Information Particle Filter Tree: An Online Algorithm for POMDPs with Belief-Based Rewards on Continuous Domains

Johannes Fischer [*1 2] **Ömer Şahin Taş** [*1 2]

Abstract

Planning in Partially Observable Markov Decision Processes (POMDPs) inherently gathers the information necessary to act optimally under uncertainties. The framework can be extended to model pure information gathering tasks by considering belief-based rewards. This allows us to use reward shaping to guide POMDP planning to informative beliefs by using a weighted combination of the original reward and the expected information gain as the objective. In this work we propose a novel online algorithm, Information Particle Filter Tree (IPFT), to solve problems with belief-dependent rewards on continuous domains. It simulates particle-based belief trajectories in a Monte Carlo Tree Search (MCTS) approach to construct a search tree in the belief space. The evaluation shows that the consideration of information gain greatly improves the performance in problems where information gathering is an essential part of the optimal policy.

Index Terms — Continuous POMDP, Planning, Monte Carlo Tree Search, Particle Filter Tree, IPFT, Reward Shaping, Information Gathering.

1. Introduction

Uncertainty is decisive in many decision making problems. While information retrieval is the sole objective of some problems, in others it is only a means to disambiguate the current situation and select optimal actions. These types of problem can be modeled with POMDPs. The optimal policy for many POMDPs can be found only approximately since finding the exact solution is computationally intractable (Papadimitriou & Tsitsiklis, 1987). In real-world applications, planning algorithms frequently have to deal with (1) problems on continuous domains and (2) a limited time available for online planning.

Online algorithms for large or continuous state spaces are derived from MCTS, the most prominent ones being POMCP (Silver & Veness, 2010), DESPOT (Somani et al., 2013), and ABT (Kurniawati & Yadav, 2016). Besides having continuous state spaces, many problems in robotics are modeled with continuous observation spaces. The further development of the mentioned algorithms resulted in the state-of-the-art solvers POMCPOW and PFT-DPW (Sunberg & Kochenderfer, 2018) and DESPOT-α (Garg et al., 2019), which can deal with continuous state and observation spaces and therefore solve the first issue.

While MCTS-based algorithms are capable of solving problems online, they can suffer from suboptimal behavior when limited planning time is available. This problem arises when rewards are sparse, e.g. only indicate success or failure at the end of each episode. Hence, the planning algorithm is required to explore many suboptimal paths until finding promising actions, especially in problems with high uncertainty. To resolve this issue, Potential-Based Reward Shaping (PBRS) can be used to implicitly guide the agent to large future rewards (Eck et al., 2016). The potentials are functions over beliefs and reflect how much reward can potentially be gained in a certain situation. This additional guidance considerably speeds up planning and thus addresses the second issue.

In this paper, we propose to combine PBRS with MCTS for solving POMDPs on continuous spaces. This tackles the two issues mentioned for planning in real-world problems. Previous research has only considered reward shaping for problems on discrete spaces. We use potentials based on information measures to guide the agent to informative beliefs. Since information gathering is typically part of the optimal policy, this significantly helps to solve the problem. The resulting belief-based reward functions can be modeled within the ρPOMDP framework, for which solution methods only exist for discrete problems (Araya-López et al., 2010). For this reason, we develop a novel online algorithm, Information Particle Filter Tree (IPFT), for solving ρPOMDPs

[*]Equal contribution [1]Karlsruhe Institute of Technology (KIT), Karlsruhe, Germany [2]FZI Research Center for Information Technology, Karlsruhe, Germany. Correspondence to: Johannes Fischer <johannes.fischer@kit.edu>, Ömer Şahin Taş <tas@fzi.de>.

Proceedings of the 37^{th} International Conference on Machine Learning, Online, PMLR 119, 2020.

on continuous spaces, which enables us to use PBRS. We approximate beliefs with small particle sets and solve an MDP on the belief space with MCTS.

2. Background and Notation

A sequential decision making problem with uncertain system dynamics can be framed as a **Markov Decision Process (MDP)**, represented by the tuple $(\mathbb{S}, \mathbb{A}, \mathcal{T}, \mathcal{R}, \gamma)$, where $\mathbb{S}$ is the state space, $\mathbb{A}$ is the action space, $\mathcal{T}$ is the transition model that defines the distribution of the successor state s', given the current state s and the chosen action a. In a continuous state space this amounts to a probability density. The reward model $\mathcal{R}$ represents the reward obtained when executing an action a in state s, and $\gamma \in [0, 1)$ is a factor discounting future rewards, which ensures a finite cumulative reward.

In some problems, the system state cannot be observed directly. A **Partially Observable Markov Decision Process (POMDP)** addresses this issue by extending the MDP model with an observation space $\mathbb{O}$ and an observation model $\mathcal{Z}$ that models the conditional distribution of receiving an observation o, given state s. In a continuous observation space this distribution can be represented by a density. The probability distribution over the current state s_t given the history $h_t = (b_0, a_0, o_1, \ldots, a_{t-1}, o_t)$ of the initial state distribution b_0 and all previous actions and observations is called the belief $b_t = p(s_t|h_t)$. It can be computed from the transition and observations models using Bayes' theorem. The belief is the basis for decision making in each step, since the true state is unknown. Therefore, policies in POMDPs map beliefs to actions. Furthermore, a POMDP can be equivalently reformulated as an MDP over the belief space, the so-called belief MDP.

An online algorithm capable of solving large MDPs is obtained by combining tree search with Monte Carlo methods. **Monte Carlo Tree Search (MCTS)** incrementally builds up an asymmetric tree by running simulations (Browne et al., 2012). Each tree node represents a state, for which the expected return is estimated, and edges represent transitions to successor states. One simulation consists of the phases: tree traversal, node expansion, rollout, and backpropagation. In each simulation a new node is added to the tree and the values of all visited nodes are refined. During the first phase, a popular action selection strategy to balance exploration and exploitation is the **Upper Confidence Bound (UCB)** (Auer et al., 2002; Kocsis & Szepesvári, 2006).

