Multi-robot active perception: 
Dec-MCTS, objective functions, 
and applications

Graeme Best

Advisors: Robert Fitch, Geoffrey Hollinger

COLLABORATIVE ROBOTICS AND INTELLIGENT SYSTEMS (CORIS) INSTITUTE
Perception tasks

Fruit tree modelling
[ACFR, 2014]

Ocean monitoring
[ACFR, 2015]

Subterranean mapping
[CMU, OSU, NEA, 2019]
Active perception

- Typically, robots execute a pre-determined path

- Instead, we should plan the paths online while considering the current knowledge and mission objectives to:
  - Improve the quality of data collected
  - Improve the performance of perception algorithms
Planning for active perception

- Typical active object classification system [Patten, 2016]

- Key challenges for path planning:
  - Plan with respect to suitable objective functions
  - Multiple robots
  - Large environments; long planning horizons
  - Online, onboard decision making
Related work

Informative path planning

Active perception

Multi-robot coordination

AI planning algorithms

[Hollinger, 2014; van Hoof, 2014; Silver 2010; ACFR]
Outline

1. Dec-MCTS

2. Objective functions → applications
Decentralised planning setting

➢ Each robot plans its own actions
➢ Communicate to coordinate plans
➢ Intermittent communication
Multi-robot active perception

➢ Find:
  • The paths for a team of robots
  • That maximises an objective function

➢ Robot $i$ plans its own action sequence while considering:
  • Budget for action costs
  • Global objective function
  • Belief of the other robots’ plans

\[ x := \{ x^1, x^2, ..., x^R \} \]
\[ g(x) \]

\[ x^i = (x_1^i, x_2^i, ...) \]
\[ \sum_{x^j \in x^i} c^i_j \leq B^i \]
\[ g(x) \]
\[ x^{(i)} := x \setminus x^i \]
Dec-MCTS overview

(a) Grow search tree for own actions

(b) Decentralised optimisation of probability distributions

(c) Communicate distributions with other robots

Performed asynchronously by each robot
Dec-MCTS overview

(a) Grow search tree for own actions
(b) Decentralised optimisation of probability distributions
(c) Communicate distributions with other robots

Performed asynchronously by each robot
Monte Carlo tree search (MCTS)

➢ Tree search algorithm
  • Biased random sampling
  • Exploits “smoothness” of search space
  • Any-time
  • Only requires evaluation of full paths
  • Can incorporate problem-specific heuristics (if available)

Computer Go [Silver, 2017]
Monte Carlo tree search (MCTS)

[Browne, 2012]
Monte Carlo tree search (MCTS)

- Upper confidence bounds for trees (UCT) [Kocsis, 2006]

![Biased tree growth](Coquelin, 2014)

Average rollout score:

$$\arg\max_a \left[ \hat{Q}(s|h), a \right] + c \sqrt{\frac{\log n(s|h)}{n(s|h), a}}$$

Parent #visits

Child #visits

Exploitation

Exploration

Multi-armed bandit
Monte Carlo tree search (MCTS)
A naïve decentralised algorithm

What’s bad about this approach?

• Requires communicating a large tree
• How does robot 1 consider robot 2’s tree?
• Reward function changes
MCTS for multi-robot teams

- **Challenge 1**: Objective is a function of *all* robots

- Estimate expected reward with:
  1. Rollout policy for own action sequence
  2. Sample probabilistic belief for other robots

1. Search tree for **own actions**

2. Belief distribution for **other robots**
Discounted-UCT

➢ **Challenge 2:** *Reward distribution changes over time*
  • Observe: Recent rollouts are more relevant

➢ *Discounted* empirical average
  • Bias the samples by an increasing weight

\[
\bar{F}_t(\gamma, s, s_j) = \frac{1}{N_t(\gamma, s, s_j)} \sum_{u=1}^{t} \gamma^{t-u} F_u(s, s_j) \mathbb{1}_{\{s_u^+=s_j\}},
\]

\[
N_t(\gamma, s, s_j) = \sum_{u=1}^{t} \gamma^{t-u} \mathbb{1}_{\{s_u^+=s_j\}}.
\]

