Dynamic Replanning with Posterior Sampling

Brian Hou and Siddhartha S. Srinivasa

***Abstract*— When navigating to a goal in an uncertain environment, a robot must simultaneously navigate the exploration-exploitation tradeoff: should it aim to gain information and reduce uncertainty, or should it simply brave the unknown? We formalize this as the Bayesian dynamic motion planning problem, and we analyze how several strategies from the literature balance these concerns via determinization and planning. Within the framework of determinization in the face of uncertainty, we shift the burden of exploration to determinization rather than planning. Dynamic Replanning with Posterior Sampling (DRPS) is very efficient: each iteration consists of a single posterior update and a shortest path query. Relative to comparative baselines across seven datasets of 2D planning problems, DRPS has a higher percentage of success, traverses lower or comparable total distances, and accelerates total planning time by 4-7×. Across a dataset of larger 7D Baxter manipulator planning problems, DRPS reduces total distance by 40% and total planning time by 18×.**

I. INTRODUCTION

We focus on the problem of motion planning under uncertainty, where a robot must navigate to a goal without exact knowledge of obstacles in the environment. Because environments contain structure (e.g., obstacles are not randomly placed, configuration-space edge collisions are correlated via the workspace), a robot can leverage its observations to infer regions that are blocked by obstacles. We formalize this task as a Bayesian motion planning problem, where the robot's understanding of its uncertain environment is captured by a posterior distribution conditioned on its past observations. Thus, its behavior must adapt to its current position and posterior distribution over environments.

This general Bayesian dynamic motion planning problem can be described as a partially observable Markov decision process (POMDP). Solving for the optimal policy tree is computationally intractable: beginning from the start state v_s and prior distribution over environments $P(\phi)$, an offline policy tree of depth D for Bayesian motion planning contains $\mathcal{O}(2^D|V|^D)$ nodes due to the "curse of history." Rather than preparing a response for all possible action-observation sequences, *online* POMDP methods interleave planning and execution only for the current information state [1]. This approach is critical for scaling to large POMDPs [2–4].

POMDP algorithms are designed to address uncertainty across a variety of sources, including the robot's state and the transition and reward functions. However, factored POMDP models provide structure to uncertainty that can be leveraged to plan more efficiently [5–7]. In Bayesian dynamic motion planning, uncertainty originates only from the robot's ignorance about the environment; the transition and reward function are deterministic given the environment, and the robot's state is fully observable.

We unify several online algorithms for dynamic motion planning under uncertainty within the framework of *determinization in the face of uncertainty* (Fig. 1). Determinization is a successful and well-studied heuristic that takes advantage of structure in probabilistic planning problems [3, 8, 9]. Algorithms alternate between constructing a deterministic estimate of the uncertain environment (*determinization*) and computing a path through the approximation (*planning*). Executing this initial solution acquires more information, thereby improving the next deterministic approximation.

We analyze popular D*-based optimistic approaches [10, 11] and uncertainty-aware Bayesian methods [12, 13] through this lens. Our key insight is:

> The determinization strategy is critical for efficiently balancing exploration with exploitation.

If the deterministic approximation preserves too many possible paths, planning must expend additional effort to avoid over-exploration. Conversely, if determinization prunes too many exploration options, planning cannot retroactively correct for this imbalance. We reinterpret *posterior sampling* (also known as Thompson sampling [14]) as a determinization strategy that effectively navigates this tradeoff. Posterior sampling achieves excellent empirical and theoretical performance in multi-armed bandits, reinforcement learning, model predictive control, and motion planning [15–17].

Dynamic Replanning with Posterior Sampling samples an environment from the posterior distribution and follows the optimal solution in that environment. Random sampling from the posterior judiciously explores the space of plausible environments, and each solution thus explores the space of plausible *paths*. We make the following contributions:

- We define the Bayesian dynamic motion planning problem, and characterize how existing algorithms navigate the exploration-exploitation tradeoff.
- We introduce Dynamic Replanning with Posterior Sampling (DRPS), an algorithm that shifts the burden of exploration to determinization rather than planning.
- We demonstrate that DRPS outperforms relevant baselines for a wide variety of 2D and 7D motion planning problems, with a lower or comparable total distance traveled and significant gains in computation time. This is especially prominent on larger 7D planning problems.

This work was (partially) funded by the National Science Foundation IIS (#2007011), National Science Foundation DMS (#1839371), the Office of Naval Research, US Army Research Laboratory CCDC, Amazon, and Honda Research Institute USA. Brian Hou was partially supported by a NASA Space Technology Research Fellowship.

Both authors are with the Paul G. Allen School of Computer Science & Engineering, University of Washington {bhou, siddh}@cs.uw.edu

