Empirical Evaluation of Ad Hoc Teamwork in the Pursuit Domain

Samuel Barrett[1] Peter Stone[1] Sarit Kraus[2]

[1]University of Texas at Austin
{sbarrett,pstone}@cs.utexas.edu

[2]Bar-Ilan University
sarit@cs.biu.ac.il

AAMAS 2011
May 4, 2011

UT Austin Learning Agents Research Group

Ad Hoc Teamwork

- Only in control of a single agent
- Unknown teammates
- Shared goals
- No pre-coordination

Examples:

- Pick up soccer
- Search and rescue

Motivation

- Agents are becoming more common and lasting longer
  - Both robots and software agents
- Pre-coordination may not be possible
- Most previous work on ad hoc teams was theoretical

Motivation

- Agents are becoming more common and lasting longer
  - Both robots and software agents
- Pre-coordination may not be possible
- Most previous work on ad hoc teams was theoretical

Research Question:
How can an ad hoc agent help its team in the pursuit domain?

Ad Hoc Agent Evaluation

Evaluate(a, A, D):

- Initialize performance (reward) counter $r = 0$.
- Repeat:
  - Sample a task d from D.
  - Randomly draw a subset of agents B, from A such that $E[s(B, d)] \geq s_{min}$.
  - Randomly select one agent $b \in B$ to remove from the team to create the team B^{-}.
  - Increment r by $s(\{a\} \cup B^{-}, d)$
- If Evaluate(a_0, A, D) $>$ Evaluate(a_1, A, D) and the difference is significant,then we conclude that a_0 is a better ad hoc team player than a_1 in domain D over the set of possible teammates A.

Pursuit Domain

- Grid world - Torus
- 4 Predators and 1 Prey
- Predators' goal is to surround the prey as quickly as possible
- Act simultaneously
- Collisions randomly decided - loser stays still

Agent Control

- Observe positions of all agents
- Cannot explicitly communicate
- 5 actions: Stay still, up, down, left, and right

Agent Types

Agent Types:

- Greedy - moves to nearest open cell neighboring the prey

Agent Types

Agent Types:

- Greedy - moves to nearest open cell neighboring the prey
- Teammate-aware - lets the farthest predator have the closest cell

Agent Types

Agent Types:

- Greedy - moves to nearest open cell neighboring the prey
- Teammate-aware - lets the farthest predator have the closest cell
- Greedy Probabilistic - greedy, but with chance of taking a longer path

Agent Types

Agent Types:

- Greedy - moves to nearest open cell neighboring the prey
- Teammate-aware - lets the farthest predator have the closest cell
- Greedy Probabilistic - greedy, but with chance of taking a longer path
- Probabilistic Destinations - moves towards a random cell that is closer to the prey

Approach

If the ad hoc agent has:

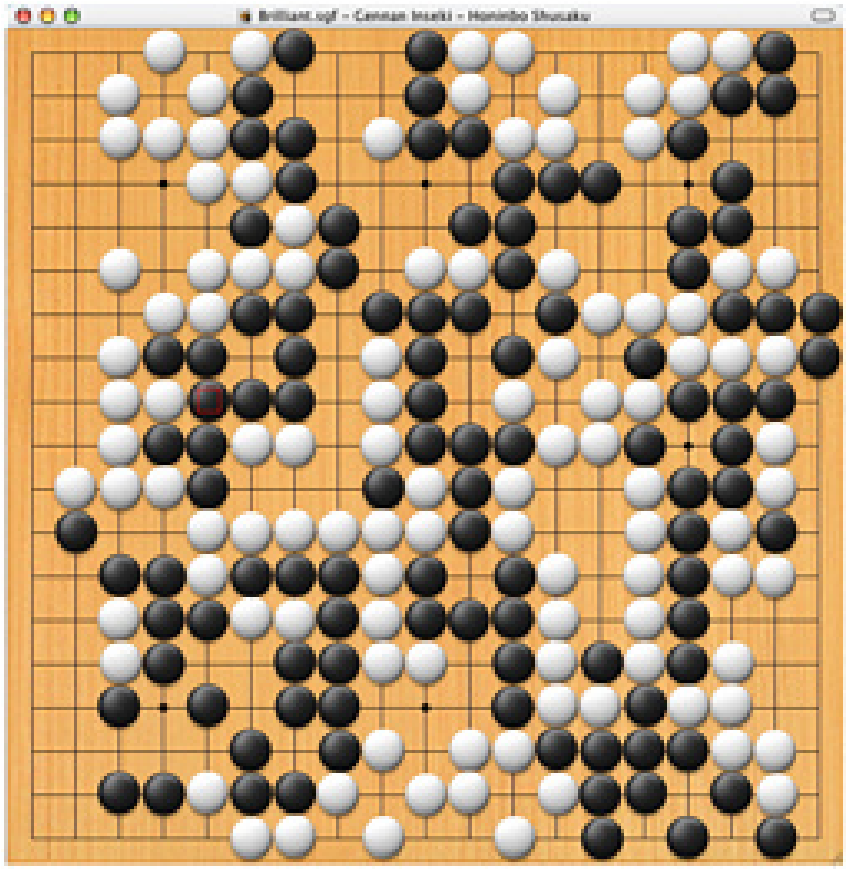
- Knows dynamics of world
- Knows prey's behavior
- Knows teammates' behavior