Silver & Veness developed the online algorithm **Partially Observable Monte Carlo Planning (POMCP)** by adapting MCTS to partially observable environments (2010). In contrast to MCTS, the tree now branches in both actions and observations. Instead of states, the nodes represent histories h, that is, action-observation sequences. Additionally, the simulations building up the tree are also used to obtain a Monte Carlo belief update in between planning steps. The algorithm only requires a generative model and achieves high performance on large problems with discrete actions and observations.

MCTS as described above cannot be applied to problems with large or continuous action spaces. Since the same action is rarely selected twice, this results in a shallow tree where the values cannot be estimated well. Progressive widening is a strategy to tackle this issue (Couëtoux et al., 2011). It limits the number of child nodes considered in a node to $\lceil kN^\alpha \rceil$ in the Nth visit, where $k > 0$ and $\alpha \in (0, 1)$ are parameters. If this limit is reached, one of the previous child nodes is selected. Additionally, this strategy can be applied to the branching induced by stochastic transitions or partial observability and is then referred to as **Double Progressive Widening (DPW)**. Sunberg & Kochenderfer applied DPW in their algorithms Particle Filter Tree (PFT-DPW) and POMCP with Observation Widening (POMCPOW) for solving POMDPs on continuous spaces (2018). POMCPOW simulates single particles and iteratively refines a particle-based belief approximation in every tree node. In contrast, PFT-DPW solves the belief MDP by simulating whole beliefs instead of single particles. A belief is approximated by a fixed number of m weighted particles which is updated as in a particle filter. The immediate reward in each step is computed as the weighted mean of the particle rewards.

Using PBRS requires to plan with belief-based reward functions $\rho(b, a)$ instead of the state-based reward $\mathcal{R}(s, a)$. An example which uses a belief-based reward is the belief MDP, where the reward is the expected state-based reward $\rho(b, a) = \int_{\mathbb{S}} \mathcal{R}(s, a) b(s) \, \mathrm{d}s$. Araya-López et al. proposed **ρPOMDPs** as an extension to POMDPs which use arbitrary belief-based rewards (2010). This allows to define pure information gathering problems as well as the utilization of reward functions combining information gain with state-based rewards, as necessary in the domain of active sensing. Analogously to the belief MDP, a ρPOMDP can be described by an MDP over the belief space with reward ρ. Offline solution methods based on value iteration were proposed to solve ρPOMDPs, but they require discrete state, action, and observation spaces and a reward function that is piecewise linear and convex (PWLC) on the belief space.

3. Information-Theoretic Reward Shaping

Shaping the reward function can result in a different policy. However, policy invariant reward shaping can be achieved by PBRS (Ng et al., 1999; Eck et al., 2016). In this approach, potentials are used as heuristics for the future reward that can be expected in a belief. The reward is shaped by adding

the potential difference of successive beliefs which acts as an implicit guidance for the agent. In our work, we use potentials based on information measures for multiple reasons.

Firstly, in highly uncertain domains the agent has to reduce the uncertainty in the belief before maximizing the reward, even if information gathering as such is not rewarded. This gives rise to the idea that planning can be guided by shaping the reward function to consider the value of information.

Secondly, information retrieval can be rewarded on any domain and does not require a manually designed potential function.

The last motivation for using information-theoretic reward shaping is that the optimal value function V^* serves as a particularly effective potential (Ng et al., 1999). Although V^* is unknown, it is known to be convex over the belief space and typically attains higher values on the boundary of the belief space, since more information allows to make better decisions. Information measures, as defined in the next paragraph, are also convex and should attain their maximal values on the boundary of the belief space, which makes them a reasonable heuristic for V^*.

To quantify the information contained in a distribution we define **information measures** as convex functions on the belief space, denoted by $\mathcal{I}$. The convexity captures the intuitive observation that the information contained in a mixture distribution $\mathcal{I}(\lambda b_1 + (1-\lambda) b_2)$ of beliefs b_1, b_2 with $\lambda \in (0,1)$ cannot exceed the weighted mean information $\lambda \mathcal{I}(b_1) + (1-\lambda)\mathcal{I}(b_2)$ of the distributions b_1, b_2.

Based on the previous considerations, we investigate and compare two slightly different potential-based shaping functions in this work. To begin with, we use the **discounted information gain** $\Delta\mathcal{I}_\gamma(b, b') = \gamma\mathcal{I}(b') - \mathcal{I}(b)$ because it guarantees the optimal policy to be invariant under PBRS for infinite horizon problems (Eck et al., 2016). We additionally consider the **undiscounted information gain** $\Delta\mathcal{I}_1(b, b') = \mathcal{I}(b') - \mathcal{I}(b)$ since it resembles the idea of gathering information more intuitively and it is used in the area of active sensing (Mihaylova et al., 2003). The shaped reward function can be written as

$$\rho(b, a, b') = \int_{\mathbb{S}} \mathcal{R}(s, a) b(s) \, \mathrm{d}s + \lambda \Delta\mathcal{I}(b, b') \qquad (1)$$

where the parameter λ weights reward maximization and information gathering.

Various information measures are presented in the literature. A widely used uncertainty measure for discrete probability distributions is the **entropy** $\mathcal{H} = -\sum_x p(x) \log p(x)$. It can be generalized to continuous distributions by integrating over the continuous domain $\mathcal{H} = -\int p(x) \log p(x) \, \mathrm{d}x$ and is then called **differential entropy**. While this loses some properties (e.g. positivity), it still serves well as an uncertainty measure.

Other information measures we considered are based on L_p-norms or the distance to the maximum entropy distribution, which represents maximal uncertainty. Similar ideas for discrete spaces are presented in (Mihaylova et al., 2003; Araya-López et al., 2010). Since the results of our evaluations were similar for all information measures considered, we only present the results for negative entropy. The variance of a belief, however, is unsuitable for measuring information. This is because a bimodal distribution with extremely narrow peaks at an arbitrarily large distance contains a lot of information but has unbounded variance.