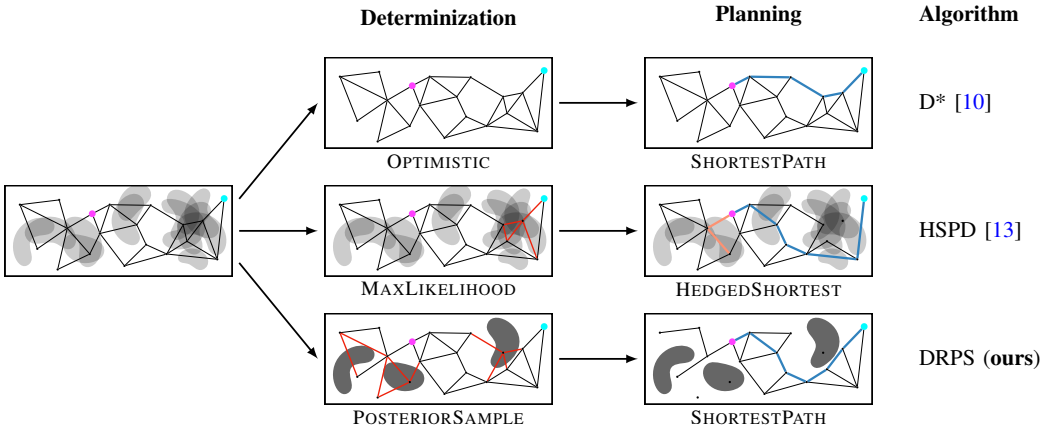


Fig. 1: Determinization in the face of uncertainty is a popular framework for solving challenging Bayesian dynamic motion planning problems. The probabilistic problem (left) reflects uncertainty about obstacles (gray). In this framework, algorithms construct a deterministic estimate of the uncertain environment by removing edges from the graph (red) and assuming the remaining edges are collision-free. Because uncertainty was eliminated by *determinization*, *planning* becomes more tractable. To navigate the exploration-exploitation tradeoff, HSPD explicitly plans an exploration path (orange) and exploitation path (blue) and follows the shorter of the two. DRPS shifts the burden of exploration to determinization, which simplifies each iteration of planning to a single shortest path query and yields significant performance gains.

II. RELATED WORK

A. Bayesian Motion Planning and Reinforcement Learning

In the general Bayesian reinforcement learning (BRL) problem, the reward and transition functions of a Markov decision process are uncertain [5]. BRL algorithms typically target the objective of Bayesian regret, which aims to bound the cumulative difference in expected reward between the optimal and learned policies. UCRL2 determinizes the BRL problem with optimism in the face of uncertainty by following the optimal policy for an optimistic MDP given its observations thus far [18]. PSRL determinizes the BRL problem with posterior sampling by following the optimal policy for a sampled plausible MDP [15, 16, 19]. In practice, PSRL seems to outperform optimistic algorithms; Osband and Van Roy hypothesize that current algorithms perform unnecessary exploration as a consequence of excessive optimism [20].

A growing thread of work views lazy motion planning through a Bayesian lens. Because collision checking is computationally expensive, Bayesian algorithms can leverage the posterior to infer many edge collision statuses. The Bayesian active learning problem of decision region determination is equivalent to feasible path planning, and a combination of offline DIRECT [21] with online BISECT [22] achieves state-of-the-art performance [23]. PSMP views anytime motion planning as an instance of Bayesian reinforcement learning and draws an equivalence between anytime search performance and the objective of Bayesian regret [17]. Because uncertainty stems from computational limitations rather than sensor limitations in these problems, algorithms can reduce uncertainty anywhere in the configuration space with the same cost. This is not true for the dynamic setting we consider in this paper, where the robot must physically move to reduce uncertainty.

B. Dynamic Replanning Under Uncertainty

Variants of D* and D* Lite have been widely deployed in uncertain environments [10, 11, 24]. As edge costs change (e.g., when new obstacles are perceived), the search tree is repaired to dramatically reduce replanning time while remaining functionally equivalent to replanning from scratch with A*. D*-based algorithms determinize optimistically in the face of uncertainty; space is assumed to be collision-free until the robot perceives obstacles. In uncluttered scenarios where this assumption is generally warranted, exploiting the optimistic path is an excellent heuristic that sidesteps most of the computational expense of modeling uncertainty (e.g., updating the Bayesian posterior). The surprising effectiveness of this simple approach highlights a key feature of the Bayesian dynamic motion planning problem: acquiring information and reducing uncertainty is relatively easy.

The Canadian Traveler's Problem describes a similar scenario, where edge blockages are discovered only upon arriving at an incident node. Planning a path that achieves a fixed suboptimality ratio to the shortest path is intractable [25]. Similarly, in dynamic motion planning problems, uncertainty is a consequence of sensor limitations. Edge blockages can be perceived only when the robot is physically nearby. Fortunately, in many real-world examples, there is correlation between edge blockages. Though just the nearby blockages can be perceived, those measurements enable the planner to infer other plausible blockages. Stochastic variants of the Canadian Traveler's Problem have been proposed, where edge costs are estimated independently [26, 27], modeled by a Gaussian Process [28], or modeled using a black-box Bayesian posterior [13]. The Blindfolded Traveler's Problem is a specific Bayesian dynamic motion planning problem, where the only source of information is contact feedback and correlations are described with an approximate posterior [29]. While these Bayesian problems remain theoretically intractable to solve optimally, approximate algorithms, such as Hedged Shortest Path under Determinization (HSPD), can achieve high-quality solutions that navigate the exploration-exploitation tradeoff [13]. DRPS is designed to solve this class of Bayesian motion planning problems.

As with POMDPs, Bayesian dynamic motion planning problems can be solved either offline or online. In the Reac-

tive Planning Problem, a complete policy must be computed offline under uncertainty about which edges are blocked. Mutual information policies that trade off exploration and exploitation produce effective approximate solutions to this problem [30]. The Learned Reactive Planning Problem extends this to a lifelong setting by allowing the policy to adapt to previously observed obstacles across multiple episodes of navigation [31]. Algorithms for both versions of the Reactive Planning Problem rely on a fixed set of possible edge subsets, which can be filtered as edge statuses are observed. HSPD and DRPS permit more flexible distributions over environments and edges.

III. BAYESIAN DYNAMIC MOTION PLANNING

Given start x_s and goal x_g in the configuration space $\mathcal{X}$, Bayesian dynamic motion planning seeks to minimize the expected total distance traveled under the distribution of environments $P(\phi)$. The robot is endowed with a model of the environment uncertainty, formalized as a posterior distribution over environments $P(\phi|\psi_t)$, where ψ_t contains the history of observations through the current time.

This work focuses on the Bayesian dynamic motion planning (BDMP) problem for roadmaps. A roadmap is a graph G with vertices V and possible edges E. Some edges will not be traversable due to collisions with the unknown environment ϕ, which must be explored. Each edge has known weight $w : E \to \mathbb{R}^+$ and unknown collision status $\phi : E \to \{0, 1\}$, where $\phi(e) = 1$ means e is collision-free in environment ϕ. A path ξ is a sequence of edges, with a total distance traveled of $w(\xi) = \sum_{e \in \xi} w(e)$. In general, it may contain repeated edges if the planner retraces its steps.

