on the current belief b_t, which is determined by the initial belief b_0 (i.e. prior $\mathcal{P}$), the history of observations ψ_t, and the update procedures of the previous section.

7.1 Collision Measure (CM)

The CM heuristic balances exploration (assuming unexplored edges are free) with exploitation (penalizing edges with low validity likelihoods). The agent chooses the next edge to traverse as follows:

$$\begin{aligned}\widehat{\mathcal{G}} &= (\mathcal{V}, \mathcal{E}, w(e) - \alpha \log P(x(e) = \text{FREE}|b_t))\\ e_t &= \left\{ e \in \mathcal{N}(v_t) \;\middle|\; e \in \textsc{ShortestPath}(\widehat{\mathcal{G}}, v_t, v_g)) \right\}\end{aligned} \tag{13}$$

$\widehat{\mathcal{G}}$ is an optimistic graph containing all the edges of $\mathcal{G}$, with 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 $(P(x(e) = \text{FREE}|b) = 0)$ yields a weight of ∞, and a known free edge $(P(x(e) = \text{FREE}|b) = 1)$ yields $w(e)$. At each iteration the CM strategy finds the shortest path over $\widehat{\mathcal{G}}$ and attempts the first edge.

CM is complete - If a path to the goal exists, CM will reach the goal. CM will never traverse a known-blocked edge so each edge traversed along any SHORTESTPATH will either be known free, or unknown. By traversing an unknown edge, the agent learns if it is blocked or free. If the edge traversed is known free, traversing the edge provides no information and does not update the belief nor $\widehat{\mathcal{G}}$, thus the agent will continue along the same SHORTESTPATH during the next iteration. If all edges in SHORTESTPATH are known free, the agent will reach the goal. If not, the agent will learn about a new edge. There are a finite number of edges, thus if a path exists the agent will eventually reach the goal.

7.2 Baselines

To benchmark our proposed Collision Measure strategy we consider three categories of strategies commonly used in POMDPs – approaches that approximate the optimal 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. These strategies are discussed in detail in Section A.

Optimism in the Face of Uncertainty (OFU) (Brafman and Tennenholtz 2002): Find the shortest path on the optimistic graph and move along the edge on it.

Thompson Sampling (TS) (Littman et al. 1995): Sample a world from the current belief, find the shortest path in that world, and move along the edge on it.

QMDP (Littman et al. 1995): 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) (Eyerich et al. 2010): Sample a world from the current belief, simulate moving along an edge and rollout with an optimistic policy. Repeat to build a Q-value estimate. Move along the edge with best Q-value.

Upper Confidence Tree (UCT) (Gelly and Silver 2007): Conduct a Monte-Carlo Tree Search (Kocsis and Szepesvári 2006) 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 (Saund and Berenson 2018): 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.

7.3 Pitfalls for Strategies

Since all strategies considered are heuristics, it is important to recognize the pitfalls that they face. We illustrate these in Fig. 6. OFU is easily tricked into exploring cul-de-sacs that do not lead to the goal (Fig. 6(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. 6(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. 6(c). The initial belief is that the solid edge is known to be feasible while only one of dotted edges is feasible. When the agent is at v_1 it wishes to move to v_2 and vice-versa because of the (incorrect) assumption that the validity of all dashed-red edges will become known and thus the optimal path to the goal will become clear.

CM is also susceptible to pitfalls because it treats the probability of collision for each edge independently. Fig. 6(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 first attempt the lower (invalid) 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. 6(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.

8 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. Each experiment involves defining a scenario with various known and unknown obstacles, then selecting a belief model and a strategy. The strategy uses the belief to select the next edge for the traveler to attempt. The attempt yields an observation which in turn updates the belief. This cycle is repeated until the goal is reached (Fig. 1).

8.1 Implementation Details:

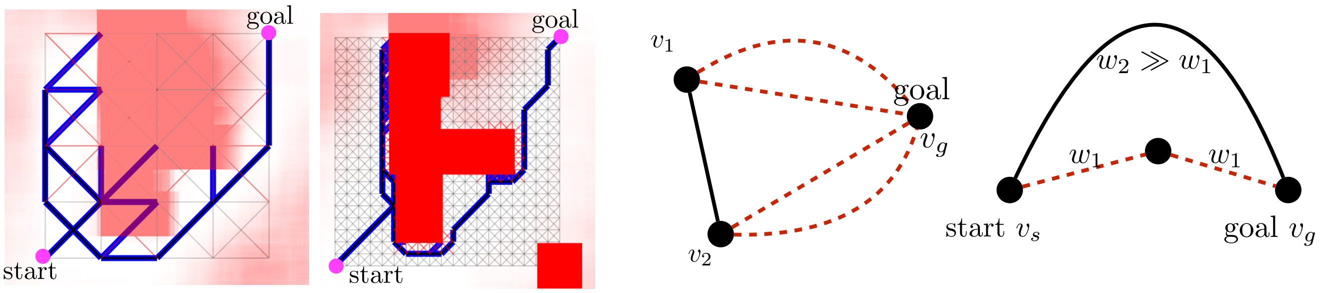
The workspace W is represented by a 200x200x200 voxel grid implemented on the GPU using GPUVoxels (Hermann et al. 2014). Computing $P(x(e)|\psi)$ involves the expensive computation of swept volumes W_e, approximated by discretizing the configurations with a distance of 0.02 rad. For efficiency we lazily 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 7 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 (Dellin and Srinivasa 2016) to minimize the number of expensive edge-evaluation operations. All trials were conducted on an i7-7700K with a NVidia-1080Ti GPU.

Joint torque feedback provided the blockage observation for each edge traversed, as described in Section 5. The Kuka iiwa controller calculated the expected torque τ^{exp} and the observed torque for each joint τ^{meas}. We manually set thresholds for each joint to be above the noise we observed to avoid false triggers. As the robot moved a controller monitored these torques and issues a “Stop” command to the arm if the threshold of any joint was exceeded. The blockage η was calculated based on the fraction of the edge traversed before contact, though in practice the blockage does not need to be computed explicitly. The position of the arm when contact is observed is used to update the beliefs.