Then it can plan about the effects of actions and their values

Value Iteration

- If the teammates' types are known
- Calculates the optimal action values
- Slow for large worlds
- Impractical because of the size of the state space

Monte Carlo Tree Search



- Sample playouts
- Focus on relevant state actions
- UCT balances exploration vs. exploitation
- Efficient

Model probabilities

- Set of known models
- Start with prior belief
- Update using the probability that a model would have taken the observed action

$$P(\text{model}|\text{actions}) = \frac{P(\text{actions}|\text{model}) * P(\text{model})}{P(\text{actions})}$$

Evaluation

- Can the ad hoc agent do better than copying its teammates' behaviors?
- Number of steps the team takes to capture the prey

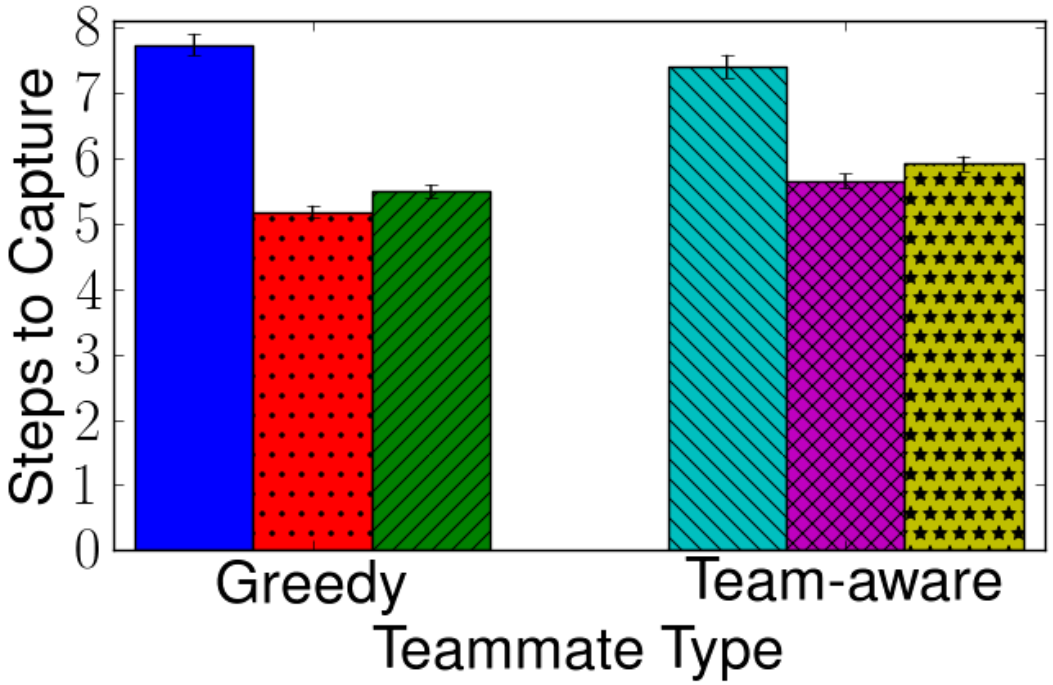
Evaluation

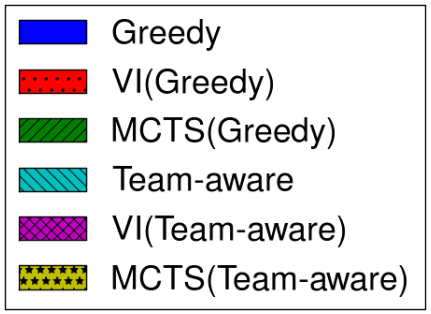
- Can the ad hoc agent do better than copying its teammates' behaviors?
- Number of steps the team takes to capture the prey
- 1,000 episodes
- No information carried between episodes
- Random start positions per episode, but same across evaluations