The online algorithms we analyze in the following section determinize the unknown environment based on the current posterior $P(\phi|\psi_t)$ and propose a path $\hat{\xi}_{t+1}$ based on that determinization. However, following the proposed path may result in a collision; ξ_{t+1} refers to the prefix of $\hat{\xi}_{t+1}$ that was actually attempted, which may truncate edges after a collision. Similar to the Blindfolded Traveler's Problem [29], the robot returns to the source of the edge after discovering a collision. The concatenated path across the entire dynamic motion planning episode is $\xi_{1:T} = (\xi_1, \cdots, \xi_T)$, with a total distance traveled of $w(\xi_{1:T}) = \sum_t w(\xi_t)$.

Due to environment uncertainty in this dynamic setting, the initial plan is unlikely to be collision-free. Thus, replanning is required to navigate to the goal. However, time is of the essence; because replanning is interleaved with execution, a slow algorithm will cause the robot to stop. An online BDMP algorithm should efficiently navigate the exploration-exploitation tradeoff to propose new paths that reduce uncertainty while minimizing the expected total distance traveled.

BDMP is closely related to the Blindfolded Traveler's Problem (BTP) [29] and the Bayesian Canadian Traveler's Problem (BCTP) [13]. However, the BTP focuses on approximate posterior distributions derived from contact feedback, while the BCTP and BDMP assume that the posterior distribution is given. The BCTP automatically reveals all edge collision statuses upon reaching an incident vertex, while the BTP and BDMP require the planner to explicitly choose an edge to sense for collisions. (D*-based algorithms support a more general form of edge blockages, which may be discovered anywhere in the graph. In practice, however, edge blockages are often discovered within a radius of the robot's position.) Together, these slight differences will help characterize how the proposed algorithms balance exploration and exploitation without confounding factors, i.e., with posterior approximation error (for BTP) or with varying amounts of information gained based on graph connectivity (for BCTP).

Algorithm 1 Determinization for Online BDMP

Require: Graph G, Posterior $P(\phi|\psi_t)$
1: **while** goal not reached **do**
2: Determinize BDMP based on $P(\phi|\psi_t)$
3: Plan path $\hat{\xi}_{t+1}$ in determinized $\widehat{G}$
4: Follow path until collision or end is reached
5: Update ψ_{t+1} with observations

IV. DETERMINIZATION FOR ONLINE BDMP

Algorithm 1 summarizes the framework of determinization in the face of uncertainty, which unifies several algorithms for online BDMP (Fig. 1). Algorithm 2 characterizes how these algorithms choose to determinize. Given a posterior distribution that describes each edge's probability of collision, the determinization strategy approximates each edge as either blocked or collision-free. In the resulting roadmap, the statuses of each edge are known; Algorithm 3 describes algorithms for planning through the determinized roadmap.

A. Balancing Exploration and Exploitation

1) D:* D* plans the shortest path from the current vertex to the goal on an optimistic determinization of the roadmap. The surprising effectiveness and popularity of this strategy demonstrates that even "pure" exploitation in BDMP results in additional information. This strategy is complete: if there exists a feasible path to the goal, D* will eventually discover it. However, D* may optimistically try many paths before it succeeds. We compare against this baseline due to its popularity and to clarify the difference between optimistic and Bayes-aware algorithms for dynamic motion planning.

2) HSPD: Hedged Shortest Path under Determinization plans on the maximum-likelihood-observation determinization [12]: only edges that are more likely to be collision-free are preserved (i.e., $P(\phi(e) = 1|\psi_t) \geq 0.5$). The shortest path from the current vertex to the goal on this determinization is deemed the "exploitation" path ξ^*. The exploitation path may be longer than the true shortest path or may not exist at all, depending on which edges the determinized roadmap is able to preserve. However, following this path and discovering a collision proves that the determinization was inconsistent with the true unknown environment, reducing the space of possible environments by a factor of 2.

HSPD also computes a second "exploration" path ξ_ψ, via ORIENTEER. Assuming that following the path would yield the maximum-likelihood collision-free observations, ξ_ψ is

the shortest path that gathers enough information to reduce the space of possible environments by a factor of 2. Although each edge preserved by this determinization is more likely than not to be collision-free, traversing that path still accumulates a reduction in the space of possible environments that corresponds to how likely the path was to be in collision. Computing the exact reduction requires a posterior update along every edge of the exploration path, with additional posterior updates to consider alternative exploration paths.

HSPD hedges between ξ^* and ξ_ψ by following the shorter of the two paths. Following ξ^* to completion means that the goal has been reached. Following either ξ^* or ξ_ψ into a collision means that the maximum-likelihood-observation determinization was inconsistent, reducing the space of possible environments by a factor of 2. Following ξ_ψ to completion also reduces the space of possible environments by a factor of 2 by construction. Since each iteration shrinks the possible environment space by at least half, HSPD reaches the goal within a logarithmic number of iterations. Qualitatively, HSPD follows the exploration path until it becomes too costly to achieve the desired $2\times$ uncertainty reduction (i.e., the exploitation path to the goal becomes shorter). On a small grid-based Bayesian Canadian Traveler's Problem, HSPD finds that an additional hyperparameter that artificially stretches the length of ξ_ψ is critical to reduce hedging-induced over-exploration.

3) DRPS: Dynamic Replanning with Posterior Sampling determinizes according to a random sample from the current posterior distribution. Posterior sampling explores the space of currently plausible environments; as the posterior concentrates around the true environment, environments sampled from the posterior concentrate in the same way. Then, DRPS plans the shortest path from the current vertex to the goal in the sampled environment.

Ideally, a BDMP algorithm would explore the space of optimal paths rather than the space of environments—the objective is to minimize the total distance traversed, not to reduce uncertainty. For example, an algorithm focused on reducing environment uncertainty may continue to explore even after all plausible environments share the same optimal path. Though directly exploring the combinatorially large space of optimal paths is challenging, DRPS accomplishes this by sampling an environment and planning the optimal path in that environment. This procedure means that DRPS explores the space of plausible optimal paths $\hat{\xi}_{t+1} \sim P(\xi^*|\psi_t)$, a strategy that balances exploration in path space using the same posterior distribution that reflects its current uncertainty. Thus, DRPS naturally avoids over-exploration without additional hyperparameters.