4. Solving ρPOMDPs on Continuous Spaces

The information-based reward shaping described in the previous section can be implemented by modeling the problem as a ρPOMDP with the belief-based reward function in Equation (1). Araya-López et al. introduced an offline algorithm for solving ρPOMDPs on discrete spaces with PWLC reward function. Since we consider reward shaping for continuous problems and the negative entropy is not PWLC, we develop a novel ρPOMDP solver for continuous domains.

Our approach adapts the PFT-DPW algorithm presented in Section 2 to belief-based reward functions. Before we present our algorithm, we show how belief-based reward functions can be evaluated on particle-based belief approximations, which are used by PFT-DPW.

4.1. Particle-Based Calculation of Belief-Dependent Rewards

While the expected reward $\int_{\mathbb{S}} \mathcal{R}(s, a) b(s) \, \mathrm{d}s$ and the negative entropy on discrete spaces can be easily evaluated for a sample of the belief, there is no straightforward way to compute the differential entropy from a particle set without making additional assumptions. In this section, we derive a method to compute the negative entropy from weighted particle sets based on kernel density estimation (KDE) (Gisbert, 2003).

In the following, we assume a belief b is approximated by a weighted particle set $\{(s_i, w_i)\}_{i=1}^m$ with normalized weights. As common for particle filters, the integral is approximated as

$$-\mathcal{H}(b) = \int_S b(s) \log b(s) \, \mathrm{d}s \approx \sum_{i=1}^{m} w_i \log b(s_i). \qquad (2)$$

Following the insights of Hall & Morton for unweighted particles (1993), the belief b can be approximated by a KDE $\hat{b}$ computed from the weighted particle set, which yields the

entropy estimate

$$-\hat{\mathcal{H}}(\hat{b}) = \sum_{i=1}^{m} w_i \log \hat{b}(s_i). \tag{3}$$

In our work we use a Gaussian kernel and select the bandwidth according to Silverman's rule of thumb (Silverman, 1986). The computational complexity of evaluating Equation (3) is $\mathcal{O}(m^2 D)$ where D is the dimension of the continuous state space. Details on KDE, bandwidth selection and computational complexity can be found in the supplemental material.

An alternative method to calculate the entropy of a running particle filter can be derived based on Bayes' theorem (Skoglar et al., 2009; Boers et al., 2010). However, this procedure requires explicit knowledge of transition and observation models, which is often not available.

4.2. The Information Particle Filter Tree Algorithm

Similar to PFT-DPW, our algorithm simulates particle sets in a MCTS fashion to construct a search tree. One iteration of IPFT is shown in Figure 1. For simplicity only belief nodes are shown, not action nodes. Particle sets of fixed size m are simulated through the tree, until reaching a node with unexplored child nodes. Then a new node is added to the tree and its value is estimated using a rollout policy. All visited nodes are updated with the returned reward.

The principal change necessary in PFT-DPW to solve ρPOMDPs is the reward computation. Instead of the mean particle reward, the belief-dependent reward model $\rho(b, a, b')$ is used. Information rewards are calculated using the particle-based approximations derived in the previous section.

Another aspect concerns the particle-based belief approximations. In PFT-DPW particle sets are only generated when a new node is added to the tree. The particles are then saved in the node and reused when it is visited again. This is problematic because a small particle set serves only as a coarse approximation of the continuous belief. Hence, belief-based rewards like the entropy cannot be estimated well from only a small sample. For this reason, in IPFT we simulate new particle beliefs instead of reusing previous samples. The algorithm averages the rewards of different particle approximations of the same belief to obtain a better estimate.

4.2.1. Convergence of Belief-Based Rewards

The belief-based reward in Equation (1) consists of the mean state-based reward and the entropy difference. Regarding the mean particle reward, averaging over multiple particle sets yields the same result as computing the mean reward of the union of all particle sets. However, this might not be true for the entropy-based reward.

In the following, we analyze how averaging entropy estimates from K different particle sets $S_k = \{(s_i^{(k)}, w_i^{(k)})\}_{i=1}^{m}$ differs from computing the entropy from the combined sample $S = \bigcup S_k$. By concatenating the K particle sets, the union can be formulated as $S = \{(\tilde{s_j}, \tilde{w_j})\}_{j=1}^{Km}$ where the weights $\tilde{w_j}$ are normalized with the factor $\frac{1}{K}$. Let $\hat{b}_k$ and $\hat{b}$ denote kernel density estimates for the sample S_k and the combined sample S, respectively. Averaging the negative entropy estimates from the different particle sets results in

$$-\frac{1}{K}\sum_{k=1}^{K} \hat{\mathcal{H}}(\hat{b}_k) = \sum_{k=1}^{K}\sum_{i=1}^{m} \frac{w_i^{(k)}}{K} \log \hat{b}_k(s_i^{(k)}). \tag{4}$$

Since $\hat{b}$ and $\hat{b}_k$ are density estimates of the same belief b, they converge to the same density for $m \to \infty$. For this reason, we assume them to be sufficiently close so that Equation (4) can be approximated by

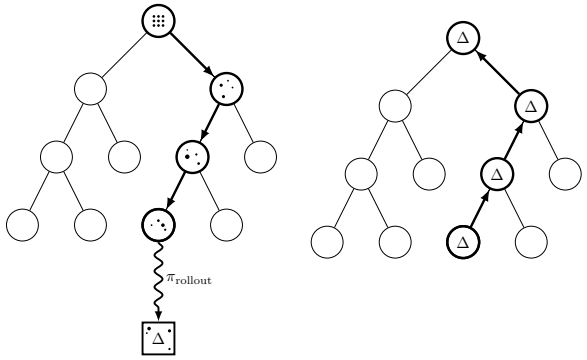
$$\begin{aligned} &\approx \sum_{k=1}^{K}\sum_{i=1}^{m} \frac{w_i^{(k)}}{K} \log \hat{b}(s_i^{(k)}) \\ &= \sum_{j=1}^{K\cdot m} \tilde{w_j} \log \hat{b}(\tilde{s_j}) = -\hat{\mathcal{H}}(\hat{b}). \end{aligned} \tag{5}$$

Hence, averaging the entropy estimates of different particle sets is approximately equal to estimating the entropy from the combined particle set. Since the KDE-based entropy estimate (5) converges for $K \to \infty$, we expect our estimate to converge as well (Hall & Morton, 1993).