8.2 Scenarios

We considered 2 real robot scenarios - `Refrigerator` and `RealTable`. In `Refrigerator`, Victor must reach into a refrigerator from behind (Fig. 7). In `RealTable`, Victor must move from below the table to above (Fig. 8). We also consider 3 simulated robot scenarios (Fig. 9) - `Bookshelf`,



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

Figure 6. Pitfalls for various strategies for a 2D BTP problems. In (a) and (b) the only paths to the goal lie to the lower right of the red obstacles, but because of the uncertainty over obstacle location (shown with transparency) OFU and ORO first attempt to traverse the upper left (blue edges). In (c) and (d) exactly one of the dashed red edges is feasible.

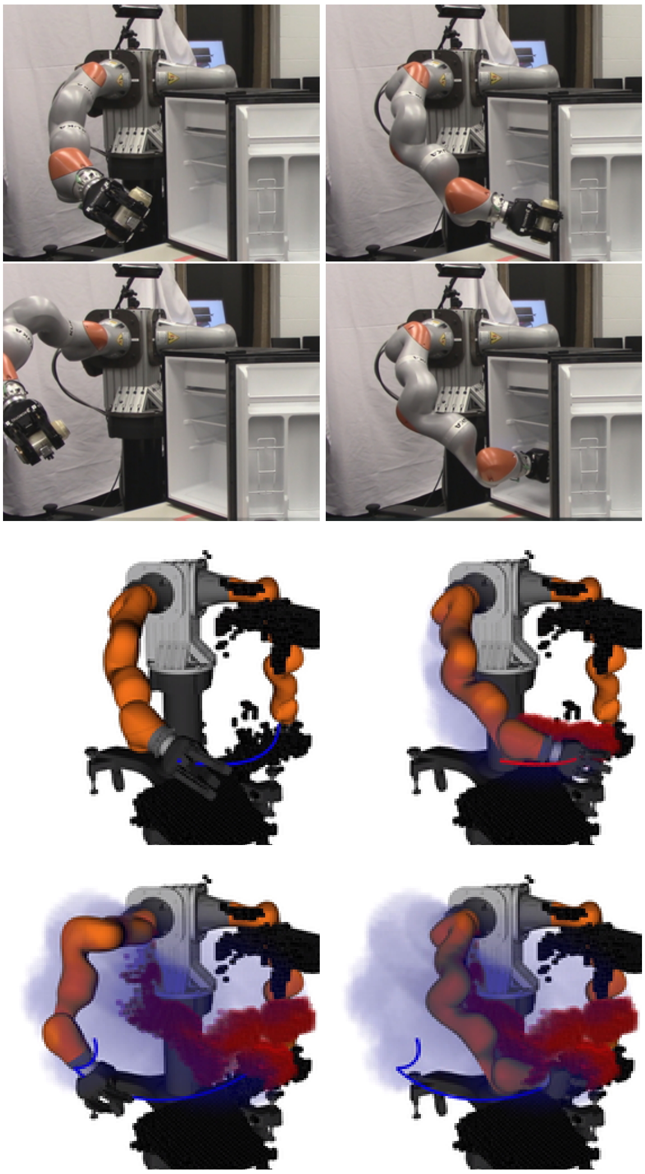


Figure 7. `Refrigerator` - Victor moving to place an object inside a refrigerator.

`Box`, and `cul-de-sac`. 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. In `cul-de-sac` Victor can entered an unseen U-shaped trap.

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 `Easy` and `Medium` scenarios the true obstacles are within a reasonable likelihood of the initial belief. In the `Hard` scenarios the true unknown obstacles are not only a large distance away from the belief obstacles, but the true obstacles are of a different physical shape. From a Bayesian perspective we should not expect methods that rely on the `Hard` prior to perform well, however from the perspective of roboticists we desire algorithms that can recover from incorrect perception. The motivation for the `Hard` scenarios is a robot that has correctly identified a chair in the corner, but did not assume the existence of the true unknown obstacles. These `Hard` scenarios illustrate the pitfalls of relying entirely on a bad prior.

In the real robot scenarios the `Easy` and `Medium` particle priors were manually generated, approximating 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 6 and all strategies from Section 7, 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