B. Posterior Queries and Updates

HSPD and DRPS assume different interfaces to the posterior distribution. HSPD requires the marginal posterior distribution $P(\phi(e) = 1|\psi_t) \propto \int_\phi P(\phi(e) = 1|\phi)P(\phi|\psi_t)$, which must be normalized to perform the MAXLIKELIHOOD determinization. However, computing the partition function to normalize this distribution is intractable for many posteriors of interest. In contrast, DRPS requires only that the posterior distribution is sampleable: $\phi \sim P(\phi|\psi_t)$. Posterior sampling ensures that only statistically plausible environments are sampled, while algorithms that consider marginal collision probabilities effectively take a weighted average across all plausible environments.

Algorithm 2 Determinization Strategies

procedure OPTIMISTIC($G, P(\phi|\psi_t)$)
  $\widehat{E} = \{e \in E \mid P(\phi(e) = 1|\psi_t) > 0\}$
  return $V, \widehat{E}$
procedure MAXLIKELIHOOD($G, P(\phi|\psi_t)$)
  $\widehat{E} = \{e \in E \mid P(\phi(e) = 1|\psi_t) \geq 0.5\}$
  return $V, \widehat{E}$
procedure POSTERIORSAMPLE($G, P(\phi|\psi_t)$)
  $\widehat{\phi} \sim P(\phi|\psi_t)$
  $\widehat{E} = \widehat{\phi}(E)$
  return $V, \widehat{E}$

Algorithm 3 Planning Strategies

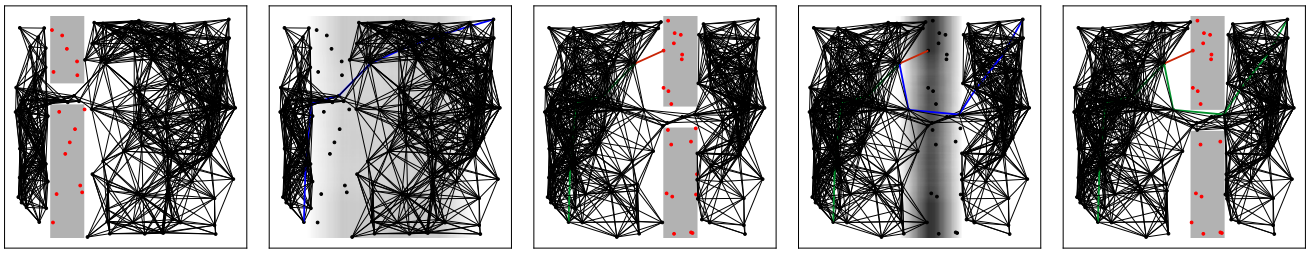
procedure SHORTESTPATH($v, v_g, \widehat{G}$)
  return A*$(v, v_g, \widehat{G})$
procedure HEDGEDSHORTEST($v, v_g, \widehat{G}, P(\phi|\psi_t)$)
  $\xi^* \leftarrow$ SHORTESTPATH($v, v_g, \widehat{G}$)
  $\xi_\psi \leftarrow$ ORIENTEER($v, \widehat{G}, P(\phi|\psi_t)$)
  return shorter of ξ^*, ξ_ψ

Both HSPD and DRPS require the posterior to be updated at the end of each iteration (Algorithm 1, Line 5). This is only partially true for D*, which does not require a full posterior update as long as newly-discovered edge blockages are reflected in the determinization. Additionally, HSPD performs several posterior updates as part of the ORIENTEER step to estimate the amount of information gained by following each hypothesized exploration subpath. This is more efficient than solving the original BDMP because each posterior update assumes the determinized maximum-likelihood observations [12], but each update can still be computationally expensive for complex posterior distributions. By shifting the burden of exploration to the posterior sampling determinization strategy, DRPS can plan on the determinized roadmap using a naïve shortest path algorithm like A*. DRPS avoids further posterior updates because it does not need to consider different exploration paths in the planning step.

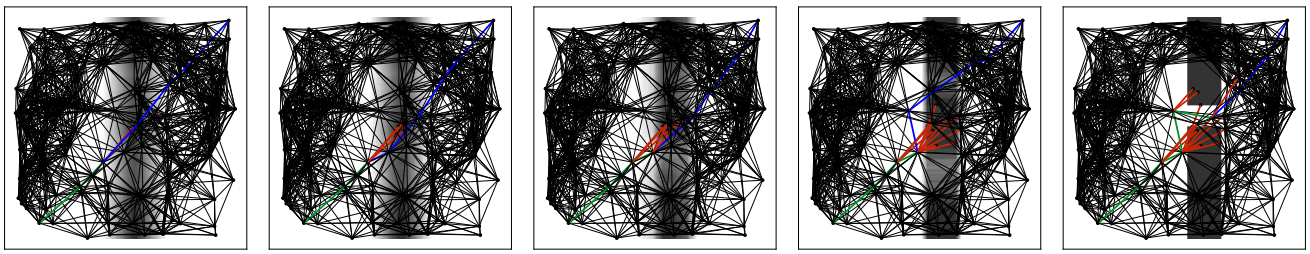
V. EXPERIMENTAL RESULTS

We implement the three baseline algorithms (D*, HSPD, and DRPS), Bayesian dynamic motion planning environments, and posterior distribution with Python and NumPy. HSPD and DRPS share the same posterior implementation, although querying the marginal posterior distribution versus sampling from the posterior distribution is necessarily slightly different. The code has been reasonably optimized for performance while preserving the modularity necessary

Fig. 2: Snapshots of DRPS and D* planning progress through the same environment. Regions with higher probability of collision are colored with darker shades of gray. Evaluated edges are found to either be in collision (red) or collision-free (green).