Estimating the entropy from the combined sample would require to maintain all particles sampled in any tree node. In contrast, our approach only needs the currently simulated particle set, which drastically reduces the required memory. Moreover, it is computationally more efficient, because the entropy computation as described in Section 4.1 scales quadratically with the number of particles m.

4.2.2. Description of the Algorithm

Algorithm 1 presents the resulting method. The outer structure of the algorithm is left out due to space constraints. It closely follows the outer structure of the algorithms provided by Sunberg & Kochenderfer (2017): The SIMULATE procedure is called from the initial belief b, an empty action-observation history h, and the maximal tree depth d as long as computational resources permit, which over time builds up the search tree. Since beliefs are represented by particle sets, the initial belief is sampled at each call of SIMULATE in the root node. In the following, a history h appended by a new action a or an action-observation pair (a, o) is denoted ha or hao, respectively.



a) Simulation of tree and rollout policy

b) Backpropagation

Figure 1. One iteration of the IPFT algorithm. Particle sets are simulated through the tree using UCB. At the end a node is expanded and its value estimated with a rollout policy. Then the value is propagated back through all visited nodes.

The first step in SIMULATE is to choose an action. Progressive widening of the action space and action selection according to UCB is covered by the function ACTIONPROGWIDEN. Then either a new observation is generated or an existing observation child node is sampled, depending on the observation widening parameters, similar as in POMCPOW. Here, $C(ha)$ denotes the set of child nodes of the tree node corresponding to history ha, which are identified by the observation o that leads to the corresponding child node. To generate a new observation, a state is sampled from the belief b at first, for which a successor state and the observation are generated. Note, that the counter M is only needed in the case of discrete observations, when the same observation can be sampled multiple times. In continuous observations spaces, M will almost surely be one for all nodes. Then the posterior belief b' is computed with a particle filter update $\mathcal{G}_{\mathrm{PF}(m)}$ and the belief-dependent reward is calculated. If a new observation was sampled, it is added as a child node and a rollout is performed to estimate the total reward R of the belief node. Otherwise, SIMULATE is called recursively on the belief b' and the successor node hao. At last, the node statistics are updated.

The computational complexity of the particle filter update is $\mathcal{O}(m)$. For each particle set sampled from the initial belief in the root node, SIMULATE is called in every node while traversing the tree. Hence, the complexity of IPFT is $\mathcal{O}(ndm)$ for n particle sets simulated in a tree of depth d. Further, IPFT is well suited for parallelization since the particle filter updates can be distributed on different workers which allows to use larger particle sets. In our experiments small particles sets of size $m = 20$ were sufficient.

Algorithm 1 SIMULATE function of Information Particle Filter Tree

Input: belief b, history h, depth d
if $d = 0$ **then**
  $R = 0$
else
  $a \leftarrow$ ACTIONPROGWIDEN(h)
  if $|C(ha)| \leq k_o N(ha)^{\alpha_o}$ **then**
    $o \leftarrow$ sample s from b, generate o from (s, a)
    $M(hao) \leftarrow M(hao) + 1$
  else
    $o \leftarrow$ select $o \in C(ha)$ w.p. $\frac{M(hao)}{\sum_{\tilde{o}} M(ha\tilde{o})}$
  end if
  $b' \leftarrow \mathcal{G}_{\mathrm{PF}(m)}(b, a, o)$
  $r \leftarrow \rho(b, a, b')$
  if $o \notin C(ha)$ **then**
    $C(ha) \leftarrow C(ha) \cup \{o\}$
    $R \leftarrow r + \gamma \cdot$ ROLLOUT$(b', hao, d-1)$
  else
    $R \leftarrow r + \gamma \cdot$ SIMULATE$(b', hao, d-1)$
  end if
  $N(h) \leftarrow N(h) + 1$
  $N(ha) \leftarrow N(ha) + 1$
  $Q(ha) \leftarrow Q(ha) + \frac{R - Q(ha)}{N(ha)}$
end if
Output: accumulated discounted reward R

4.2.3. CONVERGENCE OF IPFT

In the following, we want to provide an intuition for the convergence of IPFT. The convergence results for Partially Observable Weighted Sparse Sampling (POWSS) suggest that the particle weighting scheme used in POMCPOW is sound (Lim et al., 2020). Since IPFT samples new weighted particles in every simulation rather than reusing a previous sample, it is closely related to POMCPOW in this aspect. For this reason, IPFT can also be expected to converge to the optimal policy, provided the belief-based rewards are estimated well. For the case of entropy-based rewards this was shown previously in Section 4.2.1.

Furthermore, the optimal policy for the infinite horizon problem is invariant under information-theoretic reward shaping with the discounted information gain (Eck et al., 2016). Hence, the policy IPFT converges to is also optimal for the unshaped infinite horizon problem.

5. Evaluation

We evaluate the effects of information-based reward shaping by comparing the proposed IPFT algorithm with state-of-the-art solvers on benchmark problems of varying difficulty using the POMDPs.jl framework (Egorov et al., 2017). Fur-

thermore, we inspect the sensitivity with respect to its hyperparameters. The code for IPFT and the scenarios and evaluations is provided to the reader at GitHub[1] for further benchmarks.