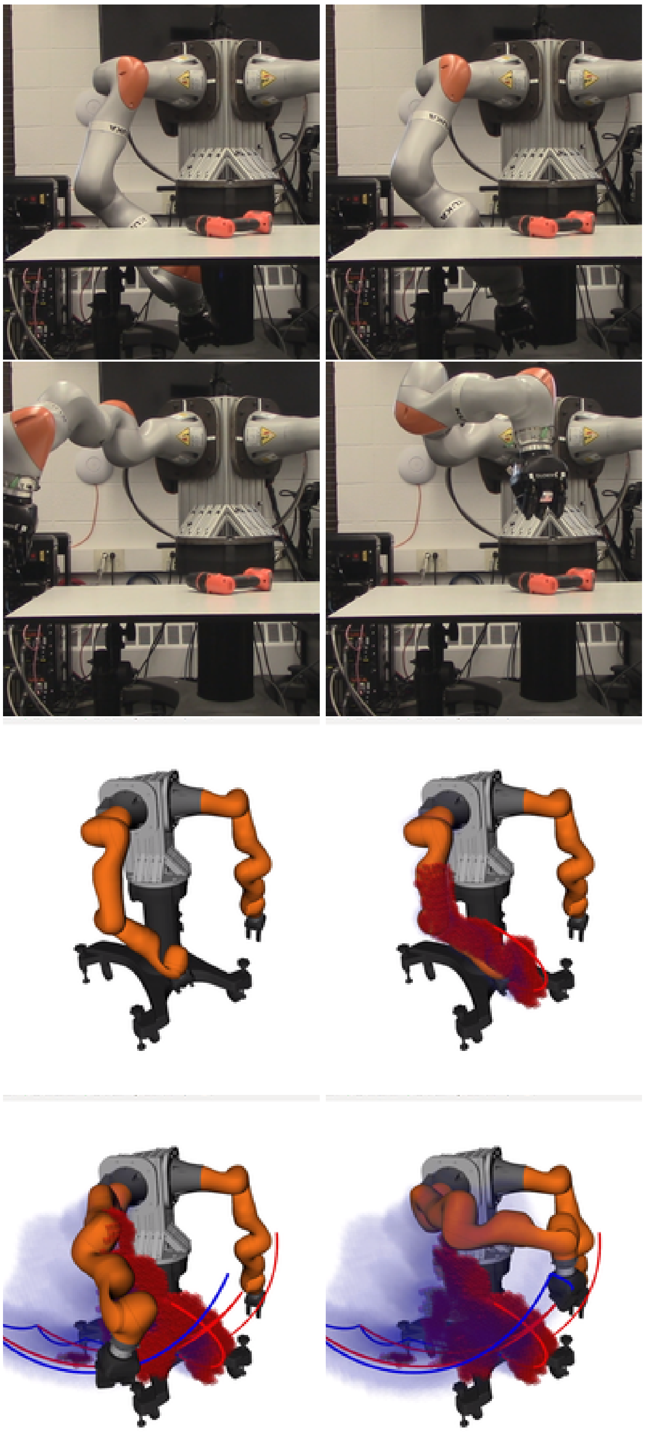


Figure 8. `RealTable` - Victor moving from below to above a table.

CM with $\alpha = 1$ and $\alpha = 10$, labeled `CM 1` and `CM 10`. Increasing α increases the cost of potential collisions and causes CM to prefer longer more conservative paths. We also compare against the (non-BTP) baseline proposed in (Saund and Berenson 2018) which interleaves an RRT with a local controller to find low cost paths through $\mathcal{C}$.

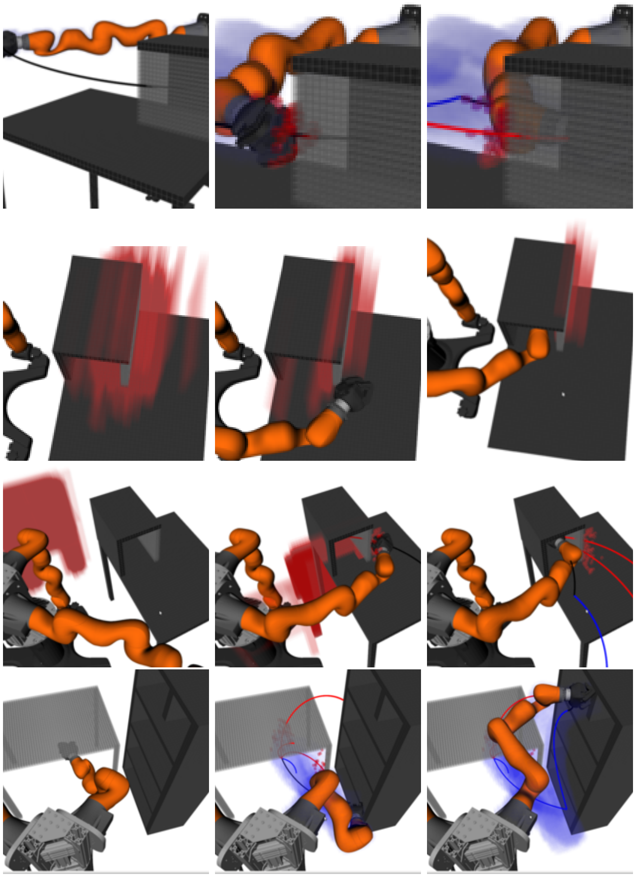


Figure 9. Simulation scenarios using the CM strategy, showing known objects (black), unknown objects (grey), swept known freespace (blue), and the belief of occupancy (red). Panes show the initial (left), first contact (middle), and completion (right)
Row 1: `Box` using CHS belief
Row 2: `Easy` setting of `Box` using MPF belief
Row 3: `Hard` setting of `Box` using MoE belief (the prior is far from the true unknown obstacles)
Row 4: `Bookshelf` using CHS belief.

8.3 Results:

Selected results for the `Bookshelf` scenario are shown in Fig. 10 with full results for all scenarios shown in Table 1. Many of the strategies are deterministic under a given belief. QMDP and MCBE are stochastic, but internally average over many samples and so each entry represents a single trial. Each non-BTP baseline result is averaged over 20 trials and each TS result is averaged over 10 trials. From these results we draw conclusions on the effectiveness of the applicability of BTP, the choice of belief model, and the choice of strategy.

BTP is effective at modeling the contact-feedback planning problem. For the non-BTP baseline (Saund and Berenson 2018) applied to the `Bookshelf` scenario we observe only 2 out of 20 trials succeeded within a 15 minute time limit. Compared to the previous baseline (Saund and Berenson 2018), we observe a significant improvement