(a) DRPS samples an environment from the posterior and plans the shortest path to the goal (blue). The path traverses through a region of uncertainty and eventually results in a collision (red). It determinizes again and plans a new path, which makes progress but results in another collision. However, this exploration has concentrated the posterior around the true environment; the next iteration of posterior sampling and planning reaches the goal.



(b) D* optimistically determinizes the posterior and plans the shortest path to the goal (blue). Due to this optimism, D* continues to plan paths through the middle region, resulting in frequent collisions. However, it eventually reaches the goal.

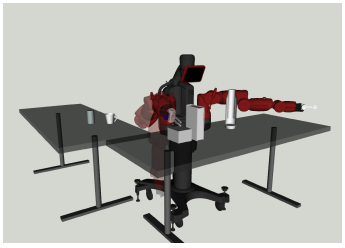


Fig. 3: Example problem from the 7-DOF Baxter manipulator dataset, where the right arm must move from below the table to above.

to evaluate different algorithms. We evaluate the algorithms on a 3.6-GHz Intel Core i7 processor with 64 GB of memory.

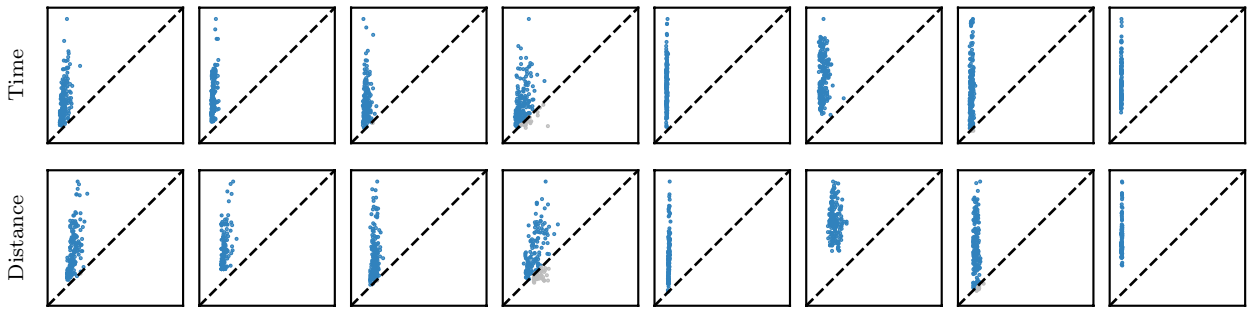
We evaluate planning performance on a wide variety of 2-DOF point robot environments [22] and cluttered 7-DOF Baxter manipulator environments (Fig. 3) [17]. The seven smaller point robot experiments let us comprehensively evaluate how these algorithms differ and qualitatively inspect their behavior (Fig. 2, Fig. 5). The Baxter experiments are important for demonstrating the efficacy of our approach in higher-dimensional planning problems with larger roadmaps. We evaluate the algorithms on 200 problems per dataset, for a total of 1600 problems. The roadmaps for the point robot problems range in size from 100-200 vertices and 2000-5000 edges, while the roadmap for the Baxter problems contains 5000 vertices and 140000 edges. Following HSPD, we randomly generate BDMP problems based on a dataset of possible template environments. HSPD has not been previously evaluated on these challenging environments (both in 2D and 7D); for more challenging problems, we find that HSPD can fail as a consequence of its maximum-likelihood-observation determinization strategy.

Fig. 4(a) and Fig. 4(b) visualize the performance of DRPS relative to D* and HSPD, respectively, on the key metrics of total planning time expended and total distance traveled. We summarize this data quantitatively in Table I, which includes additional information about the number of iterations to solve each problem. DRPS travels less distance than D* and typically requires less planning time to reach the goal. This demonstrates the value of taking a Bayesian, rather than an optimistic, approach to dynamic motion planning. We see especially significant gains in the Maze 2D and the Baxter environments, where the D* optimistic assumption is frequently violated, requiring many iterations of replanning. Fig. 2 visualizes snapshots as DRPS and D* solve the same Bayesian dynamic motion planning problem. D* expends significant travel distance trying to pass through regions that are already unlikely to be collision-free.

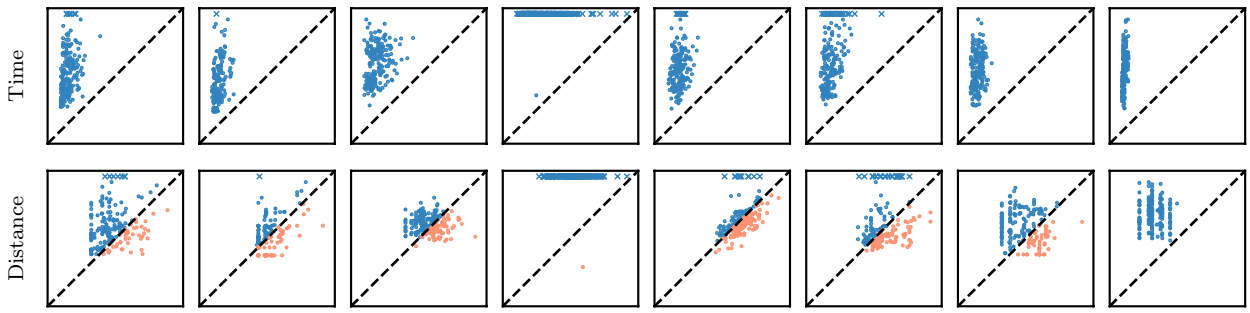
On most 2D problems, we find that DRPS travels a comparable distance to HSPD while spending less computation time. While HSPD occasionally failed to solve some problems in other datasets, it especially struggled to solve problems in the MovingWall dataset. We visualize snapshots of its progress in Fig. 5(b), which shows that the maximum-likelihood-observation determinization is too strict: edges that are crucial for connecting to the goal are eliminated because they are likely to be in collision. However, both exploration and exploitation paths are planned on this poor approximation to the BDMP; when edges are eliminated by determinization, HSPD cannot plan either path. Furthermore, HSPD's existing exploration parameter cannot help it recover from this scenario. This suggests that maximum-likelihood-observation determinization may be unsuitable for motion planning settings that are likely to be in collision.