PFT-DPW and POMCPOW are selected as benchmark solvers because they are designed to deal with continuous spaces and are promising recent developments (Sunberg & Kochenderfer, 2018). Besides, IPFT was developed based on PFT-DPW, therefore comparing them directly reveals the effects of the additional information gathering term. Other online POMDP algorithms such as DESPOT or POMCP with DPW are not well suited for problems with continuous observation spaces and are therefore not included (Sunberg & Kochenderfer, 2018). The DESPOT-α algorithm which was published only recently would be another interesting comparison, but no off-the-shelf implementation is available yet (Garg et al., 2019).

To encourage information gathering, we use reward shaping with the discounted information gain $\Delta\mathcal{I}_\gamma$ as well as the undiscounted information gain $\Delta\mathcal{I}_1$ as described in Section 3. We measure the performance with respect to the original reward of the POMDP, without the shaping reward.

5.1. Benchmark Problems

For every problem we perform 1000 simulations with each solver and average their reward. In the beginning of each simulation the problem is initialized randomly and actions are selected according to the solver, with a maximum computation time of 1 second per step. After an action is executed the agent's belief is updated with the received observation and the next action is selected until the simulation terminates. The hyperparameters used by the solvers are provided in the supplemental material. In all problems the discount was set to $\gamma = 0.95$. We performed the simulations on an Intel Core i7-6700 CPU with a clock rate of 3.4GHz and 8GB RAM.

5.1.1. Light Dark

Different variations of the Light Dark problem are used in the literature on planning under uncertainty (Perez et al., 2012; Platt, 2013; Sunberg & Kochenderfer, 2018). In our work, we consider the variant used by Sunberg & Kochenderfer, where the agent moves deterministically along a one-dimensional discrete state space.

In this problem, the goal is to reach the origin and execute action 0 to obtain a reward of 100. If action 0 is used anywhere else, the agent receives a reward of -100. For moving, the agent has step costs of -1. The problem terminates when action 0 is executed. The observations received by the agent

[1] https://github.com/johannes-fischer/icml2020_ipft

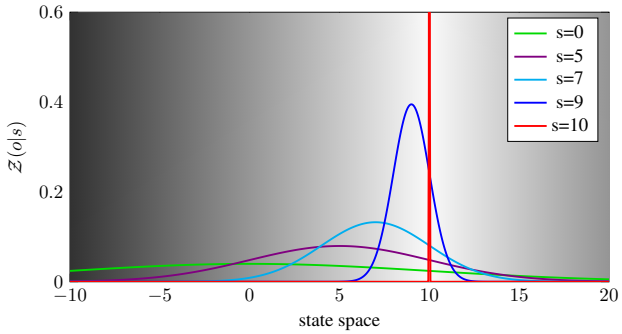


Figure 2. The Light Dark observation density for different successor states s'.

are its current state disturbed by an additive Gaussian noise where the standard deviation $\sigma_\mathcal{Z}(s) = |s - 10|$ grows linearly with the distance from a light source at $s = 10$. Figure 2 shows the probability density of the observation model $\mathcal{Z}$ for some states s. In addition to the original action space $\mathbb{A}_{10} = \{-10, -1, 0, 1, 10\}$, we consider the action space $\mathbb{A}_3 = \{-3, -1, 0, 1, 3\}$ as well.

Table 1 reveals the reward and an estimate of its standard deviation for the Light Dark problem. It can be seen that the variant with action space $\mathbb{A}_3$ is generally harder to solve. This is because it requires multiple steps to get from the light source to the goal.

For the action space $\mathbb{A}_{10}$ the addition of information rewards does not lead to an increased performance over the state-of-the-art solvers. However, POMCPOW fails to solve the version with action space $\mathbb{A}_3$. Besides, the difference between the discounted information gain $\Delta\mathcal{I}_\gamma$ and the undiscounted information gain $\Delta\mathcal{I}_1$ is small.

5.1.2. Continuous Light Dark

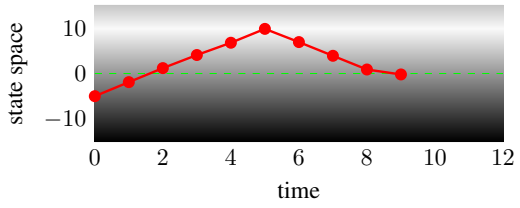
The Light Dark problem is already challenging on a discrete state space. To further increase the difficulty, we remodel the problem on a continuous state space with a stochastic transition and an increased observation noise.

In the Continuous Light Dark problem the agent moves along the real line with steps that are normally distributed around the chosen action with noise $\sigma_\mathcal{T} = 0.1$. As in the previous problem, the two action spaces $\mathbb{A}_{10}$ and $\mathbb{A}_3$ are considered. Since the origin is a null set in the continuous state space, a unit ball around the origin is chosen as the goal area where the agent receives the positive reward. While the observations remain normally distributed, their standard deviation is increased to $\sigma_\mathcal{Z}(s) = \sqrt{2}|s-10|+0.5$, leading to less informative observations.

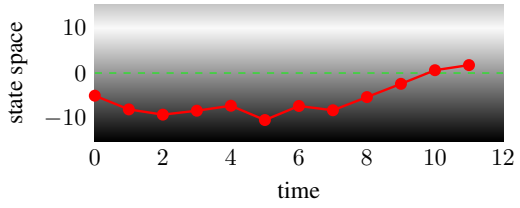
The results for the Continuous Light Dark problem for each of the two action spaces are also listed in Table 1. While

Table 1. Results for the discrete Light Dark and Continuous Light Dark problems for the two actions spaces $\mathbb{A}_{10}$ and $\mathbb{A}_3$. The table shows the mean reward of 1000 simulations and its estimated standard deviation.