Known deterministic teammates

- Ad hoc agent knows the its teammates' type
- Planning outperforms copying teammates' behavior
- Performance of MCTS is close to that of VI

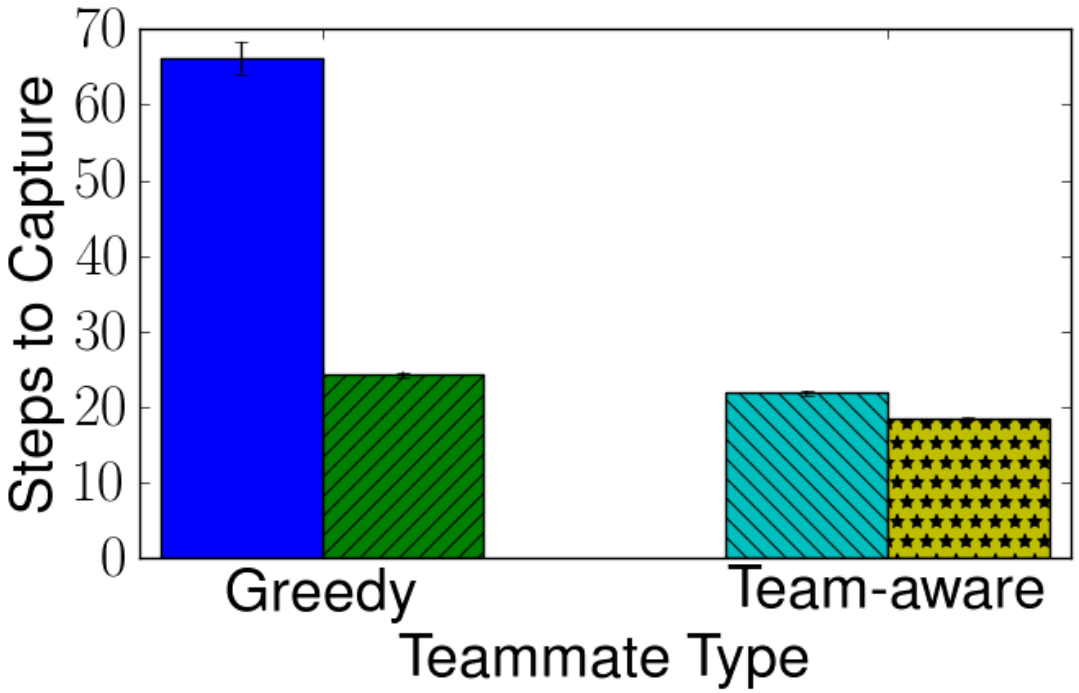
Known deterministic teammates

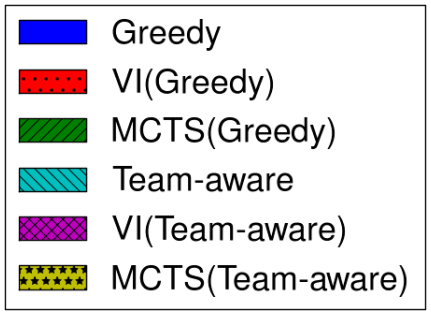




5x5 World

Known deterministic teammates





20x20 World

Incorrect type

- Ad hoc agent is incorrect about its teammates' type
- All methods perform poorly

Incorrect Type

Known set of teammates

- Set of possible agent types is known
- Ad hoc agent tracks probabilities of types
- Low loss compared to knowing correct model

Known set of teammates

Known set of teammates

Teammates of Unknown Types

- Ad hoc agent does not know behavior of teammates
- Set of known types
- True ad hoc scenario
- Planning should outperform copying

Teammates of Unknown Types

Related Work

Conclusions

- Ad hoc agents can help their teams
- Can do better than copying teammates
- MCTS is effective and efficient for planning
- Can differentiate teammate types
- Models still help when incorrect

Future Work

- Can we learn a model on the fly?
- Can we learn to correct an existing model?
- Will other domains get similar results?
- How can the ad hoc agent reason about the value of information?
- How can an ad hoc agent deal with incomplete communication?

Thank You!

- Ad hoc team agents can learn to help their teams on the fly

Learning Agents Research Group