We observed the largest improvement from HSPD to DRPS on the 7D Baxter environment, both in planning time and distance traveled. We also measured the number

Fig. 4: Pairwise performance comparisons, where each column represents a planning dataset (rightmost is the 7D Baxter dataset). Each point represents a BDMP problem, and its (x, y)-coordinates correspond to the planning time or distance traveled by (DRPS, baseline), with 0 in the bottom left corner. Point are colored based on the better-performing algorithm, either DRPS (blue) or the baseline. The farther a point is above the line $y = x$ (dashed), the more DRPS outperforms the baseline according to that metric on the corresponding BDMP problem.



(a) Pairwise comparisons with D* (gray). DRPS dominates D* in both planning time and distance traveled on most planning problems. DRPS performance is clustered narrowly along the x-axis, demonstrating relatively consistent planning time and distance traveled across problems within each dataset.



(b) Pairwise comparisons with HSPD (orange). Planning failures are labeled with an x; HSPD fails on nearly all planning problems in the MovingWall dataset (column 4). See Fig. 5 (bottom) for an illustrative example. DRPS consistently outperforms HSPD in terms of planning time. The even spread of points about $y = x$ for the 2D dataset distance plots shows that the two algorithms achieve comparable performance in this simpler setting. However, the distance plot for the 7D Baxter dataset (bottom right) shows that DRPS consistently travels shorter distances in this more difficult setting. (Fig. 4(b) visualizes the same DRPS data as Fig. 4(a), but zoomed in to compare DRPS and HSPD more effectively.)

of iterations required by each algorithm, to assess whether the root cause of the difference in planning time was due to the individual expense of each iteration or the accumulated cost of a larger number of iterations. We find that both are true: an iteration of HSPD takes 5× longer than an iteration of DRPS, and HSPD requires about 4× the number of iterations to reach the goal configuration. The relative difference in time spent on each iteration is likely due to the HEDGEDSHORTEST planning algorithm, which requires many posterior updates to plan an exploration path. The relative difference in iterations is likely because HSPD explores more than necessary. Fig. 5(a) shows that HSPD may incur additional travel distance as it continues to explore and reduce uncertainty. We conclude that BDMP algorithms must be accurately tuned for exploration. Otherwise, explicitly navigating the exploration-exploitation tradeoff during planning will incur additional computational expense without a corresponding improvement in distance traveled.

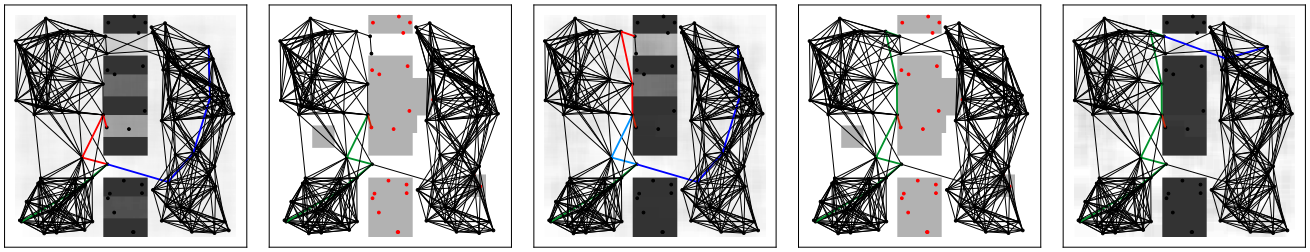
VI. DISCUSSION

Dynamic Replanning with Posterior Sampling is an efficient determinization-based strategy that carefully considers when and how expensive computations with the Bayesian posterior are performed. We analyze how other algorithms within this framework navigate the exploration-exploitation tradeoff inherent to BDMP. Experimentally, shifting the burden of exploration from planning to determinization significantly reduces total planning time—from 4-7× on 2D planning problems to 18× on 7D Baxter manipulator problems. This is generally accompanied by a small improvement in total distance traveled for 2D problems and a 40% improvement in 7D problems.

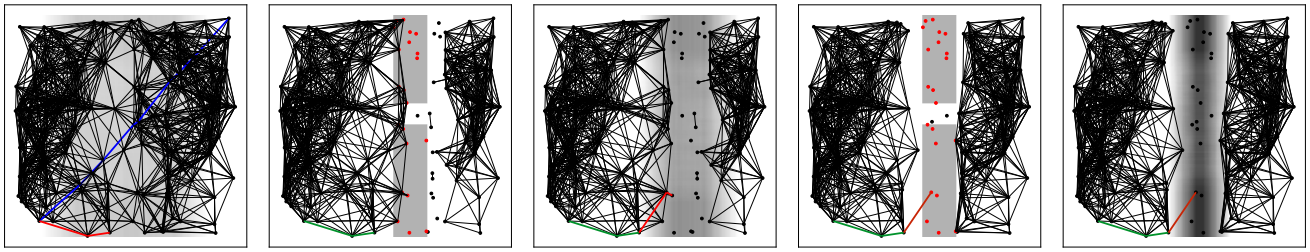
From a practitioner's standpoint, the main task is defining the posterior distribution for BDMP. We believe this is an open representation learning problem: how should one estimate and infer uncertain environments? As long as that distribution supports random sampling, DRPS is simple to implement and free of tuning parameters that manually control the exploration-exploitation tradeoff. Thus, developing generative posterior models of the robot's environment offers an exciting avenue for future work. While it expends additional computation per iteration relative to popular D*-based optimistic strategies (to perform a full posterior update), DRPS effectively leverages information from the updated distribution to quickly plan paths through uncertain environments.