	Light Dark problem		Continuous Light Dark problem	
Algorithm	action space $\mathbb{A}_{10}$	action space $\mathbb{A}_3$	action space $\mathbb{A}_{10}$	action space $\mathbb{A}_3$
IPFT($\Delta\mathcal{I}_1$)	58.2 ± 0.4	34.8 ± 0.7	35.7 ± 1.8	35.9 ± 1.0
IPFT($\Delta\mathcal{I}_\gamma$)	55.4 ± 0.5	27.8 ± 0.8	38.4 ± 1.7	32.3 ± 1.4
POMCPOW	58.6 ± 0.5	-2.6 ± 0.9	-8.5 ± 2.3	-2.9 ± 2.1
PFT-DPW	57.4 ± 0.5	33.9 ± 0.8	-33.1 ± 2.4	-19.6 ± 2.3



a) Actions generated by IPFT



b) Actions generated by POMCPOW

Figure 3. Exemplary trajectories for the continuous Light Dark problem with action space $\mathbb{A}_3$. Both trajectories start in $s_0 = -5$.

IPFT still performs well, the benchmark solvers cannot solve the problem and achieve bad results, even PFT-DPW. Both variants of the problem exhibit similar results and as for the discrete Light Dark problem the difference between using $\Delta\mathcal{I}_1$ and $\Delta\mathcal{I}_\gamma$ is small. Our findings indicate that information-based reward shaping helps to solve problems where the optimal policy involves information gathering, even in simple problems like this. Hence, the benefit might be even larger for more complex problems.

Exemplary trajectories for the problem with action space $\mathbb{A}_3$ are shown in Figure 3. Both trajectories initially start in state $s_0 = -5$. Figure 3a clearly illustrates how the trajectory directly moves towards the light source, because IPFT chooses actions that take information gain into account. After localization, it is able to immediately approach the goal. In contrast, the trajectory generated by POMCPOW in Figure 3b moves back and forth unsystematically and thus collects only noisy observations. Over time many observations still improve the belief estimate such that the trajectory ends close to the goal region eventually.

Table 2. Results for the Laser Tag problem. The table shows the mean reward of 1000 simulations and its estimated standard deviation.

	Laser Tag
IPFT($\Delta\mathcal{I}_1$)	-9.0 ± 0.2
IPFT($\Delta\mathcal{I}_\gamma$)	-8.9 ± 0.2
POMCPOW	-9.9 ± 0.2
PFT-DPW	-12.0 ± 0.2

5.1.3. Laser Tag

Somani et al. introduced the Laser Tag problem in which the agent moves in a grid world containing obstacles and a target is moving away from the agent (2013). Its goal is to find and tag the target for which it receives a reward of 10, while each step is penalized with -1. To localize the target, the agent receives normally distributed measurements of the distance to the closest obstacles in the eight cardinal directions.

As Table 2 shows, IPFT achieves higher rewards than the benchmark solvers. While it outperforms PFT-DPW, the result of POMCPOW is only slightly worse. In general, the performance gap is not as significant as for the Continuous Light Dark problem. The reason for this is that both terms in the shaped reward function in Equation (1) are maximized by the same actions: Searching for the target increases the chance for high rewards as well as gathers information. Hence, adding the information reward does not provide as much additional guidance as in other problems.

5.2. Empirical Parameter Sensitivity Analysis

We investigate the sensitivity of the IPFT solver with respect to the number of particles m and the weight λ balancing information gathering and reward maximization. To this end, we carry out more experiments on the Continuous Light Dark problem with action space $\mathbb{A}_3$ where those parameters are varied. The undiscounted information gain $\Delta\mathcal{I}_1$ is used for reward shaping.

Figure 4 shows the results on two separate axes. It can be seen that the performance degrades gracefully with varying

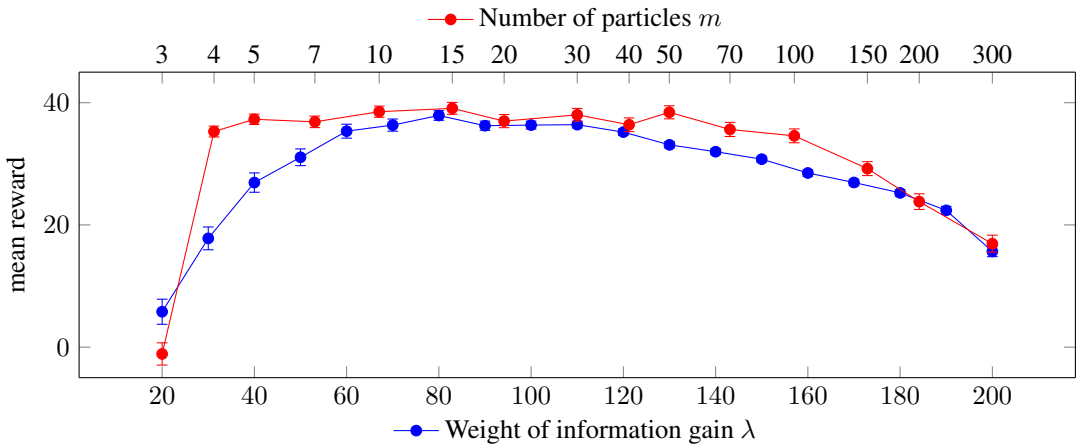


Figure 4. Parameter sensitivity of the mean reward of 1000 simulations. Shown are the results for different values of the information weight λ (blue) and for different numbers of particles m (red). Error bars indicate the estimated standard deviation of the mean reward.

λ, while keeping a fixed number of $m = 20$ particles (blue). For $\lambda \to 0$, IPFT and hence its performance converge to PFT-DPW. With increasing λ the information gain is overweighted and less rewards are collected. Nonetheless, there is a large region ($\lambda \in [60, 120]$) in between those two extremes where the performance is not very sensitive to the choice of λ, which makes it easy to find a suitable parameter in practice.

The number of particles m is varied using a constant information weight $\lambda = 60$ (red). For larger particle sets with $m > 100$, the performance degrades because the increased computation time $\mathcal{O}(m^2)$ necessary for each entropy calculation outweighs the increased accuracy of the particle set approximation. In this scenario the algorithm also works well with as few as $m = 4$ particles. Since the region where the maximal reward is attained is relatively flat, the reward is not very sensitive with respect to m.

6. Related Work

Information gathering has been integrated in sequential decision making problems before, in particular in the areas of exploration and active sensing.

