The Henry and Marilyn Taub
Faculty of Computer Science

Online Partially Observable Markov Decision Process Planning via Simplification

Ori Sztyglic

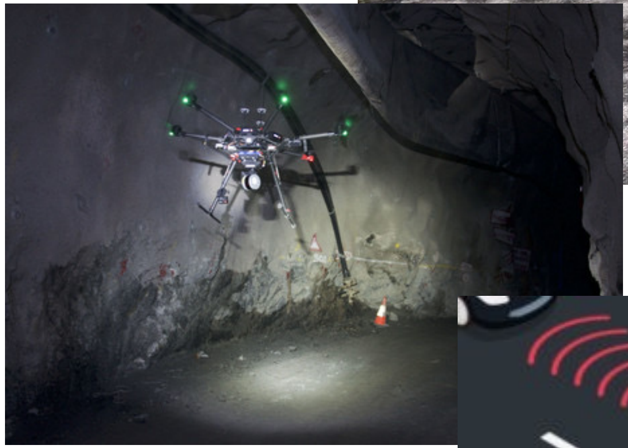
Advisor: Associate Prof. Vadim Indelman

Autonomous Navigation and Perception Lab

Technion | The Henry and Marilyn Taub Faculty of Computer Science | ANPL Autonomous Navigation and Perception Lab

Motivation

- Autonomous Agents
- Planning Under Uncertainty
- Online Agents



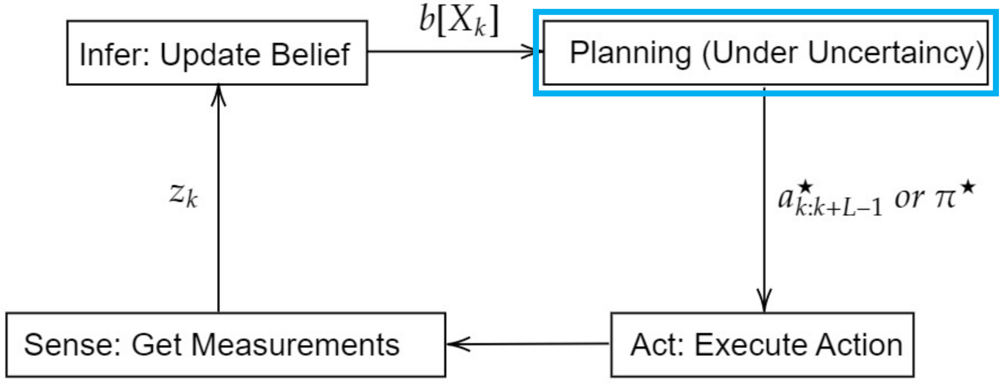
Outline

- Background
- Related Work
- Method
- Evaluation
- Conclusion

Background

Background

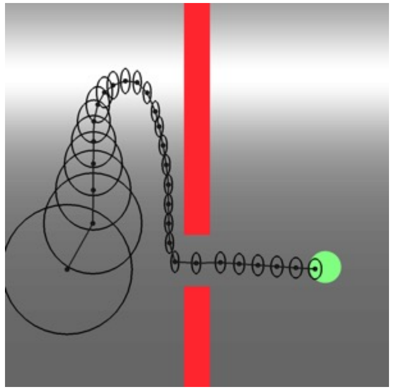
- **P**artially **O**bservable **M**arkov **D**ecision **P**rocess (POMDP)
  Commonly formulated as a tuple $\langle X, A, Z, T, O, R, \gamma \rangle$
  - X - state space
  - A - action space
  - Z - observation space
  - T - probabilistic transition model
  - O - probabilistic observation model
  - R - reward model
  - γ - discount factor



- Autonomous platform acting under uncertainty

Background

- **B**elief **S**pace **P**lanning (BSP)
  - Instead of planning over the state space, plan in the probabilistic space over the state (denoted as *belief*)
  - $b[x_k] = \mathbb{P}(x_k \mid a_{1:k-1}, z_{1:k}, b_0)$
  - Allows the use of *Information Theoretic* rewards (e.g.):
    - ➢ Differential Entropy
    - ➢ Mutual information
    - ➢ Information Gain
  - Can be very useful



Background

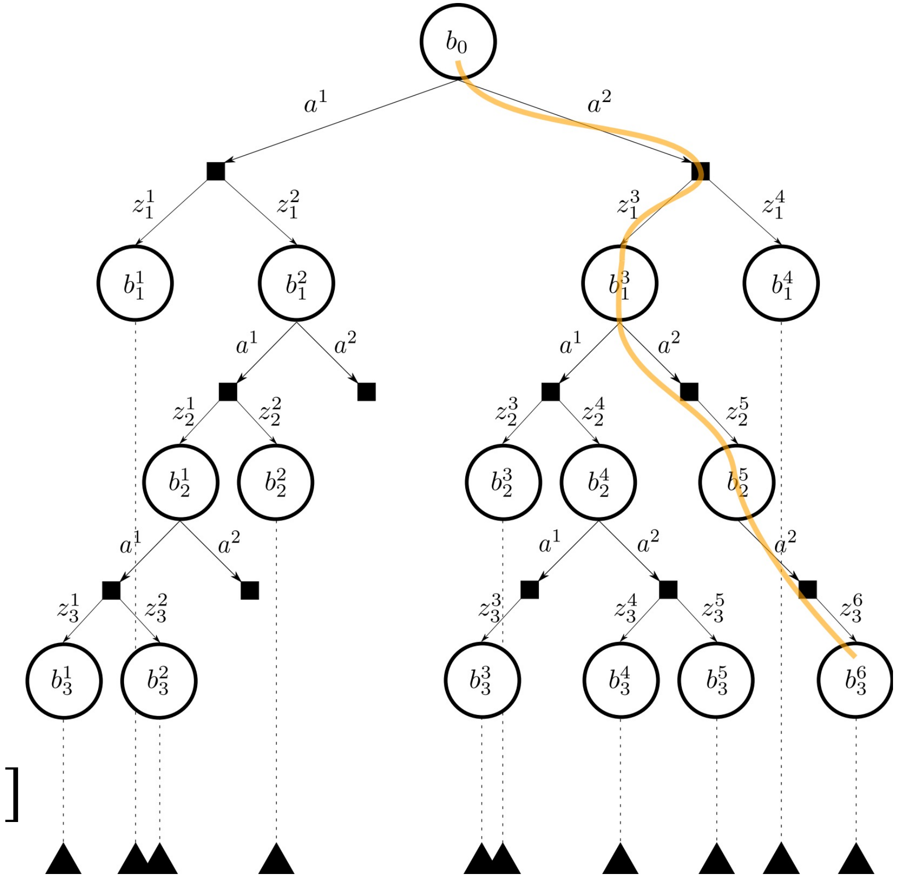
- Online Planning
  - Multiple steps ahead in time
  - Multiple realizations of action-observation sequences:
    $\{(a_0, z_1), (a_1, z_2), (a_2, z_3), \dots, (a_{L-1}, z_L)\}$
  - Commonly done by building a *Belief Tree*
    - tree root is the current time belief
    - Requires a "black box" simulator ***or*** motion and observation models access
    - Tree size limited by predefined params such as time/depth/number of nodes

Background

- Online Planning – the Belief tree
  - Each node induces a reward: $r(b,a) \in \mathbb{R}$
  - Planning goal:

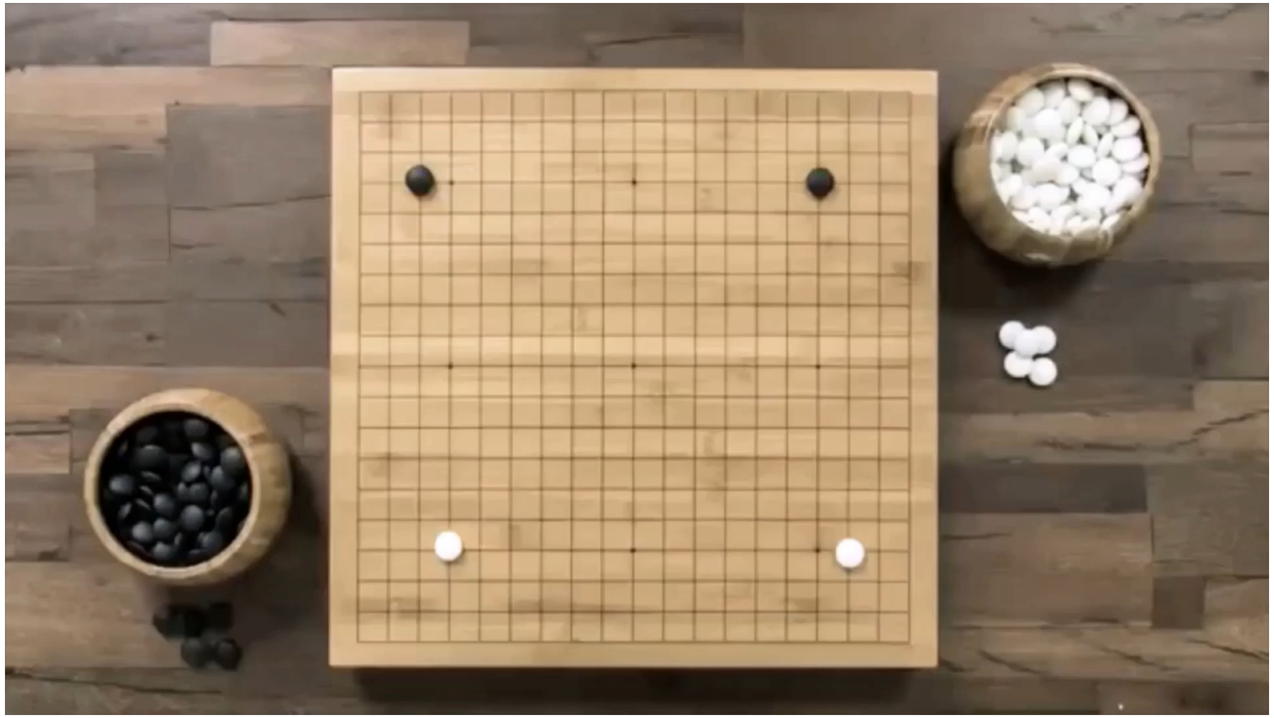
    Find the actions sequence that induces highest cumulative reward
  - More formally…
    - ➢ Find optimal *Policy* $\pi: b \rightarrow a$
    - ➢ Maximizing the *Value Function*

$$V^{\pi}(b_k) = \mathbb{E}_{z_{k+1}}[r(b_k,a) + V^{\pi}(b_{k+1})]$$



Background

- Online Planning – the Belief tree
  - Challenges?
    - ➢ Curse of History
    - ➢ Curse of Dimensionality
    - ➢ Continuous Domains
    - ➢ Non-parametric beliefs
    - ➢ Information Theoretic
    - ➢ High dimension state space