In each iteration, DRPS aims to sample from the combinatorially large space of plausible optimal paths to the goal. Because directly sampling from path space is difficult, DRPS first samples from the space of plausible environments and then plans for the optimal path in that environment. However, progress may stall if the sampled environment does

Fig. 5: Snapshots of HSPD planning progress through two environments.



(a) Given the maximum-likelihood-observation determinization of the posterior distribution, HSPD plans an exploitation path to the goal (blue) and an exploration path that explores the center obstacle region (red). Following the shorter exploration path results in a collision (red), causing an updated posterior and determinization. HSPD plans an exploitation path that backtracks and passes through the passage that is likely to be collision-free. However, HSPD instead follows the shorter exploration path (red) to completion and updates its determinization. At this final iteration, it plans only an exploitation path since there is no suitable exploration path.



(b) On the same environment as Fig. 2, HSPD plans on the maximum-likelihood-observation determinized graph. It follows the shorter exploration path, which updates the posterior distribution. However, the determinization eliminated all the edges that would connect to the goal because they are each *individually* unlikely to be collision-free. While HSPD continues to follow an exploration path, it eventually fails to plan either an exploitation path to the goal or an exploration path that reduces uncertainty. It is limited to the edges available in the maximum-likelihood-observation determinization.

not contain any paths to the goal; no path is attempted in that iteration. Even when the lack of a path plausibly reflects the posterior distribution, this behavior may be undesirable: querying a motion planning algorithm implicitly makes an optimistic assumption that a path to the goal exists, and the algorithm is tasked with finding it. Focusing sampling and planning effort on environments where a path to the goal exists presents an interesting future research direction.

DRPS does not explicitly optimize for combined planning and execution time, i.e., the wall clock time for the robot to reach the goal. (We measured the two components separately and used total distance traveled as a proxy for the latter.) We observe that there is an implicit tradeoff between planning and execution time. Intuitively, if a robot moves slowly, expending additional planning time to propose a shorter path $\hat{\xi}$ may be a worthy tradeoff. The Generalized Lazy Search framework introduces the concept of a toggle between different components of planning [32], which can be extended to toggle between planning and execution. Indeed, the Bayesian lazy motion planning problems considered by prior work, which permit collision-checking anywhere in the environment, define one end of the spectrum where the robot is infinitely fast [17, 23]. Future work that introspects both the planning algorithm and underlying robot platform will be necessary to achieve this gold standard of minimizing wall clock time.

ACKNOWLEDGMENTS

We thank Zhan Wei Lim for sharing his original Hedged Shortest Path under Determinization implementation [13]. We also thank Sandy Kaplan for her valuable feedback.

Dataset	Metric	D*	HSPD	DRPS
OneWall	Time (ms)	44.26 ± 3.6	45.17 ± 2.0	7.31 ± 0.6
	Distance	6.16 ± 0.4	2.54 ± 0.1	2.08 ± 0.1
	Iterations	20.72 ± 1.6	6.21 ± 0.2	5.11 ± 0.4
TwoWall	Time (ms)	170.71 ± 10.7	87.33 ± 4.3	14.00 ± 1.0
	Distance	6.42 ± 0.3	2.08 ± 0.1	2.05 ± 0.1
	Iterations	30.40 ± 1.6	4.22 ± 0.2	2.99 ± 0.2
Forest	Time (ms)	191.93 ± 14.7	123.65 ± 4.1	27.65 ± 1.9
	Distance	6.62 ± 0.4	2.28 ± 0.0	2.19 ± 0.0
	Iterations	30.96 ± 2.3	7.49 ± 0.1	4.67 ± 0.2
MovingWall	Time (ms)	81.28 ± 7.0		27.18 ± 2.1
	Distance	5.49 ± 0.4		3.20 ± 0.1
	Iterations	19.79 ± 1.7		6.62 ± 0.5
Maze	Time (ms)	1019.11 ± 52.6	92.74 ± 3.0	20.31 ± 0.9
	Distance	40.26 ± 2.4	2.99 ± 0.1	3.13 ± 0.1
	Iterations	215.68 ± 12.7	7.20 ± 0.2	5.75 ± 0.2
Baffle	Time (ms)	215.09 ± 10.7	105.88 ± 4.5	22.54 ± 1.5
	Distance	14.22 ± 0.4	3.09 ± 0.1	3.24 ± 0.1
	Iterations	57.61 ± 1.6	6.28 ± 0.2	5.25 ± 0.3
Bugtrap	Time (ms)	242.00 ± 19.7	101.34 ± 3.5	13.59 ± 0.9
	Distance	13.98 ± 1.0	2.85 ± 0.1	2.25 ± 0.1
	Iterations	59.61 ± 4.6	6.27 ± 0.1	2.91 ± 0.2
Baxter (7D)	Time (s)	100.44 ± 5.2	15.37 ± 0.7	0.86 ± 0.06
	Distance	743.55 ± 30.0	28.61 ± 0.6	11.82 ± 0.5
	Iterations	374.94 ± 15.0	8.19 ± 0.2	2.37 ± 0.1

TABLE I: Online Bayesian dynamic motion planning performance on 2D [23] and 7D [17] datasets. We report the 95% confidence interval on the mean for planning time, total distance traveled, and number of iterations. Only successful planning trials are included in these confidence intervals. DRPS and D* succeeded on all planning problems. We have intentionally omitted the performance of HSPD on the MovingWall dataset; it failed to solve 199 of 200 problems. Planning times are reported in milliseconds for 2D problems and in seconds for 7D problems.

REFERENCES

[1] S. Ross, J. Pineau, S. Paquet, and B. Chaib-Draa, "Online planning algorithms for POMDPs," *Journal of Artificial Intelligence Research*, vol. 32, no. 1, pp. 663–704, 2008.

[2] D. Silver and J. Veness, "Monte-Carlo planning in large POMDPs," in *Advances in Neural Information Processing Systems*, 2010.

[3] A. Somani, N. Ye, D. Hsu, and W. S. Lee, "DESPOT: Online POMDP planning with regularization," in *Advances in Neural Information Processing Systems*, 2013.