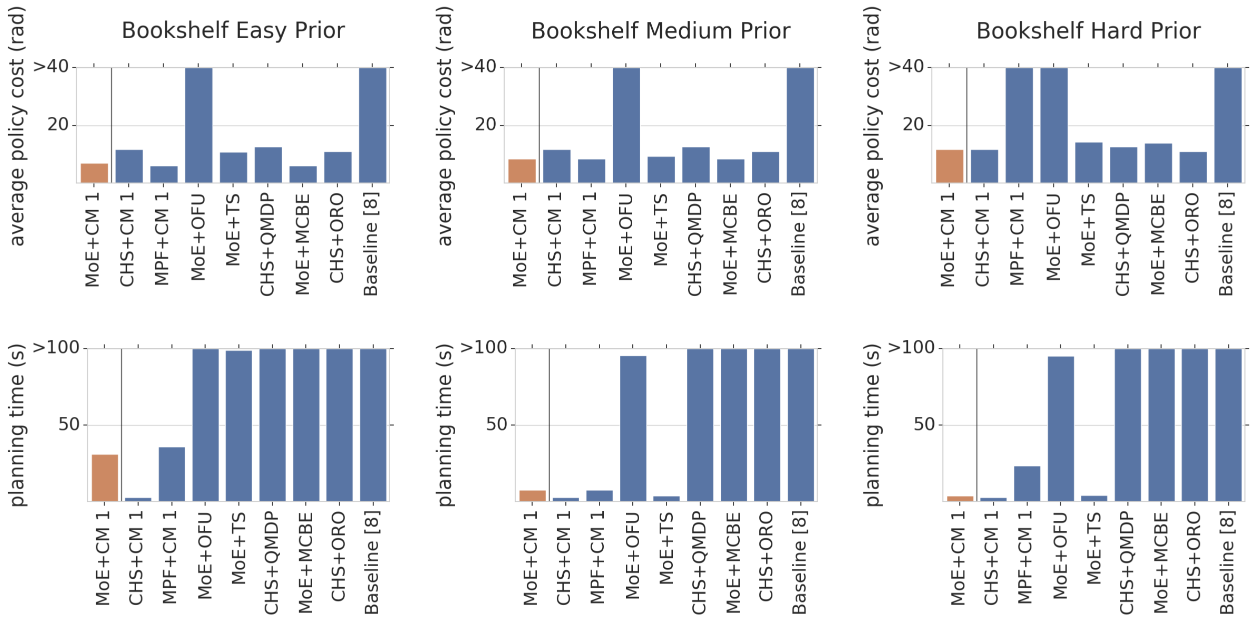


Figure 10. Results of applying various belief strategies and policies to the Bookshelf BTP. Our proposed MoE+CM is consistently fast and solves the BTP with low cost.

using the BTP framework. The baseline replans using an RRT after each collision, and therefore does not reuse planning efforts. BTP constrains the motion to a roadmap, yielding a manageable action space and depth for the search for the strategies proposed. 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. Furthermore, in BTP a collision eliminates an edge, reducing the number of possible paths, which does not happen in the baseline.

The Mixture of Experts belief representation gains the benefits of MPF and the robustness of CHS. We find all strategies across all scenarios perform better using the Easy MPF belief than CHS (with one exception in the RealTable). This is unsurprising, since the true obstacles are likely given the Easy MPF prior. Under the Hard MPF prior the true obstacles are not representable by the prior, and we find the strategies often have high costs or fail entirely. The CHS belief is agnostic to the prior so performs identically under Easy, Medium and Hard.

In our experiments for a given strategy and scenario the MoE belief yields policy costs that are approximately the minimum of using either the CHS or MPF belief alone. When using the Easy MPF prior MoE gains the benefits from the accurate knowledge of obstacles. When using the Hard MPF prior MoE down-weights the incorrect particle prior after the first contact, and achieves the robustness of the CHS prior. For example, in the Bookshelf scenario using CM 1 and the Easy prior MoE achieves policy cost of 7.0 which is only slightly worse than MPF's 6.1. Under the Medium prior MoE achieves the MPF's cost of 8.4. Under the Hard prior MPF performs poorly with a cost of 117.2, while MoE achieves the CHS cost of 11.7.

Both the Thompson sampling and the proposed Collision Measure strategies perform similarly well in our experiments. First note some strategies perform poorly on BTP. **O**ptimism in the **F**ace of **U**ncertainty performs decently under the Easy MPF prior, but is significantly worse than other strategies on all other beliefs. This is because OFU will attempt edges that the belief correctly reasons are almost certainly in collision. **O**ptimistic **RO**llout, QMDP, and MCBE in principle can achieve low policy cost, but as these strategies involve simulations of motions, contacts, and belief updates (for ORO) in many sampled worlds the planning time is prohibitively large.

Both **T**hompson **S**ampling and **C**ollision **M**easure perform similarly, with CM achieving slightly lower policy costs.

9 Discussion

Better Belief Model: We examined two belief models, each with limitations. The CHS belief performs no reasoning over object shapes, and the MPF belief model assumes the object geometry is known perfectly and only the position is unknown. In simulation it is possible to draw worlds exactly from this MPF prior, but to better model the real world we would like a prior over object shape and pose conditioned on the other robot observations, such as the kinect images. Clearly humans have learned a similar prior, as when you

	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	6.8	25.0	25.0	25.0	6.8	52.2	52.2
CM 10	6.8	7.0	19.6	19.6	6.8	7.0	52.2
OFU	46.2	46.2	46.2	46.2	58.0	52.2	52.2
ORO	-	-	-	38.1	6.9	-	-
MCBE	6.8	35.7	61.1	38.9	6.8	51.6	51.6
QMDP	-	-	-	31.6	7.2	10.3	-
TS	11.3	14.9	30.4	38.4	13.5	35.5	52.2

(e) cul-de-sac Policy Cost

	MoE			CHS	MPF		
	Easy	Med	Hard	CHS	Easy	Med	Hard
CM 1	24.1	12.3	11.2	11.1	14.3	3.3	2.4
CM 10	87.6	1454.3	2.4	2.5	53.2	1401.0	2.4
OFU	12.6	14.0	13.4	17.4	3.3	2.5	2.5
ORO	-	-	-	37287.5	2703.6	-	-
MCBE	582.5	4077.8	779.7	313.5	514.9	2760.9	112.8
QMDP	-	-	-	7489.6	1933.5	21870.3	-
TS	14.8	1298.7	3.2	4.5	21.5	640.7	5.0

(f) cul-de-sac 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

(g) 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

(h) 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

(i) 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

(j) 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.

see a new table for the first time, your brain completes the occluded region. Open challenges are how to learn such a prior given sufficient examples, and how to update such a belief model from contact measurements.

Leaving the graph: Modeling the Contact Planning Problem as a BTP restricts the robot motion to a graph. This simplifies the analysis and the decision of the next robot action to attempt, but relaxing this restriction could likely improve robot performance. For example one might desire a policy that stays in contact with an object, sliding along the edges while progressing towards the goal. To approximate this in the BTP framework, after a collision a node could be added to the graph at the contact configuration. This would allow the robot more opportunities to shortcut rather than forcing it to backtrack. Taken to the extreme, a robot could follow the surface of an object through repeated contact and node addition. A challenge would then be to avoid a endless cycle of node addition. Similarly, restricting the motion to a graph limits the contact sensing. After a contact we might desire the robot to wiggle as a cheap motion to gather information, as previously applied to robot hands (Koonjul et al. 2011).

10 Conclusion

We proposed the Blindfolded Traveler's Problem as a class of problems in planning under uncertainty and proved that it is NP-complete. We showed that contact-feedback 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.

References

Agrawal, S. and Goyal, N. (2013), Further optimal regret bounds for thompson sampling, *in* ‘AISTATS’, Scottsdale, AZ.