Video source: Google DeepMind, David Silver

Background

- Non-parametric distributions
  - A more general setting
  - Typically, approximations resort to sampling
  - A well studied problem in Statistics, Information theory, Machine learning etc.
  - Commonly in planning:
    - ➢ State samples
    - ➢ Observation samples

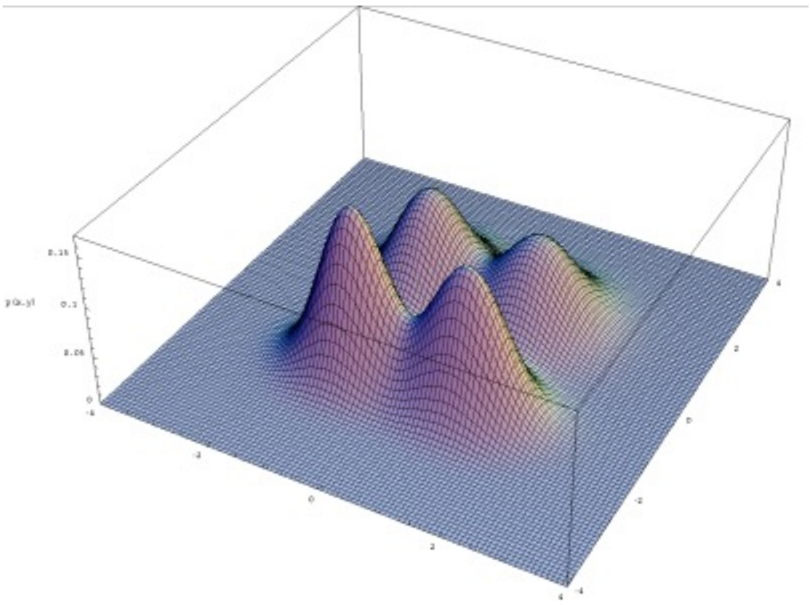
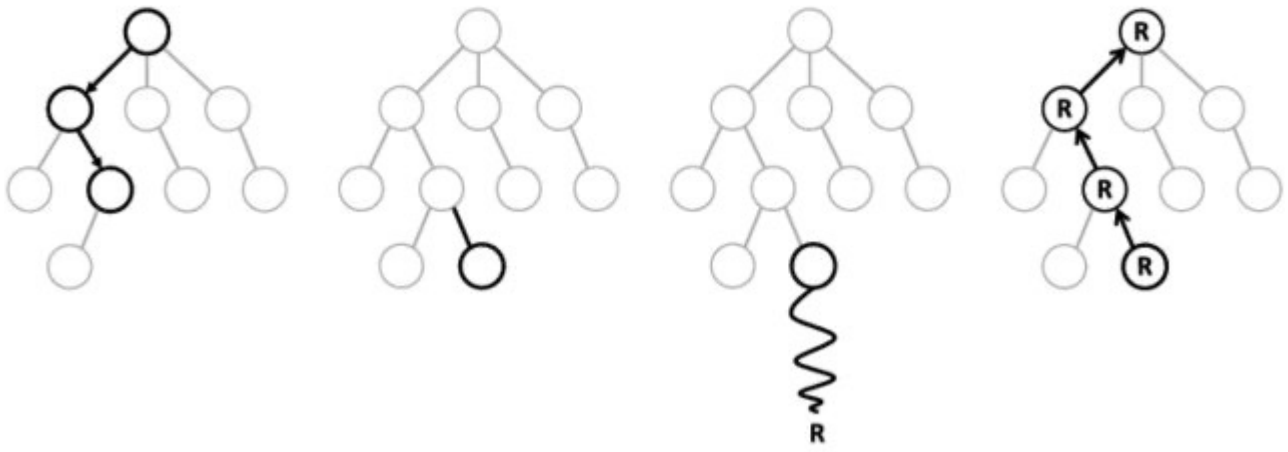


Image [source]

Background

- **M**onte **C**arlo **T**ree **S**earch (MCTS)
  - Breaks the curse of history by “revealing” only parts of the full tree.
  - Breaks the curse of dimensionality by using a predefined number of state samples

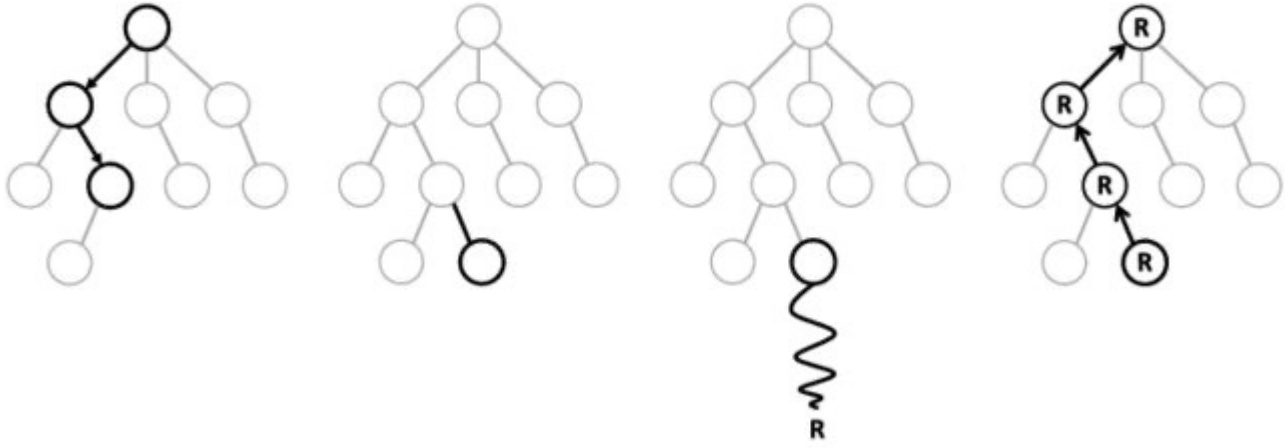


(a) Selection (b) Expansion (c) Simulation (d) Backpropagation

Image source: ‘An Analysis of Monte Carlo Tree Search’ by James et al.

Background

- **M**onte **C**arlo **T**ree **S**earch (MCTS)
  - Additional details:
    - ➢ Builds the tree incrementally using a predefined time/iterations budget
    - ➢ Requires some heuristics for exploration strategy and rollout policy, e.g., UCB



(a) Selection (b) Expansion (c) Simulation (d) Backpropagation

Image source: '[An Analysis of Monte Carlo Tree Search]' by James et al.

Related Work

Related Work

- Recall our considered setting:
  - *Online* POMDP planning
  - *Continuous* state space
  - *Continuous* observation space
  - *Information theoretic* rewards (reward over the belief)

Related Work

- Online POMDP Planners
  - POMCP (2010 Silver et al.)
    - POMCPOW (2017 Sunberg et al.)
    - PFT-DPW (2017 Sunberg et al.)
    - IPFT (2020 Fischer et al.)
    - ρ-POMCP (2021 Thomas et al.)
  - DESPOT (2017 Ye et al.)
    - DESPOT-α (2019 Garg et al.)

Related Work

- Online POMDP Planners Comparison

Algorithm	Continuous state space	Continuous observation space	Rewards over the belief	Use Particle Filter
POMCP	✓	✗	✗	✗
POMCPOW	✓	✓	✗	✗
PFT-DPW	✓	✓	✓	✓
IPFT	✓	✓	✓	✓
ρ-POMCP	✓	✗	✓	✓
DESPOT	✓	✗	✗	✗
DESPOT-α	✓	✓	✗	✓

- Many other solvers exist, but aren’t designed to continuous state space and/or Online setting: PBVI, HSVI, HSVI2, SARSOP, ABT, SARISA, ρ-POMDP, LC-HSVI etc.

Contribution

- Novel simplification for our POMDP setting
- Novel simplification based differential entropy approximation bounds
- Embedding into a Sparse-Sampling planning scheme
- Embedding into a state-of-the-art MCTS planning scheme
- Theoretical guarantees for:
  - Tree-Consistency
  - Solution consistency
  - Time complexity analysis

Method

Method - Preliminaries

- Simplification
  - Solving a POMDP accurately is not tractable
  - Many approximation methods take place
  - Simplification deals with relaxation of the decision-making problem (e.g.)
    - ➢ Simplified decision making in the belief space using belief sparsification by K. Elimelech and V. Indelman IJRR 2021 accepted
    - ➢ Ft-bsp: Focused topological belief space planning by M. Shienman, A. Kitanov, and V. Indelman RA-L 2021
  - Ideally provides the same solution
  - If not possible, the potential objective error is bounded

Method - Preliminaries

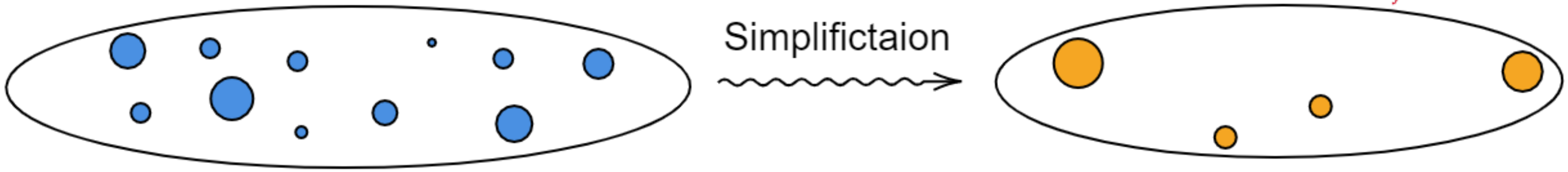
- Differential entropy approximation
  - The belief is approximated as a set of particles
  - Approximation can be achieved via Kernel Density Estimation or a method by Boers et al.