[4] Z. N. Sunberg and M. J. Kochenderfer, "Online algorithms for POMDPs with continuous state, action, and observation spaces," in *International Conference on Automated Planning and Scheduling*, 2018.

[5] M. Ghavamzadeh, S. Mannor, J. Pineau, and A. Tamar, "Bayesian reinforcement learning: A survey," *Foundations and Trends® in Machine Learning*, vol. 8, no. 5-6, pp. 359–483, 2015.

[6] S. C. Ong, S. W. Png, D. Hsu, and W. S. Lee, "Planning under uncertainty for robotic tasks with mixed observability," *International Journal of Robotics Research*, vol. 29, no. 8, pp. 1053–1068, 2010.

[7] M. Chen, E. Frazzoli, D. Hsu, and W. S. Lee, "POMDP-lite for robust robot planning under uncertainty," in *IEEE International Conference on Robotics and Automation*, 2016.

[8] S. W. Yoon, A. Fern, and R. Givan, "FF-Replan: A baseline for probabilistic planning." in *International Conference on Automated Planning and Scheduling*, 2007.

[9] S. W. Yoon, A. Fern, R. Givan, and S. Kambhampati, "Probabilistic planning via determinization in hindsight," in *AAAI Conference on Artificial Intelligence*, 2008.

[10] A. Stentz, "The focussed D* algorithm for real-time replanning," in *International Joint Conference on Artificial Intelligence*, 1995.

[11] S. Koenig and M. Likhachev, "Improved fast replanning for robot navigation in unknown terrain," in *IEEE International Conference on Robotics and Automation*, 2002.

[12] R. Platt, R. Tedrake, L. P. Kaelbling, and T. Lozano-Pérez, "Belief space planning assuming maximum likelihood observations," in *Robotics: Science and Systems*, 2010.

[13] Z. W. Lim, D. Hsu, and W. S. Lee, "Shortest path under uncertainty: Exploration versus exploitation," in *Conference on Uncertainty in Artificial Intelligence*, 2017.

[14] W. Thompson, "On the likelihood that one unknown probability exceeds another in view of the evidence of two samples," *Biometrika*, vol. 25, no. 3/4, pp. 285–294, 1933.

[15] I. Osband, D. Russo, and B. Van Roy, "(More) efficient reinforcement learning via posterior sampling," in *Advances in Neural Information Processing Systems*, 2013.

[16] K. Wabersich and M. Zeilinger, "Bayesian model predictive control: Efficient model exploration and regret bounds using posterior sampling," in *Conference on Learning for Dynamics & Control*, 2020.

[17] B. Hou, S. Choudhury, G. Lee, A. Mandalika, and S. S. Srinivasa, "Posterior sampling for anytime motion planning on graphs with expensive-to-evaluate edges," in *IEEE International Conference on Robotics and Automation*, 2020.

[18] T. Jaksch, R. Ortner, and P. Auer, "Near-optimal regret bounds for reinforcement learning," *Journal of Machine Learning Research*, vol. 11, no. Apr, pp. 1563–1600, 2010.

[19] M. Strens, "A Bayesian framework for reinforcement learning," in *International Conference on Machine Learning*, 2000.

[20] I. Osband and B. Van Roy, "Why is posterior sampling better than optimism for reinforcement learning?" in *International Conference on Machine Learning*, 2017.

[21] Y. Chen, S. Javdani, A. Karbasi, J. A. Bagnell, S. S. Srinivasa, and A. Krause, "Submodular surrogates for value of information," in *AAAI Conference on Artificial Intelligence*, 2015.

[22] S. Choudhury, S. Javdani, S. S. Srinivasa, and S. Scherer, "Near-optimal edge evaluation in explicit generalized binomial graphs," in *Advances in Neural Information Processing Systems*, 2017.

[23] S. Choudhury, S. S. Srinivasa, and S. Scherer, "Bayesian active edge evaluation on expensive graphs," in *International Joint Conference on Artificial Intelligence*, 2018.

[24] D. Ferguson and A. Stentz, "Field D*: An interpolation-based path planner and replanner," in *International Symposium on Robotics Research*, 2007.

[25] C. H. Papadimitriou and M. Yannakakis, "Shortest paths without a map," *Theoretical Computer Science*, vol. 84, no. 1, pp. 127–150, 1991.

[26] P. Eyerich, T. Keller, and M. Helmert, "High-quality policies for the Canadian traveler's problem," in *AAAI Conference on Artificial Intelligence*, 2010.

[27] J. J. Chung, A. J. Smith, R. Skeele, and G. A. Hollinger, "Risk-aware graph search with dynamic edge cost discovery," *International Journal of Robotics Research*, vol. 38, no. 2-3, pp. 182–195, 2019.

[28] D. Dey, A. Kolobov, R. Caruana, E. Kamar, E. Horvitz, and A. Kapoor, "Gauss meets Canadian traveler: shortest-path problems with correlated natural dynamics," in *International Conference on Autonomous Agents and Multi-agent Systems*, 2014.

[29] B. Saund, S. Choudhury, S. S. Srinivasa, and D. Berenson, "The blindfolded robot: A Bayesian approach to planning with contact feedback," in *International Symposium on Robotics Research*, 2019.

[30] R. A. MacDonald and S. L. Smith, "Reactive motion planning in uncertain environments via mutual information policies," in *Workshop on the Algorithmic Foundations of Robotics*, 2020.

[31] F. Tsang, T. Walker, R. A. MacDonald, A. Sadeghi, and S. L. Smith, "LAMP: Learning a motion policy to repeatedly navigate in an uncertain environment," *IEEE Transactions on Robotics*, pp. 1–15, 2021.

[32] A. Mandalika, S. Choudhury, O. Salzman, and S. Srinivasa, "Generalized Lazy Search for Robot Motion Planning: Interleaving Search and Edge Evaluation via Event-based Toggles," in *International Conference on Automated Planning and Scheduling*, 2019.