Awerbuch, B. and Kleinberg, R. (2004), Adaptive routing with end-to-end feedback: Distributed learning and geometric approaches, *in* ‘ACM symposium on Theory of computing’.

Bhattacharjee, T., Grice, P., Kapusta, A., Killpack, M., Park, D. and Kemp, C. (2014), ‘A robotic system for reaching in dense clutter that integrates model predictive control, learning, haptic mapping, and planning’, *IROS* .

Bnaya, Z., Felner, A. and Shimony, S. (2009), Canadian traveler problem with remote sensing., Pasadena, CA, pp. 437–442.

Bohlin, R. and Kavraki, L. E. (2000), Path planning using lazy PRM, *in* ‘ICRA’.

Brafman, R. I. and Tennenholtz, M. (2002), ‘R-max-a general polynomial time algorithm for near-optimal reinforcement learning’, *Journal of Machine Learning Research* .

Cesa-Bianchi, N. and Lugosi, G. (2012), ‘Combinatorial bandits’, *Journal of Computer and System Sciences* .

Chapelle, O. and Li, L. (2011), An empirical evaluation of thompson sampling, *in* ‘NIPS’.

Chen, Y., Javdani, S., Karbasi, A., Bagnell, D., Srinivasa, S. and Krause, A. (2015), Submodular surrogates for value of information., *in* ‘AAAI’.

Choudhury, S., Dellin, C. and Srinivasa, S. (2016), Pareto-optimal search over configuration space beliefs for anytime motion planning, *in* ‘IROS’.

Choudhury, S., Javdani, S., Srinivasa, S. and Scherer, S. (2017), Near-optimal edge evaluation in explicit generalized binomial graphs, *in* ‘Advances in Neural Information Processing Systems’, Curran Associates, Inc.

Choudhury, S., Srinivasa, S. and Scherer, S. (2018), Bayesian active edge evaluation on expensive graphs, *in* ‘IJCAI’.

Cohen, B., Phillips, M. and Likhachev, M. (2015), Planning single-arm manipulations with n-arm robots, *in* ‘Eigth Annual Symposium on Combinatorial Search’.

Dellin, C. M. and Srinivasa, S. (2016), A unifying formalism for shortest path problems with expensive edge evaluations via lazy best-first search over paths with edge selectors, *in* ‘ICAPS’.

Dor, A., Greenshtein, E. and Korach, E. (1998), ‘Optimal and myopic search in a binary random vector’, *Journal of applied probability* .

Eyerich, P., Keller, T. and Helmert, M. (2010), High-quality policies for the canadian traveler’s problem., *in* ‘AAAI’.

Ferguson, D. and Stentz, A. (2007), Field d*: An interpolation-based path planner and replanner, *in* ‘ISRR’.

Fried, D., Shimony, S. E., Benbassat, A. and Wenner, C. (2013), ‘Complexity of canadian traveler problem variants’, *Theoretical Computer Science* .

Gelly, S. and Silver, D. (2007), Combining online and offline knowledge in uct, *in* ‘ICML’.

Golovin, D., Krause, A. and Ray, D. (2010), Near-optimal bayesian active learning with noisy observations, *in* ‘NIPS’.

Guez, A., Silver, D. and Dayan, P. (2012), Efficient bayes-adaptive reinforcement learning using sample-based search, *in* ‘Advances in neural information processing systems’.

Guzzi, J., Chavez-Garcia, R., Gambardella, L. and Giusti, A. (2019), On the impact of uncertainty for path planning, *in* ‘ICRA’.

György, A., Linder, T., Lugosi, G. and Ottucsák, G. (2007), ‘The on-line shortest path problem under partial monitoring’, *Journal of Machine Learning Research* .

Haddadin, S., De Luca, A. and Albu-Schäffer, A. (2017), ‘Robot collisions: A survey on detection, isolation, and identification’, *IEEE Transactions on Robotics* **33**(6), 1292–1312.

Haghtalab, N., Mackenzie, S., Procaccia, A., Salzman, O. and Srinivasa, S. (2018), The provable virtue of laziness in motion planning, *in* ‘ICAPS’.

Hauser, K. (2015), Lazy collision checking in asymptotically-optimal motion planning, *in* ‘ICRA’, pp. 2951–2957.

Hermann, A., Drews, F., Bauer, J., Klemm, S., Roennau, A. and Dillmann, R. (2014), Unified GPU voxel collision detection for mobile manipulation planning, *in* ‘IROS’.

Hou, B., Choudhury, S., Lee, G., Mandalika, A. and Srinivasa, S. S. (2020), Posterior sampling for anytime motion planning on graphs with expensive-to-evaluate edges, *in* ‘ICRA’.

Janson, L., Hu, T. and Pavone, M. (2018), ‘Safe motion planning in unknown environments: Optimality benchmarks and tractable policies’, *arXiv preprint arXiv:1804.05804* .

Kavraki, L., Svestka, P., Latombe, J. and Overmars, M. (1996), ‘Probabilistic roadmaps for path planning in high-dimensional configuration spaces’, *Robotics and Automation, IEEE Transactions on* .

Kimmel, A., Sintov, A., Tan, J., Wen, B., Boularias, A. and Bekris, K. E. (2019), Belief-space planning using learned models with application to underactuated hands, *in* ‘ISRR’.

Klingensmith, M., Koval, M., Srinivasa, S., Pollard, N. and Kaess, M. (2016), ‘The manifold particle filter for state estimation on high-dimensional implicit manifolds’.

Kocsis, L. and Szepesvári, C. (2006), Bandit based monte-carlo planning, *in* 'European conference on machine learning'.

Koenig, S. and Likhachev, M. (2002), D* lite, *in* 'AAAI'.

Koonjul, G. S., Zeglin, G. J. and Pollard, N. S. (2011), Measuring contact points from displacements with a compliant, articulated robot hand, *in* '2011 IEEE International Conference on Robotics and Automation', pp. 489–495.

Kuppuswamy, N., Alspach, A., Uttamchandani, A., Creasey, S., Ikeda, T. and Tedrake, R. (2020), Soft-bubble grippers for robust and perceptive manipulation, *in* '2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)', pp. 9917–9924.

Laurent, H. and Rivest, R. (1976), 'Constructing optimal binary decision trees is np-complete', *Information processing letters* .