Method

- Chosen Simplification
  - Belief node simplification – use a sub-set of particles
  - Instead of expensive belief dependent reward calculation,
    calculate simplification-based reward bounds
  - Reward bounds can be generalized to
    Value function/Action-Value function bounds
  - We consider differential entropy approximation by Boers as a reward function

$$b = \left\{x^i,\ w^i\right\}_{i=1}^{N}$$

$$b^s = \left\{x^j,\ w^j\right\}_{j=1}^{N^s}$$



Novel Differential Entropy Bounds

Method

- Novel Simplification based bounds
  - Differential entropy: $\mathcal{H}(X) = -\int_x b(x) \cdot \log\big(b(x)\big)\, dx$
  - Boers original approximation:

$$\hat{\mathcal{H}}(b_{k+1}) \triangleq \log\left[\sum_i \mathbb{P}(z_{k+1} \mid x_{k+1}^i) w_k^i\right] - \sum_i w_{k+1}^i \cdot \log\left[\mathbb{P}(z_{k+1} \mid x_{k+1}^i) \sum_j \mathbb{P}(x_{k+1}^i \mid x_k^j, a_k) w_k^j\right]$$

  - Our novel bounds (over: $-\widehat{\mathcal{H}}$):

$$u \triangleq -\log\left[\sum_i \mathbb{P}\ (z_{k+1}|x_{k+1}^i) w_k^i\right] + \sum_{i \in \neg A_{k+1}^s} w_{k+1}^i \cdot \log\left[\text{const} \cdot \mathbb{P}\ (z_{k+1}|x_{k+1}^i)\right]$$
$$+ \sum_{i \in A_{k+1}^s} w_{k+1}^i \cdot \log\left[\mathbb{P}\ (z_{k+1}|x_{k+1}^i) \sum_j \mathbb{P}\ (x_{k+1}^i|x_k^j, a_k) w_k^j\right]$$

$$\ell \triangleq -\log\left[\sum_i \mathbb{P}\ (z_{k+1}|x_{k+1}^i) w_k^i\right] + \sum_i w_{k+1}^i \cdot \log\left[\mathbb{P}\ (z_{k+1}|x_{k+1}^i) \sum_{j \in A_k^s} \mathbb{P}\ (x_{k+1}^i|x_k^j, a_k) w_k^j\right]$$

Method

- Novel Simplification based bounds
  - Our novel bounds:
  - Where:
    - $\mathbb{P}(z \mid x)$ observation model
    - $\mathbb{P}(x' \mid x, a)$ motion model
    - w^i weight of state sample x^i
    - A^s set of simplified state indexes
    - $\neg A^s$ compliment of A^s
    - const is $\max_{x'} \mathbb{P}(x' \mid x, a)$

$$u \triangleq -\log\left[\sum_i \mathbb{P}\ (z_{k+1}|x^i_{k+1})w^i_k\right] + \sum_{i\in\neg A^s_{k+1}} w^i_{k+1}\cdot\log\left[\mathrm{const}\cdot\mathbb{P}\ (z_{k+1}|x^i_{k+1})\right]$$

$$+\sum_{i\in A^s_{k+1}} w^i_{k+1}\cdot\log\left[\mathbb{P}\ (z_{k+1}|x^i_{k+1})\sum_j \mathbb{P}\ (x^i_{k+1}|x^j_k, a_k)w^j_k\right]$$

$$\ell \triangleq -\log\left[\sum_i \mathbb{P}\ (z_{k+1}|x^i_{k+1})w^i_k\right] + \sum_i w^i_{k+1}\cdot\log\left[\mathbb{P}\ (z_{k+1}|x^i_{k+1})\sum_{j\in A^s_k} \mathbb{P}\ (x^i_{k+1}|x^j_k, a_k)w^j_k\right]$$

Method

- Novel Simplification based bounds
  - Our bounds properties
    - Convergence
    - Monotonically increasing & decreasing
    - On-demand tightening
    - Complexity of $\mathrm{O}(N \cdot N^s)$ instead of $\mathrm{O}(N \cdot N)$
    - User defined simplification levels
    - Calculation reuse
    - No time loss whatsoever

N – number of particles representing original belief b
N^s – number of particles representing *simplified belief* b^s

Method

- Extending the bounds to objective bounds
  - Objective function:

$$J(b_k, \pi_{k+}) = r(b_k, a_k) + \mathop{\mathbb{E}}_{z_{k+1}} \{J(b_{k+1}, \pi_{(k+1)+})\}$$

Where: $\pi_{k+} \triangleq \pi_{k:k+L}$

  - Planning:

$$J(b_k, \pi^\star_{k+}) = \max_{\pi_k} \{r(b_k, a_k) + \mathop{\mathbb{E}}_{z_{k+1}} \{J(b_{k+1}, \pi^\star_{(k+1)+})\}\}$$

  - Rewards bounds translate to objective bounds:

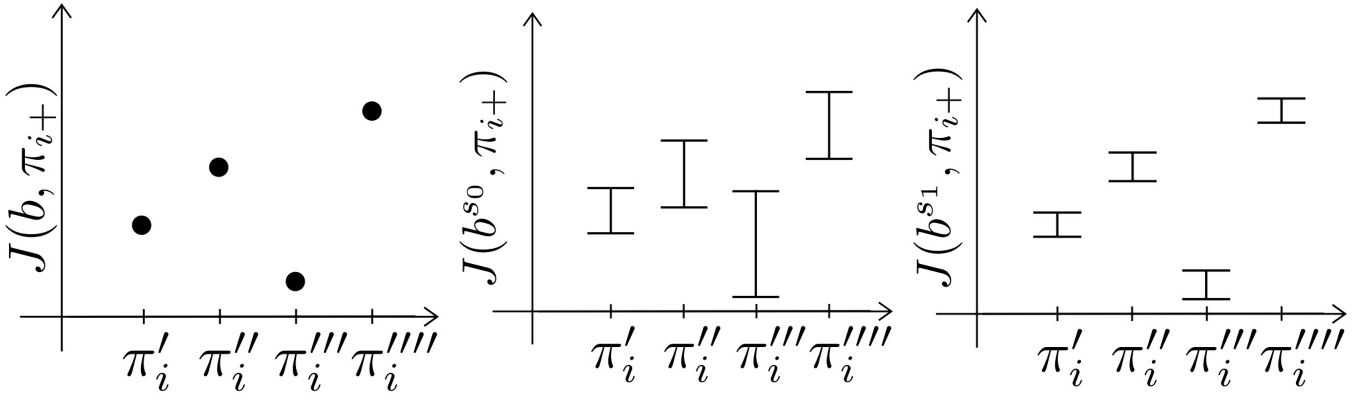
$$\mathbf{lb}(b^s, b, a) \leq r(b, a) \leq \mathbf{ub}(b^s, b, a) \quad \Rightarrow \quad \mathcal{UB}(b_i, \pi_{i+}) = \mathbf{ub}(b^s_i, b_i, a)) + \mathop{\mathbb{E}}_{z_{i+1}} \{\mathcal{UB}(b_{i+1}, \pi_{(i+1)+})\}$$

$$\mathcal{LB}(b_i, \pi_{i+}) = \mathbf{lb}(b^s_i, b_i, a) + \mathop{\mathbb{E}}_{z_{i+1}} \{\mathcal{LB}(b_{i+1}, \pi_{(i+1)+})\},$$

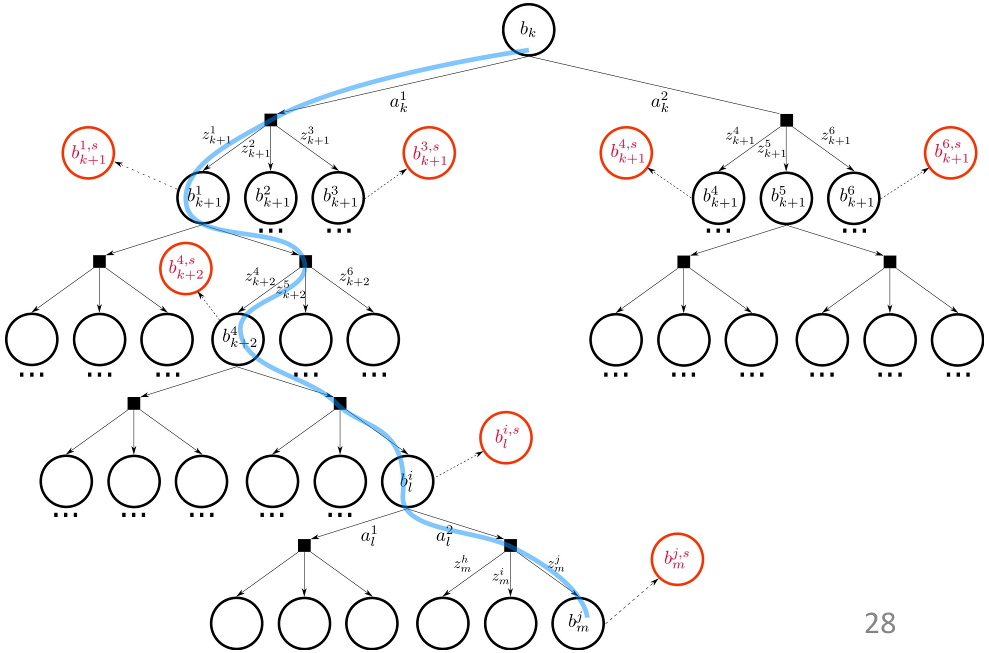
Simplified Information Theoretic BSP (SITH-BSP)