➢ The D-UCB policy is

\[
s_t^+ = \arg \max_{s_j} \left[ \bar{F}_t(\gamma, s, s_j) + c_t(\gamma, s, s_j) \right]
\]

• where

\[
c_t(\gamma, s, s_j) = B \sqrt{\frac{\xi \log N_t(\gamma, s)}{N_t(\gamma, s, s_j)}}
\]
Tree compression

- **Challenge 3:** Robots need to communicate their intentions
  - (a) Grow search tree for own actions
  - (b) Decentralised optimisation of probability distributions
  - (c) Communicate distributions with other robots

Performed asynchronously by each robot.
Tree compression: Probability distribution over paths

➢ Key insight: Communicate \textit{probabilistic} plans

probability $q^i(x^i)$

Optimised using distributed gradient descent

Selected as a subset of paths with the highest expected reward
Joint distribution

\[ p(x^1, x^2, ..., x^R) \]

Product distribution approximation

\[ \prod_{r=1}^{R} q^r(x^r) \approx \]

minimum KL divergence

\[ \prod_{r=1}^{R} q^r(x^r) \]
Distributed gradient descent over probability distributions

➢ We adapt a type of variational method: *probability collectives* [Wolpert, 2004]

➢ Gradient descent scheme:

\[
q^i(x^i_j) \leftarrow q^i(x^i_j) - \alpha q^i(x^i_j) \left[ \frac{\mathbb{E}_q[f^i] - \mathbb{E}_q[f^i | X^i = x^i_j]}{\beta} \right] + H(q^i) + \ln \left( q^i(x^i_j) \right)
\]

- Probability of an action sequence
- Reward improvement
- Entropy regulator
**Dec-MCTS overview**

(a) Grow search tree for own actions

(b) Decentralised optimisation of probability distributions

(c) Communicate distributions with other robots

Performed asynchronously by each robot
Theoretical analysis

- Decentralised planner for general multi-robot problems

- **Result 1:** The D-UCT policy guarantees a rate of regret in the case of abruptly changing distributions $q^i(x^i)$
  - Proven using result for a non-stationary multi-armed bandit problem [Garivier, 2011]
  - Extend result for trees by induction
  - **Tree search balances exploration and exploitation**

- **Result 2:** Restricting the domain $\hat{x}_n^i \subset X^i$ for the distributions is an approximation of importance sampling
  - Random subset selection: standard probability collectives
  - Our approach: reasonably accurate representation of $q_n(X)$
  - **Converges towards distribution that minimises the KL-divergence from the product to the joint distribution**
Useful practical properties

- Decentralised
- Asynchronous communication
- Robust to communication loss
- Balances exploration and exploitation
- Any-time
- Efficient replanning
Outline

1. Dec-MCTS

2. Objective functions → applications
A planning algorithm’s performance can only be as good as its objective function!

- Solve the problem I’m actually interested in solving
- Planning is difficult!

- Improved planner performance
- Exploit problem-specific characteristics
- Strong optimality guarantees → with respect to abstraction
1. Generalised orienteering problem

- Discrete set of elements to observe or collect
  - Features, landmarks, regions, targets, or more abstract quantities...
  - Observation dependencies modelled as overlapping subsets

Active object classification [Best, 2016]  
Precision agriculture [Calleija, 2014]

Sensor network data-collection [Faigl, 2014]  
Area coverage [Dornhege, 2016]
1. Generalised orienteering problem

Orienteering problem (duel of TSP)

Maximum set cover problem

Generalised team orienteering

Multiple robots

Continuous set cover
1. Generalised orienteering problem

- Abstraction of information gathering tasks
  - Viewpoint-dependent rewards
  - Some dependencies can be modelled

- Relatively easy to solve
  - Objective is fast to evaluate = more MCTS samples
  - Problem-specific solutions can exploit geometry of the problem
1. Generalised orienteering problem (time-varying)
2. Active object classification
2. **Active object classification**