Lee, A., Duan, Y., Patil, S., Schulman, J., McCarthy, Z., van den Berg, J., Goldberg, K. and Abbeel, P. (2013), Sigma hulls for gaussian belief space planning for imprecise articulated robots amid obstacles, *in* 'IROS', pp. 5660–5667.

Lim, Z. W., Hsu, D. and Lee, W. S. (2015), Adaptive stochastic optimization: From sets to paths, *in* 'Advances in Neural Information Processing Systems'.

Lim, Z. W., Hsu, D. and Lee, W. S. (2016), 'Adaptive informative path planning in metric spaces', *IJRR* .

Lim, Z. W., Hsu, D. and Lee, W. S. (2017), Shortest path under uncertainty: Exploration versus exploitation, *in* 'UAI'.

Littman, M. L., Cassandra, A. R. and Kaelbling, L. P. (1995), Learning policies for partially observable environments: Scaling up, *in* 'Machine Learning Proceedings 1995'.

Mandalika, A., Choudhury, S., Salzman, O. and Srinivasa, S. (2019), Generalized lazy search for robot motion planning: Interleaving search and edge evaluation via event-based toggles, *in* 'ICAPS'.

Mandalika, A., Salzman, O. and Srinivasa, S. (2018), Lazy Receding Horizon A* for Efficient Path Planning in Graphs with Expensive-to-Evaluate Edges, *in* 'icaps'.

Mitash, C., Boularias, A. and Bekris, K. E. (2018), 'Improving 6d pose estimation of objects in clutter via physics-aware monte carlo tree search', *ICRA* pp. 1–8.

Narayanan, V. and Likhachev, M. (2017), Heuristic search on graphs with existence priors for expensive-to-evaluate edges, *in* 'ICAPS'.

Nikolova, E. and Karger, D. R. (2008), Route planning under uncertainty: The canadian traveller problem., *in* 'AAAI'.

Ong, S., Png, S., Hsu, D. and Lee, W. (2009), POMDPs for robotic tasks with mixed observability, *in* 'Proc. Robotics: Science and Systems'.

Osband, I., Russo, D. and Van Roy, B. (2013), (more) efficient reinforcement learning via posterior sampling, *in* 'NIPS'.

Pang, T., Umenberger, J. and Tedrake, R. (2021), Identifying external contacts from joint torque measurements on serial robotic arms and its limitations, *in* 'ICRA'.

Papadimitriou, C. H. and Yannakakis, M. (1991*a*), 'Shortest paths without a map', *Theoretical Computer Science* .

Papadimitriou, C. and Yannakakis, M. (1991*b*), 'Shortest paths without a map', *Theoretical Computer Science* .

Park, D., Kapusta, A., Hawke, J. and Kemp, C. C. (2014), Interleaving planning and control for efficient haptically-guided reaching in unknown environments, *in* 'Humanoids'.

Platt, R., Tedrake, R., Kaelbling, L. and Lozano-Perez, T. (2010), Belief space planning assuming maximum likelihood observations, *in* 'Proceedings of Robotics: Science and Systems', Zaragoza, Spain.

Páll, E., Sieverling, A. and Brock, O. (2018), Contingent contact-based motion planning, *in* 'IROS', pp. 6615–6621.

Richter, C., Vega-Brown, W. and Roy, N. (2015), Bayesian learning for safe high-speed navigation in unknown environments, *in* 'ISRR'.

Ross, S. (2014), *Introduction to stochastic dynamic programming*, Academic press.

Russo, D. and Roy, B. V. (2018), 'A tutorial on thompson sampling', *Foundations and Trends in Machine Learning* .

Saund, B. and Berenson, D. (2018), Motion planning for manipulators in unknown environments with contact sensing uncertainty, *in* 'ISER'.

Saund, B., Choudhury, S., Srinivasa, S. and Berenson, D. (2019), The blindfolded robot : A bayesian approach to planning with contact feedback, *in* 'ISRR'.

Siegel, M., Gunatilake, P. and Podnar, G. (1998), 'Robotic assistants for aircraft inspectors', *IEEE Instrumentation & Measurement* .

Silver, D., Huang, A., Maddison, C., Guez, A., Sifre, L., Driessche, G., Schrittwieser, J., Antonoglou, I., Panneershelvam, V., Lanctot, M., Dieleman, S., Grewe, D., Nham, J., Kalchbrenner, N., Sutskever, I., Lillicrap, T., Leach, M., Kavukcuoglu, K., Graepel, T. and Hassabis, D. (2016), 'Mastering the game of go with deep neural networks and tree search', *nature* .

Silver, D. and Veness, J. (2010), Monte-carlo planning in large POMDPs, *in* 'NIPS'.

Stentz, A. (1994), Optimal and efficient path planning for partially-known environments, *in* 'Proceedings of the 1994 IEEE International Conference on Robotics and Automation', pp. 3310–3317 vol.4.

Talebi, M. S., Zou, Z., Combes, R., Proutiere, A. and Johansson, M. (2017), 'Stochastic online shortest path routing: The value of feedback', *IEEE Transactions on Automatic Control* .

Thompson, W. R. (1933), 'On the likelihood that one unknown probability exceeds another in view of the evidence of two samples', *Biometrika* .

Thrun, S., Burgard, W. and Fox, D. (2005), *Probabilistic Robotics (Intelligent Robotics and Autonomous Agents)*, The MIT Press.

Tong, S. and Koller, D. (2001), 'Support vector machine active learning with applications to text classification', *Journal of machine learning research* .

Wettels, N., Santos, V., Johansson, R. and Loeb, G. (2008), 'Biomimetic tactile sensor array', *Advanced Robotics* **22**, 829–849.

Wirnshofer, F., Schmitt, P. S., Meister, P., v. Wichert, G. and Burgard, W. (2019), State estimation in contact-rich manipulation, *in* '2019 International Conference on Robotics and Automation (ICRA)', pp. 3790–3796.

Yoon, S. W., Fern, A., Givan, R. and Kambhampati, S. (2008), Probabilistic planning via determinization in hindsight., *in* 'AAAI'.