Method

- Planning using objective bounds
  - Analytical bounds along the tree
  - We can prune sub optimal branches traversing up the tree if the objective bounds do not overlap

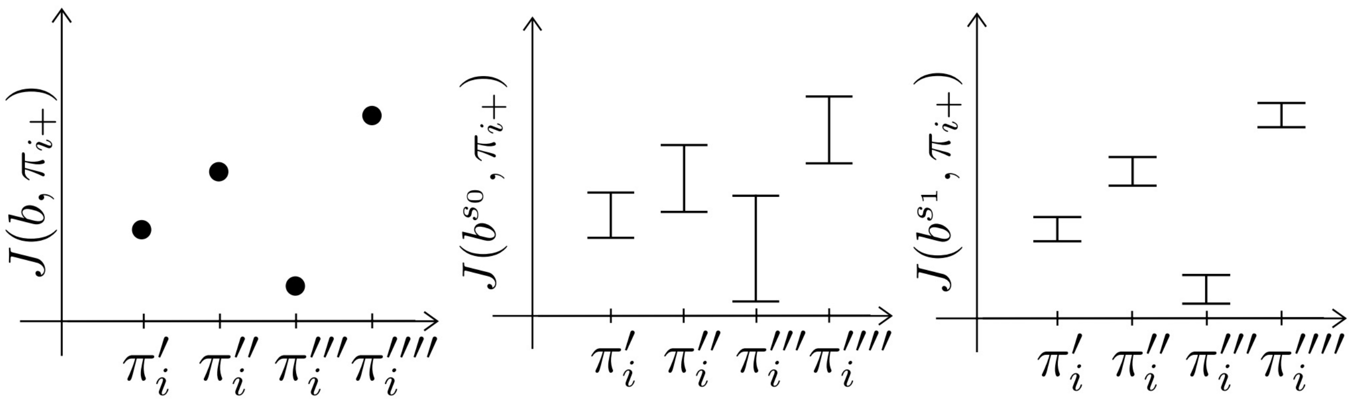


(a) **(b)** **(c)**

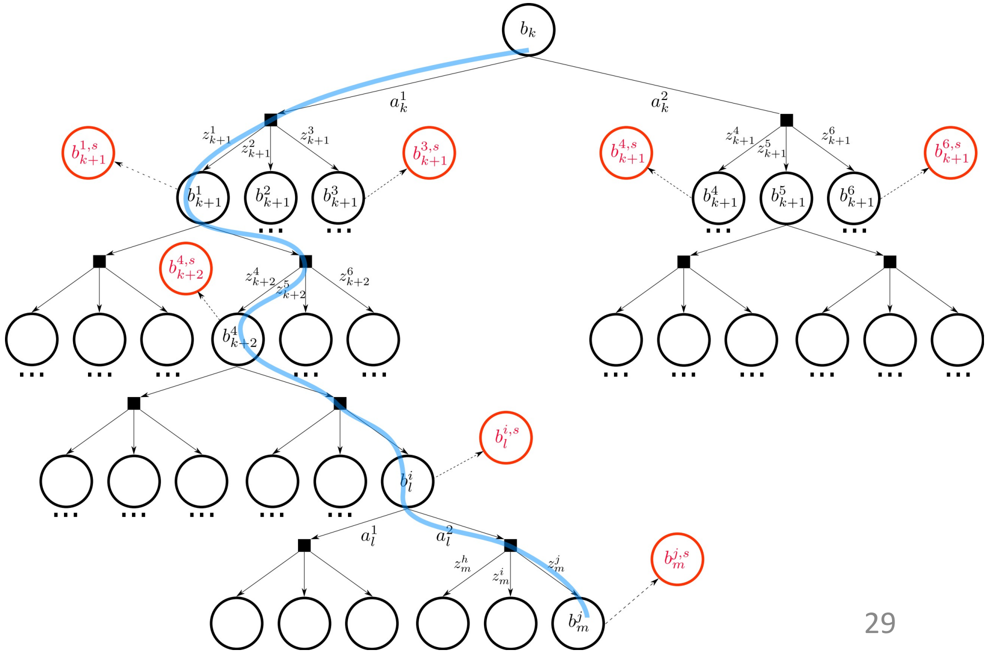


Method

- Planning using objective bounds
  - Overlapping bounds?
    - Increment the simplification level, in our case - take more particles to represent the simplified belief.
    - This is done with calculation re-use

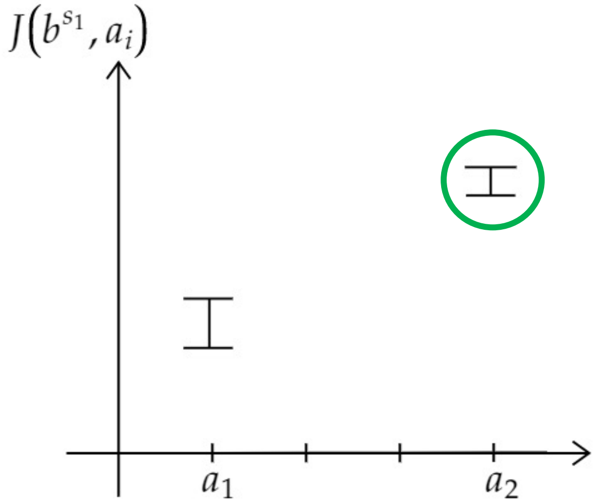


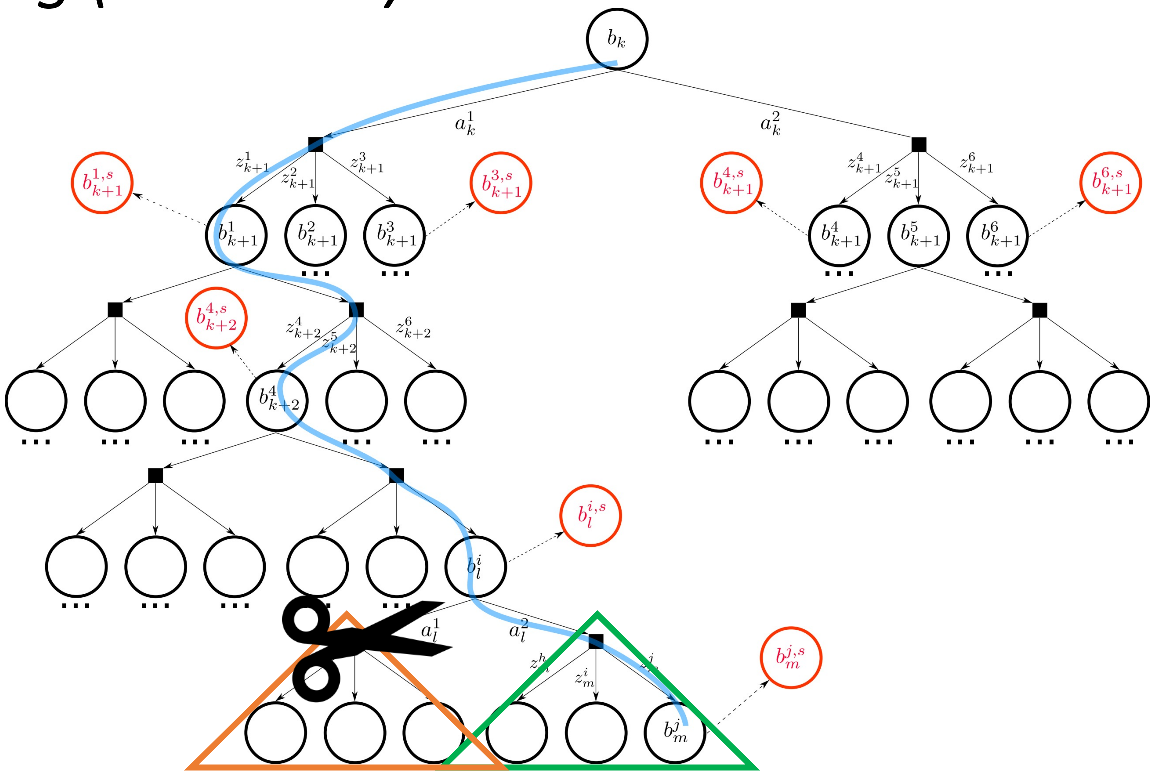
(a) (b) (c)



Method

- Full algorithmic scheme: *Simplified Information Theoretic Belief Space Planning (SITH-BSP)*





Method

- Full algorithmic scheme: *Simplified Information Theoretic Belief Space Planning (SITH-BSP)*

Algorithm 1 Prune Branches

procedure PRUNE
  Input: (belief-tree root, b; bounds of root's children, $\{\mathcal{LB}^m, \mathcal{UB}^m\}_{m=1}^{C}$) going out of b.
  $\mathcal{LB}^\star \leftarrow \max_m \{\mathcal{LB}^m\}_{m=1}^{C}$
  for all children of b **do**
    if $\mathcal{LB}^\star > \mathcal{UB}^m$ **then**
      prune child m from the belief tree
    end if
  end for
end procedure