1. Segment and classify objects in observed point cloud
2. “Hallucinate” unobserved parts using MLE for object class and location
3. Predict observation using ray-tracing
4. Evaluate predicted observation
   - Expected class estimation entropy

---

[Patten, 2016]
2. Active object classification

- Experimental results: Key findings:
  - Objective function suitable for Dec-MCTS
  - Long-horizon planning outperforms greedy
  - Replanning outperforms offline planning

Diagram:
- Planning
- Navigation
- Update
- Perception

Objects:
- Dec-MCTS
- Greedy

Graphical representation of navigation and planning processes.
3. Fruit tree reconstruction
3. Fruit tree reconstruction

Viewpoint evaluation

\[ g(x) = \text{ROI}_{\text{coverage}} + \text{exploration} \]

Motion roadmap
3. Fruit tree reconstruction

speed: 2.5x

viewpoint selection: coordinated
observations: 0
ROI points: 0
4. Monitoring ocean fronts

An extreme ocean front
4. Monitoring ocean fronts

<table>
<thead>
<tr>
<th>Oceanographers’ observation</th>
<th>Desired robot behaviour</th>
<th>Solution</th>
</tr>
</thead>
<tbody>
<tr>
<td>Interesting phenomena occur in ocean fronts</td>
<td>Observe high-gradient regions</td>
<td>$\sum_{edges}</td>
</tr>
<tr>
<td>The location of the front is unknown</td>
<td>Observe unknown regions</td>
<td>$\sum_{edges} \text{variance}$</td>
</tr>
<tr>
<td>Want to visit new parts of the front</td>
<td>Observe unvisited, high-gradient regions</td>
<td>$\sum_{edges}</td>
</tr>
<tr>
<td>Robots are affected by currents</td>
<td>Accurately predict arrival times</td>
<td>Current-dependent edge costs</td>
</tr>
<tr>
<td>Long-term predictions are unreliable</td>
<td>Do not over-trust long-term predictions</td>
<td>Time decay</td>
</tr>
<tr>
<td>Turning corrupts sensor data</td>
<td>Favour straight paths</td>
<td>Multiplicative turning penalty</td>
</tr>
<tr>
<td>Need to avoid collisions</td>
<td>Don’t crash!</td>
<td>Modify time-varying roadmap</td>
</tr>
<tr>
<td>Robots surface asynchronously</td>
<td>Use current information</td>
<td>Asynchronous replanning</td>
</tr>
<tr>
<td>Fronts drift over time</td>
<td>Robots track front over time</td>
<td>Plan relative to a moving frame</td>
</tr>
</tbody>
</table>
4. Monitoring ocean fronts

Mission Duration: 1 day, 23:52:00  0.2 m/s

Ground Truth in Planning Frame

Estimate Field  0.2 m/s

Estimate Variance
Conclusion and Outlook
Conclusions

1. Dec-MCTS
   • *General decentralised planner for information gathering*

2. Objective functions → applications
   • *Objective function design is important!*

“perfect” representation of your **problem**

? abstraction suitable for your **planner**
Outlook

Communication

Human-robot

Field experiments

Deep learning

Mixed centralised-decentralised

References

➢ Dec-MCTS algorithm

➢ Generalised team orienteering

➢ Object classification
References

Region of interest reconstruction


Other

Multi-robot active perception: Dec-MCTS, objective functions, and applications

Graeme Best
bestg@oregonstate.edu

Collaborators: Robert Fitch, Geoffrey Hollinger, Oliver Cliff, Timothy Patten, Ramgopal Mettu, Seth McCammon, Fred Sukkar, Andrew Smith, Jan Faigl, Michael Forrai, Chanyeol Yoo, (USyd, OSU, Czech Tech., UTS, Tulane)

COLLABORATIVE ROBOTICS AND INTELLIGENT SYSTEMS (CORIS) INSTITUTE