Yuan, W., Dong, S. and Adelson, E. H. (2017), 'Gelsight: High-resolution robot tactile sensors for estimating geometry and force', *Sensors* **17**(12).
URL: *https://www.mdpi.com/1424-8220/17/12/2762*

Appendix for "The Blindfolded Traveler's Problem: A Search Framework for Motion Planning with Contact Estimates"

A Analysis of BTP strategies

We map each strategy previously discussed onto the BTP definition, and when applicable provide analysis. Since we established that BTP is a hard problem (Section 4.2), 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 currently at a vertex v_t and must decide which edge e_t from the set of outgoing edges $\mathcal{N}(v_t)$ to traverse. Each strategy considers an agent at vertex v_t and must decide which edge e_t from the set of outgoing edges $\mathcal{N}(v_t)$ to traverse. Naturally, the edge chosen will depend on the current belief b_t, which is determined by the initial b_0, the history of observations ψ_t, and the update procedures of the previous section.

A.1 Heuristic Estimates of Q-values

One class of approaches approximates 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, and 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. Strategies within this class are Optimism in the Face of Uncertainty (OFU), Thompson Sampling (TS), QMDP, MCBE, and our proposed Collision Measure (CM).

A.1.1 Optimism in the Face of Uncertainty (OFU) A common approach for planning under uncertainty is to be optimistic (Brafman and Tennenholtz 2002), and pick a world from the plausible set of worlds that leads to the lowest cumulative cost to reaching a goal. 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 (Stentz 1994; Koenig and Likhachev 2002) as well as for solving CTP (Bnaya et al. 2009).

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:

$$\begin{aligned} \widehat{\mathcal{G}} &= (\mathcal{V}, \mathcal{E} \setminus \{e \mid P(x(e) = \mathrm{BLOCKED}|b_t) = 1\}, \mathcal{W}) \\ e_t &= \left\{ e \in \mathcal{N}(v_t) \;\middle|\; e \in \textsc{ShortestPath}(\widehat{\mathcal{G}}, v_t, v_g) \right\} \end{aligned} \tag{14}$$

Here $\widehat{\mathcal{G}}$ is the optimistic graph created by removing all edges that are known with certainty to be invalid under the current belief generated from history ψ. 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 attempts the outgoing edge belonging to the shortest path.

We can bound the sub-optimality of a variant of the optimistic policy which backtracks to the start 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 the 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 $x(e) = \mathrm{BLOCKED}$.
3. If it hits a blocked edge, it back tracks to start v_s and repeats.

Then the following theorem is true

Theorem 3. *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*

$$c(\pi^{\mathrm{OFU2}}(x,\eta)) \leq 2Kw^* \tag{15}$$

Proof 3. *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.*

Consider access to an oracle that could query the validity of any edge. BTP is then equivalent to the shortest path planning problem on expensive graphs (Dellin and Srinivasa 2016). π^{OFU2} tests unknown edges in the equivalent order to LazySP (Dellin and Srinivasa 2016) (with a forward edge selector) which has been shown to be optimal (Mandalika et al. 2018). Compared to the shortest path planning problem, BTP is challenging because without an oracle the cost of querying an edge is dependent on the agent's current location, which itself depends on the previous edges queried. A natural question that we do not address in this work is then, how much cost would an agent be willing to incur for access to an oracle?

A.1.2 Thompson Sampling (TS) This is a commonly used heuristic for the Bayesian Multi-armed Bandit (MAB) problem based on the idea of randomized probability matching (Thompson 1933). At every decision step, TS samples a world from the current belief and selects the optimal action given that world. Hence action selection probability is matched to the posterior of actions being optimal. In recent literature, Thompson Sampling has shown to be empirically successful (Chapelle and Li 2011), theoretically sound (Agrawal and Goyal 2013) and applicable beyond MAB to RL (Osband et al. 2013).

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

$$\begin{aligned} \hat{x} &\sim P(x|b_t), \\ \widehat{\mathcal{G}} &= (\mathcal{V}, \mathcal{E} \setminus \{e \mid \hat{x}(e) = \text{BLOCKED}\}, \mathcal{W}) \\ e_t &= \left\{ e \in \mathcal{N}(v_t) \;\middle|\; e \in \textsc{ShortestPath}(\widehat{\mathcal{G}}, v_t, v_g)) \right\} \end{aligned} \tag{16}$$

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 can provide a bound for MAB w.r.t Bayesian regret, i.e., the expected regret under the prior (Russo and Roy 2018). These bounds are meaningful for repeated trials on the same world, which is not the case for BTP.

However, if we consider a repeated instance of BTP such regret bounds apply. Consider the repeated variation of a BTP where

1. At each iteration i the robot is tasked with solving a BTP, moving to the goal and then back to the start, receiving reward $-\sum_{e_j \in \xi} \mathcal{W}(e_j)$ where ξ is the path traversed. If the goal is not reached the robot receives some large negative reward.
2. Observations ψ are shared and the obstacles remain fixed between iterations.

This Repeated BTP is analogous to the Experienced Lazy Path Search problem, for which Thompson sampling (within an algorithm called PSMP) has bounded regret compared to the optimal policy always taking the shortest path (Hou et al. 2020). Consider the strategy that attempts the path from Thompson sampling, and if a collision occurs backtracks to the start and executes the shortest path found so far. This strategy accumulates at most 2 times the cost of the PSMP strategy (which is able to query edges arbitrarily), and thus has the same bounded regret.

A.1.3 QMDP This is one of the most commonly used heuristics for POMDPs (Littman et al. 1995). 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 $\pi^{\text{QMDP}} = \arg\min_a \hat{Q}(b,a)$. Mapping this back to BTP, the agent chooses edge e_t as follows:

$$\begin{aligned} e_t = &\arg\min_{e \in \mathcal{N}(v_t)} \\ &\mathbb{E}_{(x,\eta)\sim P(\cdot|b_t)}\left[c + w(\textsc{ShortestPath}(\mathcal{G}(x), v', v_g))\right] \end{aligned} \tag{17}$$

$$\text{where } (v', c) = \Gamma(v_t, e, x, \eta) \tag{18}$$

$$\text{and } \mathcal{G}(x) = (\mathcal{V}, \mathcal{E} \setminus \{e \mid x(e) = 0\}, \mathcal{W}) \tag{19}$$

Here we sample a set of worlds $(x,\eta) \sim P(\cdot|b_t)$. For each candidate edge $e \in \mathcal{N}(v_t)$, we simulate moving along the edge (which may or may not result in a collision) 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.