Algorithm 2 Simplified Information Theoretic Belief Space Planning (SITH-BSP)

procedure FIND OPTIMAL POLICY(belief-tree: $\mathbb{T}$)
  $s \leftarrow s_0$
  return ADAPT SIMPLIFICATION($\mathbb{T}$,s)
end procedure
procedure ADAPT SIMPLIFICATION(belief-tree: $\mathbb{T}$, s_i)
  if $\mathbb{T}$ is a leaf **then**
    return $\{\text{lb}, \text{ub}\}$
  end if
  Set simplification level: $s \leftarrow s_i$
  for all subtrees $\mathbb{T}'$ in $\mathbb{T}$ **do**
    ADAPT SIMPLIFICATION($\mathbb{T}'$,s)
    Calculate $\mathcal{LB}^{s^j}$,$\mathcal{UB}^{s^j}$ according to s and (11)
  end for
  Using $\{\mathcal{LB}^{s^j}, \mathcal{UB}^{s^j}\}_{j=1}^{|\mathcal{A}|}$ and Alg. 1 prune branches
  while not all $\mathbb{T}'$ but 1 in $\mathbb{T}$ pruned **do**
    Increase simplification level: $s \leftarrow s + 1$
    ADAPT SIMPLIFICATION($\mathbb{T}$,s)
  end while
  Update $\{\mathcal{LB}^{s^j\star}, \mathcal{UB}^{s^j\star}\}$ according to (14)
  return optimal action branch that left $a^\star$ and $\{\mathcal{LB}^{s^j\star}, \mathcal{UB}^{s^j\star}\}$.
end procedure

- Submitted to ICRA/RA-L 2022

Online POMDP Planning via Simplification

Method

- Restricting assumption?
  - The belief tree is given
  - State-of-the-Art methods build the tree incrementally

Simplified Information Theoretic Particle Filter Tree (SITH-PFT)

Method

- Following work
  - We incorporate the bounds into a state-of-the-art POMDP planner
  - Not straightforward
  - The goal was to show speed up compared to the baseline
- Chosen baseline
  - PFT-DPW (Sunberg et al. 2017)
  - Chosen because it is the least restricting.
  - Uses Particle Filter with Double Progressive Widening over a MCTS framework

Method

- MCTS Adaptation
  - Main Challenge: Build the same tree as PFT-DPW without calculating the rewards (only the bounds)
  - Baseline tree build is guided by UCB1:

$$UCB1(ha) = Q(ha) + c \cdot \sqrt{\frac{\log(N(h))}{N(ha)}}$$

Where:

- ➢ h, a are history (belief representation) and action respectively
- ➢ $Q(ha)$ belief action value function (known as Q function)
- ➢ c exploration constant
- ➢ $N(\cdot)$ belief/belief-action node visitation counter

Method

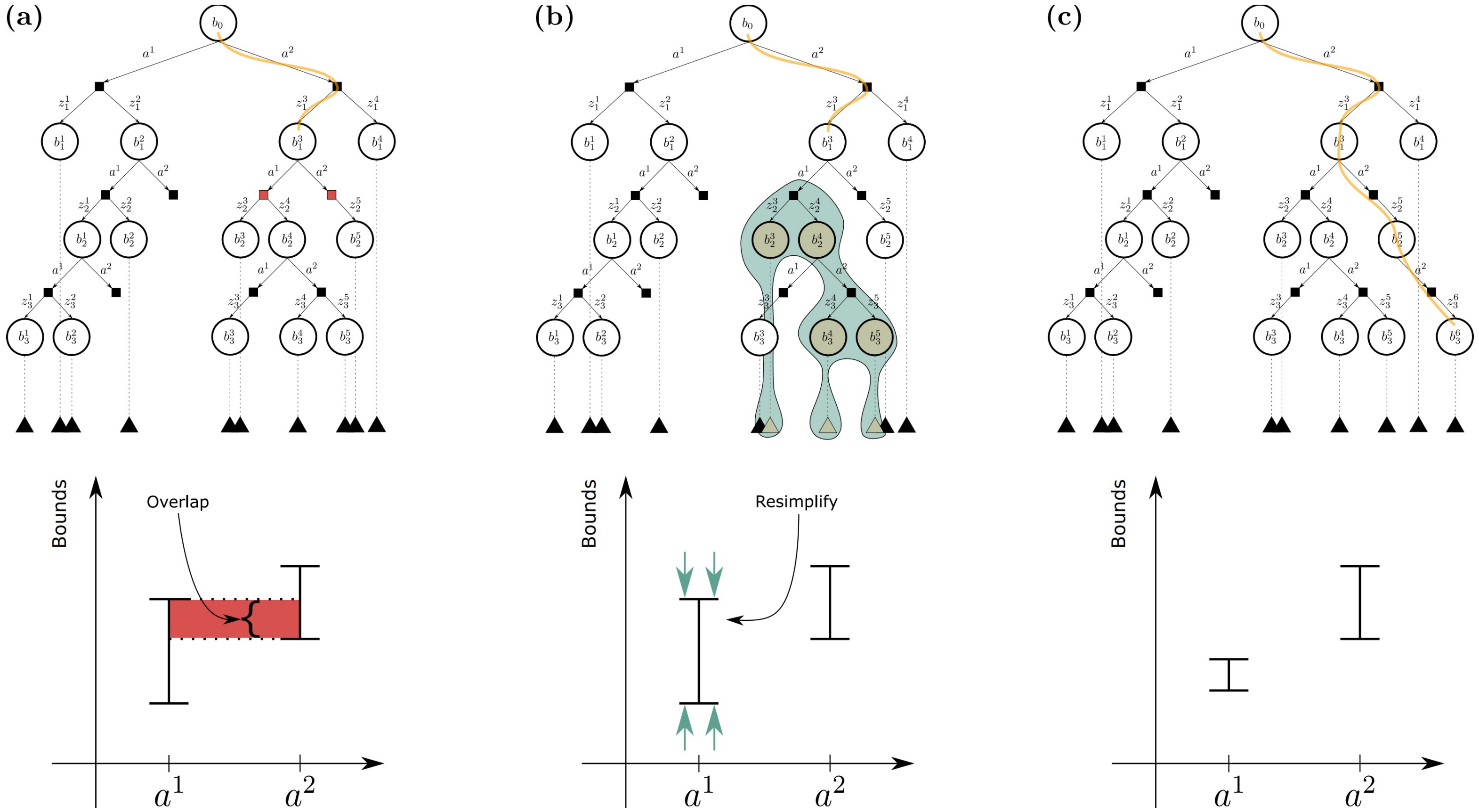
- MCTS Adaptation
  - Main Challenge: Build the same tree as PFT-DPW without calculating the rewards (only the bounds)
  - Solution: We use the bounds to lower and upper bound the UCB:

$$\underline{\mathrm{UCB}}(ha) \triangleq Q^{x}(ha) + \lambda\mathcal{LB}(ha) + c \cdot \sqrt{\frac{\log(N(h))}{N(ha)}}$$

$$\overline{\mathrm{UCB}}(ha) \triangleq Q^{x}(ha) + \lambda\mathcal{UB}(ha) + c \cdot \sqrt{\frac{\log(N(h))}{N(ha)}}$$

Method

- Algorithmic Overview



Light green section is determined by following a specific “Resimplification Strategy”

Method

- Theorems:
  - **Theorem 1.** *The SITH-PFT and PFT-DPW are Tree Consistent Algorithms*
  - **Theorem 2.** *The SITH-PFT provides the same solution as PFT-DPW*
  - **Theorem 3.** *The specific resimplification strategy is a converging and finite-time resimplification strategy*

Method

- Full proofs along with time complexity analysis can be found in the original paper:
  - '[Simplified Belief-Dependent Reward MCTS Planning with Guaranteed Tree Consistency]' by O. Sztyglic*, A. Zhitnikov*, V. Indelman 2021 (submitted to NeurIPS 2021)

Algorithm 1 SITH-PFT

procedure PLAN(belief: b)
  $h \leftarrow \emptyset$
  for $i \in 1:n$ **do**
    SIMULATE($b, d_{\max}, h$)
  end for
  return ACTION SELECTION(b, h
nullified exploration constant c
end procedure
procedure SIMULATE(belief: b, dept
  if $d = 0$ **then**
    return 0
  end if
  $a \leftarrow$ ACTION SELECTION(b, h)
  if $|C(ha)| \leq k_o N(ha)^{\alpha_o}$ **then**
    $o \leftarrow$ sample x from b, genera
    $b', r^x \leftarrow G_{\mathrm{PF}(m)}(bao)$
    Calculate initial u', ℓ' for b' b
minimal simp. level
    $C(ha) \leftarrow C(ha) \cup \{(r^x, \ell', u$
    $R, L, U \leftarrow r^x, \ell', u' + \gamma$ ROL
  else
    $(r^x, \ell', u', b', o) \leftarrow$ sample un
    $R, L, U \leftarrow r^x, \ell', u' + \gamma$ SIM
  end if
  if deepest resimplification depth <
for updated deeper in the tree bounds
    reconstruct $\mathcal{LB}(ha), \mathcal{UB}(ha)$
  end if
  $N(h) \leftarrow N(h) + 1$
  $N(ha) \leftarrow N(ha) + 1$
  $Q^x(ha) \leftarrow Q^x(ha) + \frac{R - Q^x(ha)}{N(ha)}$
  $\mathcal{LB}(ha) \leftarrow \mathcal{LB}(ha) + \frac{L - \mathcal{LB}(ha)}{N(ha)}$
  $\mathcal{UB}(ha) \leftarrow \mathcal{UB}(ha) + \frac{U - \mathcal{UB}(ha)}{N(ha)}$
  return R, L, U
end procedure