Cassandra et al. used POMDPs to solve a localization problem (1996). They guide the search by choosing an information gathering action whenever the entropy exceeds a threshold. Another work used an optimization criterion combining the value function with the one-step expected entropy, resulting in a myopic algorithm (Burgard et al., 1997). Kreucher et al. maximized the Rényi divergence to choose the most informative sensor (2005). They approximated distributions over a continuous state space with a particle filter. Hoffmann et al. estimate information theoretic costs from particle filters and maximize the mutual information of multiple agents in the next time step to localize an object with decentralized planning (2006). In contrast to our work, these approaches result in myopic, information-greedy policies because they only consider the information gained in the next step.

Some works were able to consider information gathering in long-term planning. Roy & Thrun augmented the state with the entropy and solved the resulting problem with dynamic programming (2000). Unlike our work, they used a parametric approach which allows only unimodal beliefs. Another strategy to obtain non-myopic policies is to sample subgoals where the observation distribution has a low entropy and the immediate reward is high (He et al., 2010; Ma & Pineau, 2015). However, the uncertainty in the observation distribution does not necessarily reflect how much information is actually gained. Spaan et al. proposed to implement state-based information rewards by enlarging the action space with so-called commit actions, which result in high rewards if the state can be guessed correctly (2015). Because one such commit action is necessary for every state that has to be distinguished, the action space quickly becomes intractable. Dressel & Kochenderfer implemented belief-dependent rewards in the offline POMDP solver SARSOP to solve localization problems (2017). Besides being unsuitable for online planning, this also restricts the reward function to be PWLC. They chose reward functions based on the ℓ_∞-norm and on the commit actions from Spaan et al. mentioned above.

Reward shaping based on information maximization was used by Mafi et al. in the context of reinforcement learning (2011). Similar to our work, their reward function combines information gain and task execution. Eck et al. applied PBRS to online planning in POMDPs (2016). They describe four types of belief-based potential functions that can

be used to guide the agent and provide theoretical results. A MCTS-based online planning approach using information-theoretic rewards is provided in the context of robotic exploration problems (Lauri & Ritala, 2016). They resort to open-loop planning to make problems with large observation spaces tractable. The POMDP-lite algorithm solves the subclass of POMDPs where the unobservable states are constant or change deterministically (Chen et al., 2016). It measures the information gain as the L_1 divergence between consecutive beliefs and adds it as a bonus to the reward. Saborío & Hertzberg develop the idea of partial goal satisfaction to construct a domain-independent heuristic for PBRS (2018). They apply PBRS to states sampled by POMCP instead of planning directly over beliefs. In contrast to our work, all of these approaches consider only discrete state spaces.

The referenced works indicate the benefits of utilizing information measures in decision making. Nevertheless, none of them could efficiently do closed-loop online planning with arbitrary belief-based rewards on continuous domains.

7. Conclusion and Future Work

Decision making under uncertainty is particularly challenging in problems on large spaces where the reward is only obtained after long action sequences of information gathering. In this work, we propose to consider the value of information in the objective function in order to facilitate the search for promising actions. We develop a novel online algorithm that enables information-theoretic reward shaping on continuous spaces. Our algorithm, IPFT, performs MCTS with particle-based belief simulations and utilizes DPW to accommodate continuous action and observation spaces. As a result, the algorithm is capable of solving arbitrary ρPOMDPs on continuous domains.

Our evaluation reveals that problems which require information gathering for task execution can be solved much more efficiently using our approach than with state-of-the-art algorithms. Furthermore, the algorithm can be easily used on new problems since our analysis shows that the performance is not very sensitive to its hyperparamters.

In our future research we will use IPFT to investigate more complex scenarios. For instance, search and rescue is an important application which likely benefits from information-based reward shaping. Since we developed a universal ρPOMDP solver this allows us to also consider pure information gathering tasks on continuous spaces, as are common in active sensing.

Acknowledgements

The authors gratefully acknowledge the fruitful discussions and review of this work by Martin Lauer on the original version of this document.

References

Araya-López, M., Buffet, O., Thomas, V., and Charpillet, F. A POMDP Extension with Belief-dependent Rewards. In *Advances in Neural Information Processing Systems 23*, pp. 64–72, 2010.

Auer, P., Cesa-Bianchi, N., and Fischer, P. Finite-time Analysis of the Multiarmed Bandit Problem. *Mach. Learn.*, 47(2-3):235–256, May 2002.

Boers, Y., Driessen, H., Bagchi, A., and Mandal, P. Particle Filter Based Entropy. In *Proc. of the IEEE International Conference on Information Fusion*, pp. 1–8, July 2010.

Browne, C. B., Powley, E., Whitehouse, D., Lucas, S. M., Cowling, P. I., Rohlfshagen, P., Tavener, S., Perez, D., Samothrakis, S., and Colton, S. A Survey of Monte Carlo Tree Search Methods. *IEEE Trans. on Computational Intelligence and AI in Games*, 4(1):1–43, March 2012.

Burgard, W., Fox, D., and Thrun, S. Active mobile robot localization by entropy minimization. In *Proc. EUROMICRO Workshop on Advanced Mobile Robots*, pp. 155–162, October 1997.

Cassandra, A. R., Kaelbling, L. P., and Kurien, J. A. Acting under Uncertainty: Discrete Bayesian Models for Mobile Robot Navigation. In *Proc. IEEE International Conference on Intelligent Robots and Systems*, volume 2, pp. 963–972, 1996.

Chen, M., Frazzoli, E., Hsu, D., and Lee, W. S. POMDP-lite for robust robot planning under uncertainty. In *Proc. IEEE International Conference on Intelligent Robots and Systems*, pp. 5427–5433, 2016.

Couëtoux, A., Hoock, J.-B., Sokolovska, N., Teytaud, O., and Bonnard, N. Continuous Upper Confidence Trees. In *Proc. of the International Conference on Learning and Intelligent Optimization*, pp. 433–445, 2011.