A.1.4 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

$$\pi^{\text{MCBE}} = \arg\max_a \mathbb{E}_{s\sim b}\left[\mathbb{I}(a \in \arg\min_{a'} Q^*(s,a'))\right]$$

Mapping this back to BTP, the agent chooses edge e_t as follows:

$$\mathcal{G}(x) = (\mathcal{V}, \mathcal{E} \setminus \{e \mid x(e) = 0\}, \mathcal{W}) \quad (20)$$

$$e_t = \underset{e \in \mathcal{N}(v_t)}{\arg\max} \; \mathbb{E}_{(x,\eta)\sim P(\cdot|b_t)} \left[\mathbb{I}(e \in \textsc{ShortestPath}(\mathcal{G}(x), v_t, v_g))\right] \quad (21)$$

Here we sample a set of worlds $(x, \eta) \sim P(\cdot|b_t)$, 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.

A.1.5 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 Choudhury et al. (2016) 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:

$$\widehat{\mathcal{G}} = (\mathcal{V}, \mathcal{E}, w(e) - \alpha \log P(x(e) = 1|b_t))$$
$$e_t = \left\{ e \in \mathcal{N}(v_t) \;\middle|\; e \in \textsc{ShortestPath}(\widehat{\mathcal{G}}, v_t, v_g) \right\} \quad (22)$$

Here $\widehat{\mathcal{G}}$ is an optimistic graph created by removing all edges that are invalid with probability 1 under the current belief b_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 $(P(x(e) = 1|b) = 0)$ yields a weight of ∞, and a known free edge $(P(x(e) = 1|b) = 1)$ yields $w(e)$.

We provide theoertical justification behind such a heuristic. We begin by mapping BTP to a Bayesian Search (Ross 2014) problem. Let $\Xi = (\xi_1, \xi_2, \dots, \xi_n)$ be the set of simple paths from v_s to v_g. The probability of edge validity $P(x)$ maps to a joint probability $P((\xi_1, \xi_2, \dots, \xi_n))$ 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 $\mathbb{E}[c(\sigma)]$ be the expected cost of a sequence. The optimal sequence σ^* has minimal expected cost, i.e. $\mathbb{E}[c(\sigma^*)] \leq \mathbb{E}[c(\sigma)]$ 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.

$$\sigma^g(i+1) = \underset{j}{\arg\max} \frac{P(\xi_j = 1|\xi_{\sigma^g(1)} = 0, \xi_{\sigma^g(2)} = 0, \cdots, \xi_{\sigma^g(i)} = 0)}{c(\xi_j)} \quad (23)$$

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. (1998)(Theorem 4.1) proved that greedy has an optimality bound of 4

Theorem 4. *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

$$\mathbb{E}[c(\sigma^g)] \leq 4\mathbb{E}[c(\sigma^*)] \quad (24)$$

Proof 4. *We refer the reader to Theorem 4.1 in Dor et al. (1998).*

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 4

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

$$\mathbb{E}[\bar{c}(\sigma^g)] \leq 8\mathbb{E}[\bar{c}(\bar{\sigma}^*)] \quad (25)$$

Proof 5. *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*

$$\bar{c}(\sigma^g) \leq c(\sigma^g) \leq 4c(\sigma^*) \leq 4c(\bar{\sigma}^*) \leq 8\bar{c}(\bar{\sigma}^*) \quad (26)$$

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

$$\pi^{\text{CM2}} \equiv \left\{ e \in \mathcal{N}(v_t) \;\middle|\; e \in \arg\min_{\xi} \frac{w(\xi)}{P(x(\xi) = 1 | b_t)} \right\} \tag{27}$$

The optimization in (27) 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 $\alpha \geq \frac{\frac{1}{p_{\min}} - 1}{\log \frac{1}{p_{\min}}}$

$$(1 - \log p) \leq \frac{1}{p} \leq (1 - \alpha \log p) \tag{28}$$

Hence $(1 - \alpha \log p)$ is a good family of approximators to $\frac{1}{p}$ which justifies (22) is an approximation.

A.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 A.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 A.1 can be used as a rollout policy. For any such policy, we have the following upper bound

$$\hat{Q}(b,a) \geq \mathbb{E}_{s \sim b}\left[c(s,a) + V^{\pi^*}(s')\right] \geq Q^*(b,a) \tag{29}$$

If this is close to matching lower bounds from Section A.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.

While both the QMDP and MCBE strategies from Section A.1 involve one step of simulation, we limit this section to policies that require longer rollouts. We consider the simulation-based policies of Optimistic Rollout (ORO) and Upper Confidence Tree (UCT).

A.2.1 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 $V^{\pi^{\text{OFU}}(v,b)}(x,\eta)$ be the evaluation of the policy starting from vertex v with belief b on an underlying graph (x,η). The agent chooses edge e_t as follows:

$$\begin{aligned} e_t &= \arg\min_{e \in \mathcal{N}(v_t)} \mathbb{E}_{(x,\eta) \sim P(\cdot|b_t)} \left[c + V^{\pi^{\text{OFU}}(v',b')}(x,\eta)\right] \\ &\text{where } (v',c) = \Gamma(v_t, e, x, \eta) \text{ and } b' = b_t \cup (x(e), \eta(e)) \end{aligned} \tag{30}$$

A.2.2 Upper Confidence Tree (UCT)

This is a state of the art algorithm from planning under uncertainty (Kocsis and Szepesvári 2006) which combines the framework Monte-Carlo Tree Search with Upper Confidence Bound (UCB) for action selection. It has successfully been used for solving games (Gelly and Silver 2007; Silver et al. 2016), POMDPs (Silver and Veness 2010) and Bayesian RL (Guez et al. 2012). 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:

$$\arg\max_{a_i} B \sqrt{\frac{\log N(b,a_i)}{N(b,a_i)}} - \hat{Q}(b,a) \tag{31}$$

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 Eyerich et al. (2010) 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.

A.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) (Lim et al. 2017) 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.