Algorithm 2 Action Selection

procedure ACTION SELECTION(b, h)
  while true **do**
    Status, $a \leftarrow$ SELECT BEST(b,
    if Status **then**
      break
    else
      for all $b', o \in C(ha)$ **do**
        RESIMPLIFY(b', hao)
      end for
      reconstruct $\mathcal{LB}(ha), \mathcal{UB}($
    end if
  end while
  return a
end procedure
procedure SELECT BEST(b, h)
  Status $\leftarrow$ true
  $\tilde{a} \leftarrow \arg\max_a \{\underline{\mathrm{UCB}}(ha)\}$
  gap $\leftarrow 0$
  child-to-resimplify $\leftarrow \tilde{a}$
  for all ha children of b **do**
    if $\underline{\mathrm{UCB}}(h\tilde{a}) < \overline{\mathrm{UCB}}(ha) \wedge a$
      Status $\leftarrow$ false
      if $\mathcal{UB}(ha) - \mathcal{LB}(ha) >$ g
        gap $\leftarrow \mathcal{UB}(ha) - \mathcal{LB}$
        child-to-resimplify $\leftarrow$
      end if
    end if
  end for
  return Status, child-to-resimplify
end procedure

Algorithm 3 Resimplification

procedure RESIMPLIFY(b, h)
  if b is a leaf **then**
    REFINE$_{\{\ell,u\}}(b)$
    RESIMPLIFY ROLLOUT(b, h)
    return
  end if
  $\tilde{a} \leftarrow \arg\max_a \{N(ha) \cdot (\mathcal{UB}(ha) - \mathcal{LB}(ha))\}$
  for all $b', o \in C(h\tilde{a})$ **do**
    RESIMPLIFY($b', h\tilde{a}o$)
  end for
  reconstruct $\mathcal{LB}(h\tilde{a}), \mathcal{UB}(h\tilde{a})$
  REFINE$_{\{\ell,u\}}(b)$
  RESIMPLIFY ROLLOUT(b, h)
  return
end procedure
procedure RESIMPLIFY ROLLOUT(b, h)
  $b^{\mathrm{rollout}} \leftarrow$ find weakest link in rollout
  REFINE$_{\{\ell,u\}}(b^{\mathrm{rollout}})$
end procedure
procedure REFINE$_{\{\ell,u\}}(b)$
  if (12) holds for b, refine its ℓ, u and promote its simplification level
end procedure

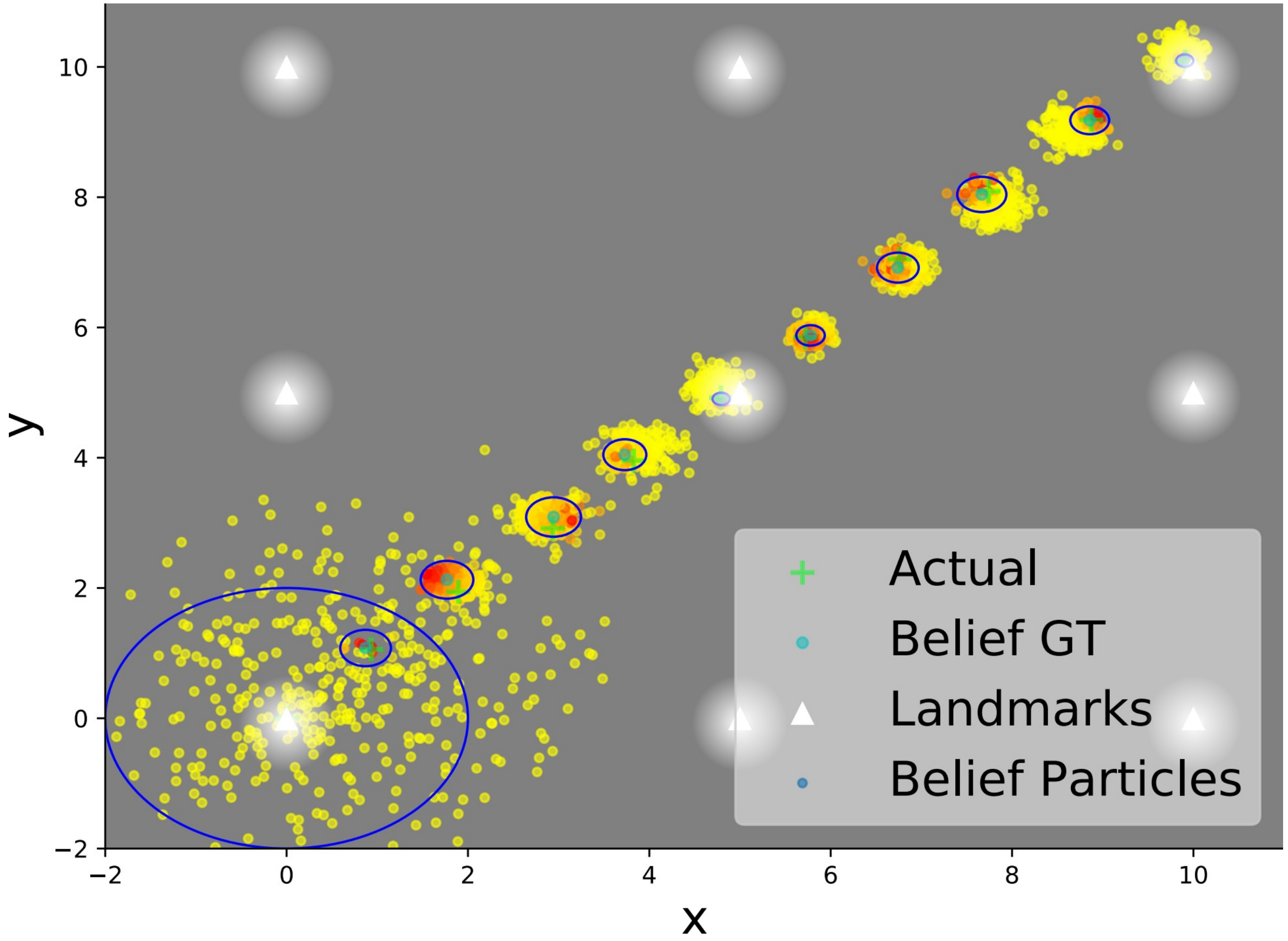
Evaluation

Evaluation – SITH-BSP

- Bounds Convergence study
  - Predefined action sequence
  - True belief is Gaussian so we can access the ground truth differential entropy
  - The agent maintains a belief as a weighted particle set
  - We experiment with changing number of particles
- Scenario setting: Continuous 2D *'Light-Dark' problem*
  - Map is known along with motion and observation models
  - *Belief* is over the agent 2D location
  - Near scattered '*Light-Beacons*' the uncertainty is reduced

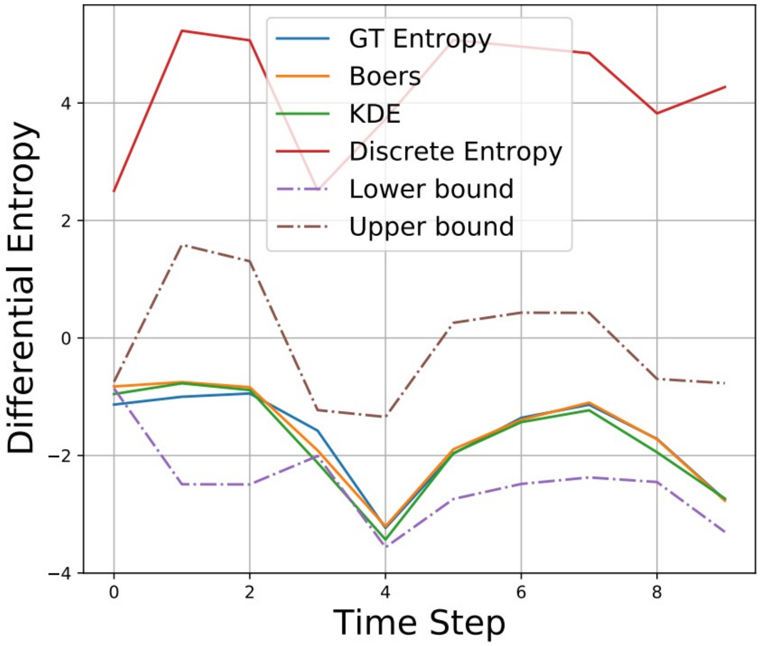
Evaluation – SITH-BSP

- Scenario:

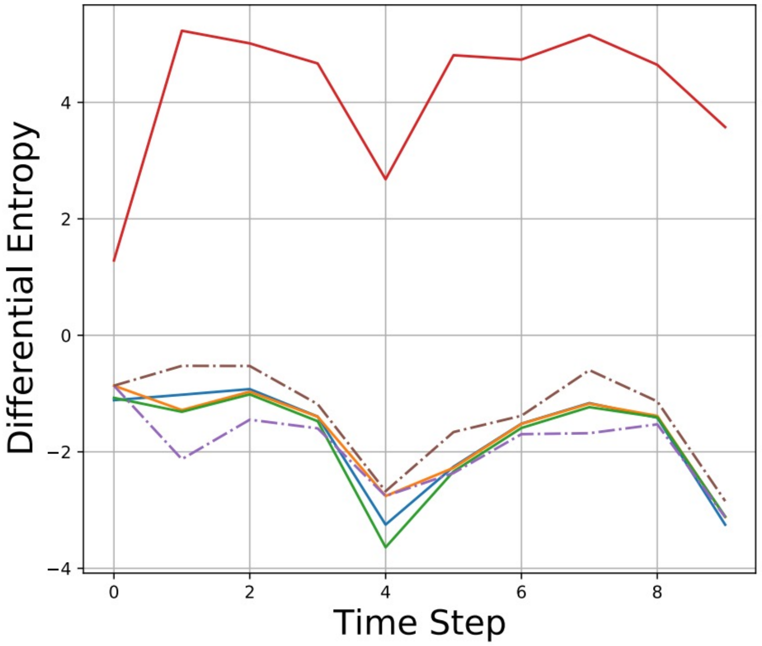


Evaluation – SITH-BSP

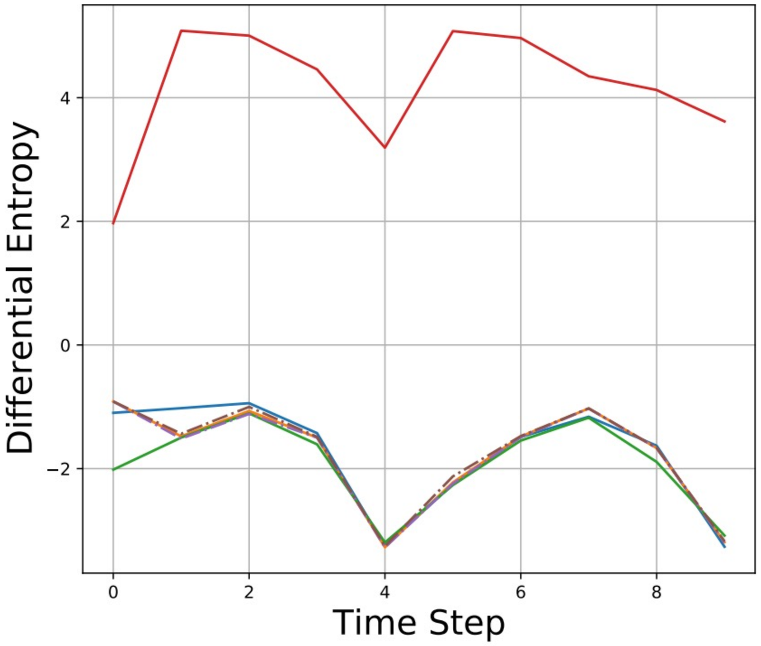
- Bounds Comparison (200 particles):



Simplification level: 0.1



Simplification level: 0.5



Simplification level: 0.9

Evaluation – SITH-BSP

- Bounds Comparison:

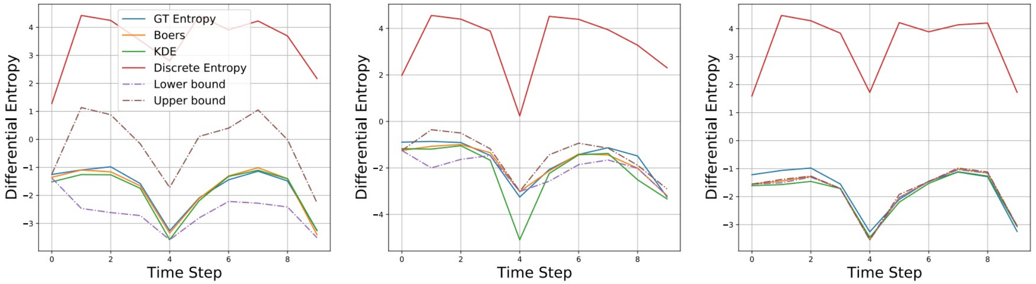


Fig. 1: Differential Entropy Approximations ans Bounds. Calculations were done using 100 particles. From left to right: Simplification is $N^s = \{0.1, 0.5, 0.9\} \cdot N$

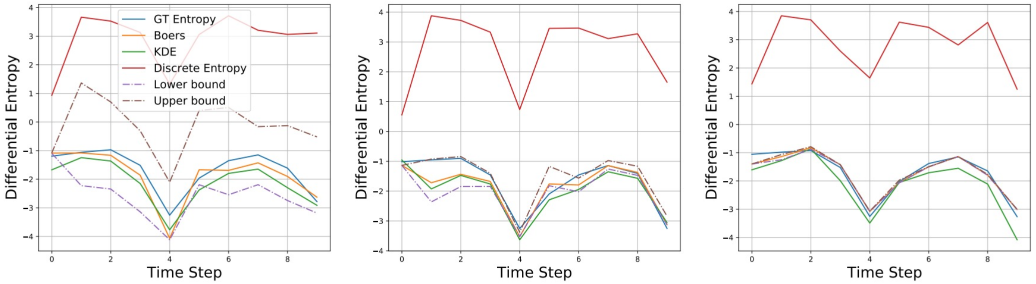


Fig. 2: Differential Entropy Approximations ans Bounds. Calculations were done using 50 particles. From left to right: Simplification is $N^s = \{0.1, 0.5, 0.9\} \cdot N$

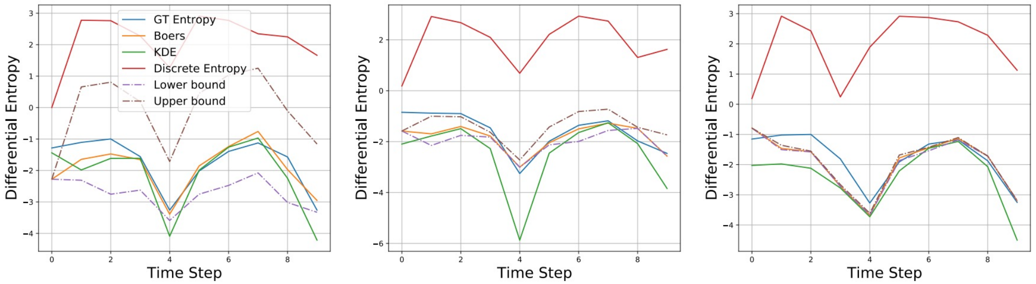


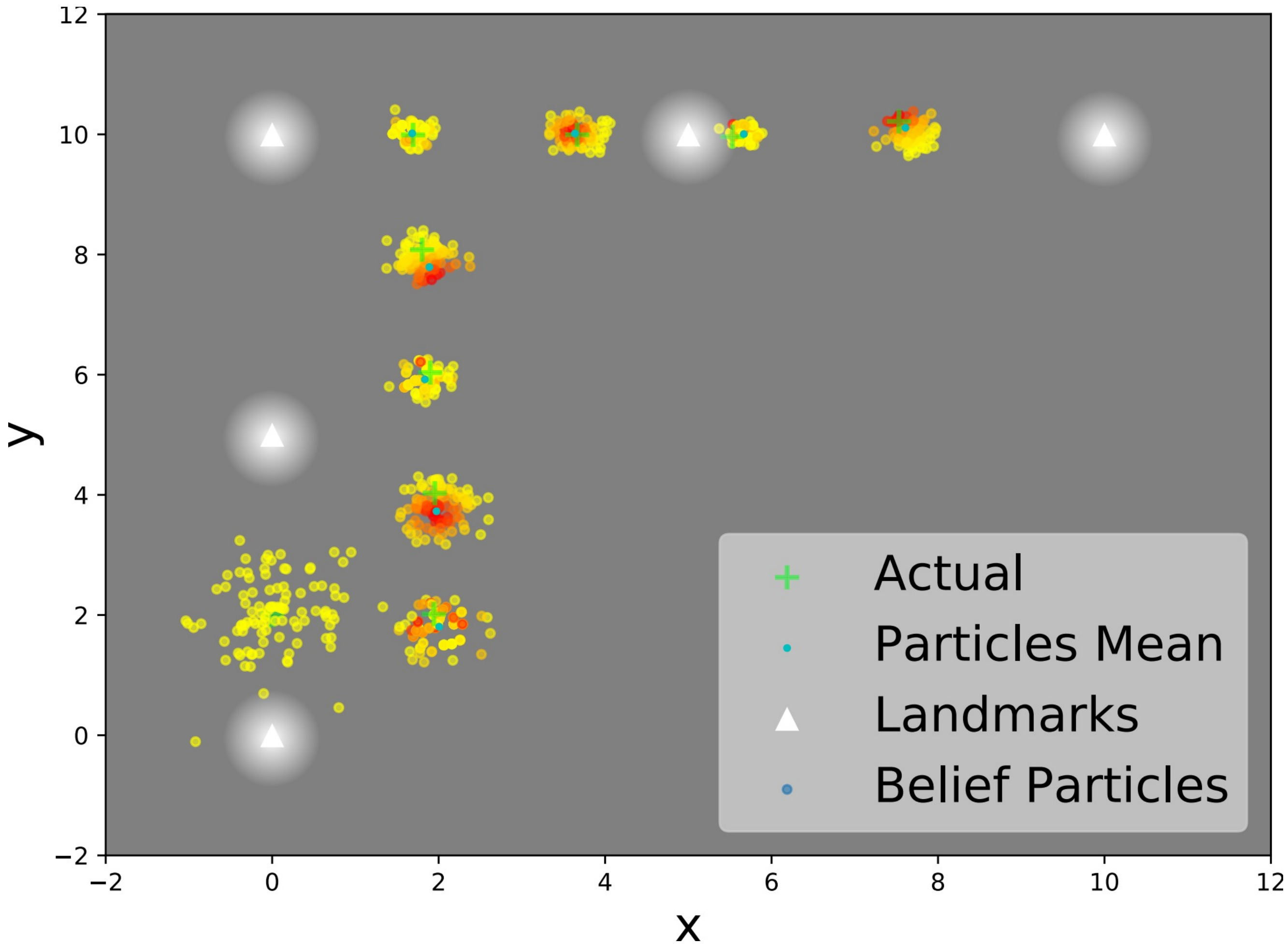
Fig. 3: Differential Entropy Approximations ans Bounds. Calculations were done using 20 particles. From left to right: Simplification is $N^s = \{0.1, 0.5, 0.9\} \cdot N$

Evaluation – SITH-BSP

- Planning baseline: A ‘Sparse-Sampling’ scheme
  - Tree predefined observation branching factor
  - Find optimal action sequence/policy using Bellman updates
  - Different tree structures and a ‘hard’ and an ‘easy’ scenarios
- Scenario setting: Continuous 2D *‘Light-Dark’ problem*
  - Map, motion, and observation models are known
  - *Belief* is over the agent 2D location
  - *‘Light-Beacons’ for uncertainty reduction*
  - Reward model: *‘distance to goal’* & *differential entropy approximation*

Evaluation – SITH-BSP

- Scenario:

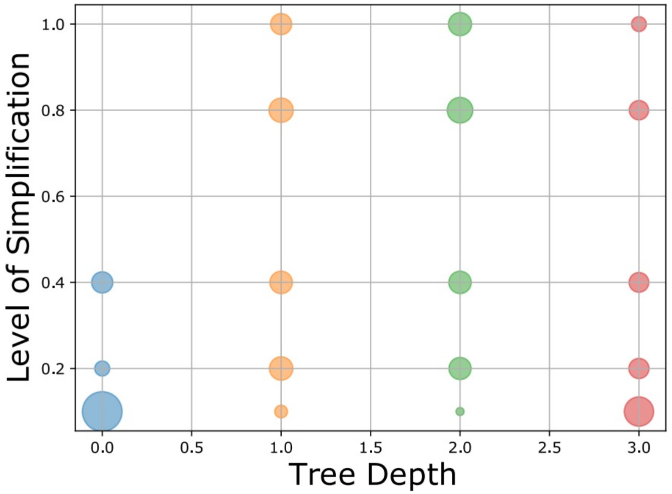


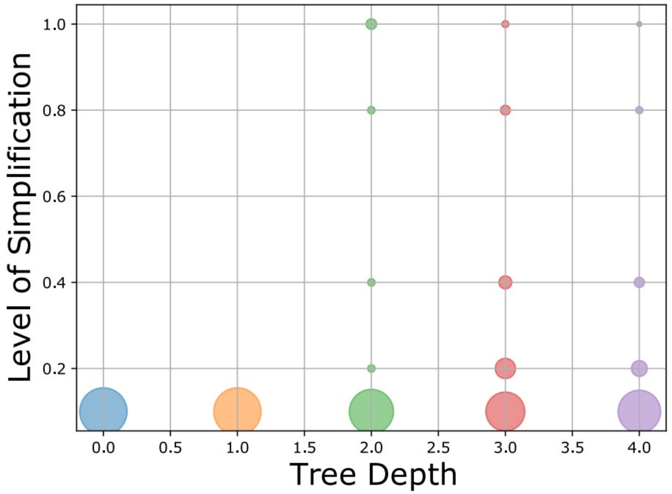
Evaluation – SITH-BSP

- Results (Planning time in seconds):

Simulation	Horizon	[14] Tree			Horizon	[21] Tree			Horizon	[2] Tree		
		20	50	100		10	20	30		20	50	100
Setting I	1	0.124/**0.043**	0.741/**0.192**	2.892/**0.667**	1	0.554/**0.287**	4.065/**1.437**	12.908/**3.953**	5	1.13/**0.776**	6.625/**2.008**	28.19/**7.232**
	2	0.364/**0.129**	2.196/**0.584**	8.616/**2.042**	2	11.02/**5.386**	-	-	10	2.648/**2.555**	15.342/**8.214**	-
	3	0.853/**0.339**	5.059/**1.324**	19.899/**4.658**	3	-	-	-	15	4.2/**3.677**	26.205/**20.174**	-
Setting II	1	0.245/**0.099**	1.513/**0.4**	5.855/**2.018**	1	1.112/**0.953**	8.501/**5.143**	26.375/**11.977**	5	1.383/**0.733**	8.417/**3.864**	33.244/**10.97**
	2	1.209/**0.738**	7.195/**3.821**	30.638/**13.49**	2	-	-	-	10	2.985/**2.112**	17.293/**6.092**	-
	3	5.027/**3.212**	31.515/**18.288**	-	3	-	-	-	15	4.53/**3.701**	27.712/**11.385**	-

- Simplification level:



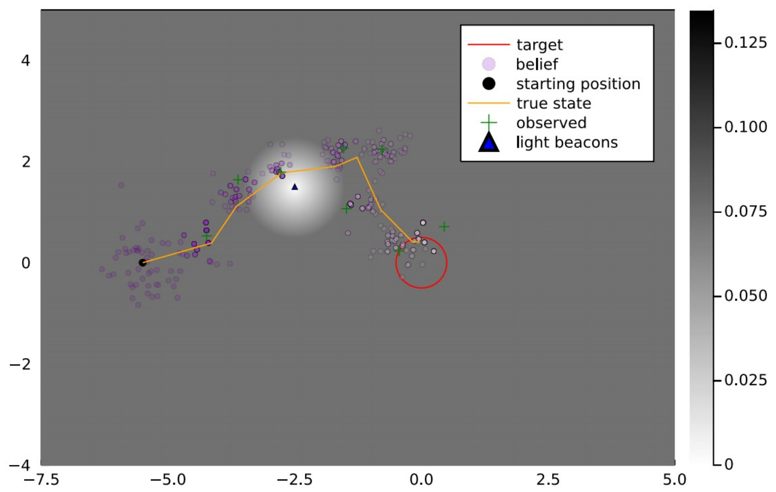


Evaluation – SITH-PFT

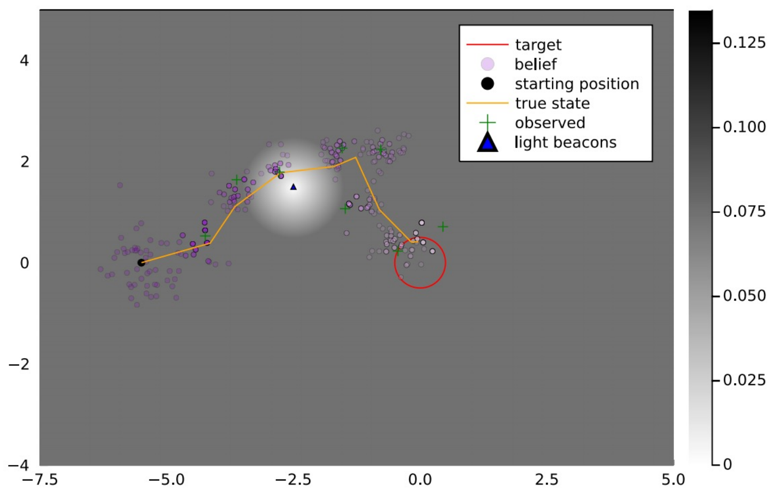
- Planning baseline: PFT-DPW with entropy approximation
    - Some comparison with IPFT that incorporates entropy approximation with PFT-DPW
- Scenario setting: Continuous 2D *'Light-Dark'*
    - Map, motion, and observation models are known
    - *Belief* is over the agent 2D location
    - *'Light-Beacons' for uncertainty reduction*
    - Reward model: *'distance to goal'* & *differential entropy approximation*

Evaluation – SITH-PFT

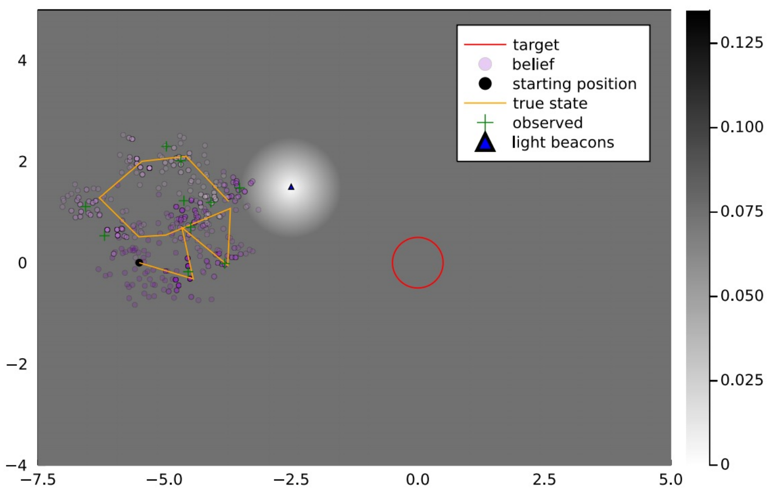
- Scenario:



(a) SITH-PFT



(b) PFT-DPW



(c) IPFT

Evaluation – SITH-PFT

- Time results:

$(m, d, \#\text{iter.})$	Algorithm	planning time [sec]
(50, 30, 200)	PFT-DPW	3.54 ± 0.4
	SITH-PFT	2.96 ± 0.49
(50, 50, 500)	PFT-DPW	9.82 ± 1.31
	SITH-PFT	8.1 ± 1.33
(100, 30, 200)	PFT-DPW	13.42 ± 1.49
	SITH-PFT	10.77 ± 1.73
(100, 50, 500)	PFT-DPW	35.06 ± 4.44
	SITH-PFT	26.7 ± 4.37
(200, 30, 200)	PFT-DPW	55.89 ± 5.41
	SITH-PFT	39.46 ± 7.09
(200, 50, 500)	PFT-DPW	142.14 ± 12.39
	SITH-PFT	100.09 ± 14.67
(400, 30, 200)	PFT-DPW	211.86 ± 24.18
	SITH-PFT	160.36 ± 31.02
(400, 50, 500)	PFT-DPW	570.13 ± 45.48
	SITH-PFT	414.65 ± 53.37
(600, 30, 200)	PFT-DPW	503.78 ± 31.61
	SITH-PFT	374.0 ± 44.23
(600, 50, 500)	PFT-DPW	1204.78 ± 119.16
	SITH-PFT	912.92 ± 116.08

Conclusion

Conclusion

For the setting of POMDP with belief-dependent rewards:

- We introduced novel highly functional bounds over differential entropy approximation based on weighted particles
- Developed a general Sparse-Sampling adaptation to such simplification based converging bounds, leading to substantial speed up.
- Developed a general MCTS adaptation to such simplification based converging bounds, leading to speed up.

Conclusion

- Future possible work:
  - Incorporation of the bounds into other POMDP planning algorithms
  - Incorporation of the bounds into other Domains such as SLAM
  - Given other analytical converging bounds, they can be incorporated into our existing Sparse-Sampling and MCTS adaptations
  - Usage of the bounds (or some linear variant of them) as an exploration heuristics for rollout estimators required by MCTS algorithms

Thank you for your time, any questions?