Dressel, L. and Kochenderfer, M. J. Efficient Decision-Theoretic Target Localization. In *International Conference on Automated Planning and Scheduling*, pp. 9, 2017.

Eck, A., Soh, L.-K., Devlin, S., and Kudenko, D. Potential-based reward shaping for finite horizon online POMDP planning. *Autonomous Agents and Multi-Agent Systems*, 30:403–445, May 2016.

Egorov, M., Sunberg, Z. N., Balaban, E., Wheeler, T. A., Gupta, J. K., and Kochenderfer, M. J. POMDPs.jl: A Framework for Sequential Decision Making under Uncertainty. *Journal of Machine Learning Research*, 18(26), 2017.

Garg, N. P., Hsu, D., and Lee, W. S. DESPOT-Alpha: Online POMDP planning with large state and observation spaces. In *Proc. of Robotics: Science and Systems*, June 2019.

Gisbert, F. J. G. Weighted samples, kernel density estimators and convergence. *Empirical Economics*, 28(2):335–351, 2003.

Hall, P. and Morton, S. C. On the Estimation of Entropy. *Annals of the Institute of Statistical Mathematics*, 45(1): 69–88, 1993.

He, R., Brunskill, E., and Roy, N. PUMA: Planning under Uncertainty with Macro-Actions. In *Proc. of the AAAI Conference on Artificial Intelligence*, pp. 1089–1095, 2010.

Hoffmann, G., Waslander, S., and Tomlin, C. Distributed Cooperative Search Using Information - Theoretic Costs for Particle Filters, with Quadrotor Applications. In *AIAA Guidance, Navigation, and Control Conference and Exhibit*, August 2006.

Kocsis, L. and Szepesvári, C. Bandit Based Monte-Carlo Planning. In *European Conference on Machine Learning*, pp. 282–293, 2006.

Kreucher, C., Kastella, K., and Hero III, A. O. Sensor Management Using an Active Sensing Approach. *Signal Processing*, 85(3):607–624, March 2005.

Kurniawati, H. and Yadav, V. *An Online POMDP Solver for Uncertainty Planning in Dynamic Environment*, pp. 611–629. Springer International Publishing, Cham, 2016.

Lauri, M. and Ritala, R. Planning for robotic exploration based on forward simulation. *Robotics and Autonomous Systems*, 83:15–31, September 2016.

Lim, M. H., Tomlin, C. J., and Sunberg, Z. N. Sparse tree search optimality guarantees in POMDPs with continuous observation spaces. In *International Joint Conference on Artificial Intelligence (IJCAI)*, 2020.

Ma, H. and Pineau, J. Information Gathering and Reward Exploitation of Subgoals for POMDPs. In *Proc. of the AAAI Conference on Artificial Intelligence*, pp. 3320–3326, 2015.

Mafi, N., Abtahi, F., and Fasel, I. Information theoretic reward shaping for curiosity driven learning in POMDPs. In *Proc. IEEE International Conference on Development and Learning*, volume 2, pp. 1–7, August 2011.

Mihaylova, L., Lefebvre, T., Bruyninckx, H., Gadeyne, K., and De Schutter, J. A Comparison of Decision Making Criteria and Optimization Methods for Active Robotic Sensing. In *Numerical Methods and Applications*, volume 2542, pp. 316–324. 2003.

Ng, A. Y., Harada, D., and Russell, S. J. Policy invariance under reward transformations: Theory and application to reward shaping. In *Proceedings of the International Conference on Machine Learning*, pp. 278–287, 1999.

Papadimitriou, C. H. and Tsitsiklis, J. N. The Complexity of Markov Decision Processes. *Mathematics of Operations Research*, 12(3):441–450, 1987.

Perez, A., Platt, R., Konidaris, G., Kaelbling, L., and Lozano-Perez, T. LQR-RRT*: Optimal sampling-based motion planning with automatically derived extension heuristics. In *IEEE International Conference on Robotics and Automation*, pp. 2537–2542, 2012.

Platt, R. Convex receding horizon control in non-gaussian belief space. *Algorithmic Foundations of Robotics X*, pp. 443–458, 2013.

Roy, N. and Thrun, S. Coastal Navigation with Mobile Robots. In Solla, S. A., Leen, T. K., and Müller, K. (eds.), *Advances in Neural Information Processing Systems*, pp. 1043–1049, 2000.

Saborío, J. C. and Hertzberg, J. Towards Domain-independent Biases for Action Selection in Robotic Task-planning under Uncertainty:. In *Proc. of the International Conference on Agents and Artificial Intelligence*, pp. 85–93, 2018.

Silver, D. and Veness, J. Monte-Carlo planning in large POMDPs. In *Advances in Neural Information Processing Systems*, pp. 2164–2172, 2010.

Silverman, B. W. *Density Estimation for Statistics and Data Analysis*. Chapman & Hall, London, 1986.

Skoglar, P., Orguner, U., and Gustafsson, F. On Information Measures Based on Particle Mixture for Optimal Bearings-Only Tracking. In *In Proc. of the IEEE Aerospace Conference*, pp. 1–14, 2009.

Somani, A., Ye, N., Hsu, D., and Lee, W. S. DESPOT: Online POMDP planning with regularization. In *Advances in Neural Information Processing Systems*, pp. 1772–1780, 2013.

Spaan, M. T. J., Veiga, T. S., and Lima, P. U. Decision-Theoretic Planning under Uncertainty with Information

Rewards for Active Cooperative Perception. *Autonomous Agents and Multi-Agent Systems*, 29(6), November 2015.

Sunberg, Z. and Kochenderfer, M. Online algorithms for POMDPs with continuous state, action, and observation spaces (extended version). *CoRR*, abs/1709.06196, 2017.

Sunberg, Z. and Kochenderfer, M. Online Algorithms for POMDPs with Continuous State, Action, and Observation Spaces. In *Proceedings of the International Conference on Automated Planning and Scheduling*, pp. 259–263, 2018.