Ad Hoc Teamwork by Learning Teammates' Task

Sudeep Basnet, Tyler Bienhoff, Ian Howell, Yi Liu

Melo and Sardinha (2016). *Autonomous Agents and Multi-Agent Systems,* **30**:175-219

Introduction

- Provides the novel perspective in the ad hoc teamwork problem, emphasizing the importance of task and teammate identification to influence task planning.
- Presents two new ways of modeling an ad hoc agent
  - An online learning approach (OL)
  - Partially observable Markov decision problem (POMDP)

Some Background

The ad hoc teamwork problem is the problem that a generalized agent, join teammate agent(s) must determine the task the teammate(s) are working towards and the role it plays in that task.

- Pick up soccer games, join an N - 1 team
- E-Commerce with specialized agents, building personal computer packages

The ad hoc agent receives no reward from the environment. The only feedback it receives are the actions it’s teammate(s) take. Using this information, the ad hoc agent must recognize what task it needs to do from a set of possible tasks.

Some Background

Assumption 1: Bounded Rationality

- Teammate agents have a finite amount of memory, uses at most the last N steps to make next decision
- Selects a best response to the ad hoc agent’s actions

E-Commerce Scenario

Two agents build a computer package.
One is specialized to build the LCD monitor
 - Needs to purchase the LCD panel
One is specialized to build the desktop computer
 - Needs to purchase the Motherboard
Specialized agents save $2 if they build their specialized item

Packages are sold for $25

Agents must cooperate to maximize profit

E-Commerce Scenario

Two suppliers for the components with different pricing schemes.
Buying from the same supplier reduces shipping cost.

Table 1 Price and shipping cost of different parts

	LCD panel price	Motherboard price	Shipping cost
Supplier A	$10	$7	$2
Supplier B	$7	$7	$5

E-Commerce Scenario

Must determine which payoff matrix is being used.

X axis point (C, T) represents the specialized agent purchasing T from C supplier, while Y axis point (D, S) represents the ad hoc agent purchasing S from D supplier.

	A, LCD	B, LCD	A, Motherboard	B, Motherboard
A, LCD	−22	−24	6	1
B, LCD	−24	−19	4	6
A, Motherboard	4	2	−16	−21
B, Motherboard	−1	4	−21	−19

Fig. 2 Payoff matrix for the task "Replace the agent optimized to build LCD Monitors"

	A, LCD	B, LCD	A, Motherboard	B, Motherboard
A, LCD	−22	−24	4	−1
B, LCD	−24	−19	2	4
A, Motherboard	6	4	−16	−21
B, Motherboard	1	6	−21	−19

Fig. 3 Payoff matrix for the task "Replace the agent optimized to build desktop computers"

Pursuit Scenario

- Played on a toroidal finite grid world
- Predator agents move to surround prey agent
- Modified: Two predator agents capture prey by enclosing the prey in a line
  - One such line of the four possible is the capture configuration; four possible tasks
  - Dimensionality forces only iteration to be viable
  - Positive value given to teammates on capture

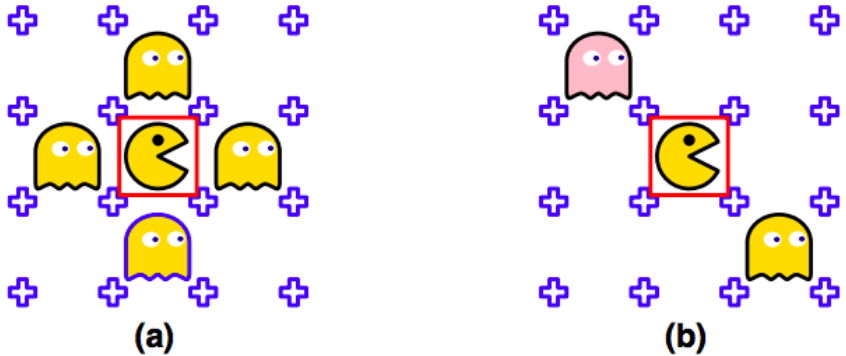


Fig. 13 Capture configurations **a** in the classical pursuit domain; **b** in the modified pursuit domain

Online Learning Approach - Action and Loss

- The ad-hoc agent, at each time-step *n*, should predict the action of its teammate, $A_{-\alpha}(n)$, using the model specified in Assumption 1 (bounded rationality).
- The ad-hoc agent's ability to predict teammate's task, is an indicator of the agent's identification of **target** task and teammate **strategy**.
- At each time-step *n*, ad-hoc agent selects action:

$$\hat{A}(n) = \langle A_{\alpha}(n), \hat{A}_{-\alpha}(n) \rangle$$

- And incurs loss, which penalizes wrong predictions.

$$\ell(\hat{A}(n), A_{-\alpha}(n)) = 1 - \delta(\hat{A}_{-\alpha}(n), A_{-\alpha}(n))$$

- With this loss defined, it is possible to recast the teamwork problem as a simple *online learning problem*.

Online Learning Approach - Valuation and Expert

- For each task (τ), we define estimated valuation of each action of agent k:

$$\hat{V}_\tau^k(h_{1:n}, a_k) = \frac{1}{N}\sum_{t=0}^{N-1} U_\tau(\langle a_k, a_{-k}(n-t)\rangle), \qquad k = \alpha, -\alpha.$$

history

- Set of maximizing actions denoted by: $\hat{\mathcal{A}}_\tau^k(h_{1:n})$ ((maximum value actions) $E_\tau : \mathcal{H} \times \mathcal{A} \to [0, 1]$
- **Expert** : , **for each task**, based on all finite histories and E_τ ions.
- is the probability of selecting the joint action *a* as a best response to the history according to the task.

Online Learning Approach - Loss

- Predictor $P : \mathcal{H} \times \mathcal{A} \to [0, 1]$
  such that, $\sum_{a \in \mathcal{A}} P(h_{1:n}, a) = 1$
- The value $P(h_{1:n}, a)$ is an estimate of the probability that *A(n+1)* is action *a*, given the history. This is a generalization of the notion of experts.
- The cumulative loss of expert
  $L_\tau(h_{1:n}) \triangleq \sum_{t=0}^{n-1} \ell_\tau(h_{1:t}, a_{-\alpha}(t+1))$
- Similarly, cumulative loss of predictor:
  $L_P(h_{1:n}) = \sum_{t=0}^{n-1} \ell_P(h_{1:t}, a_{-\alpha}(t+1))$

Online Learning Approach - Expected Regret

- Cumulative loss of Expert: $L_\tau(h_{1:n})$
- Cumulative loss of Predictor: $L_P(h_{1:n})$

- Solving the ad hoc teamwork problem, now, consists of determining a predictor *P* that minimizes the expected regret, given by:

$$R_n(P, \mathcal{E}) = \mathbb{E}\left[L_P(h_{1:n}) - L_\tau(h_{1:n})\right]$$

- where, $\mathcal{E} = \{E_\tau, \tau \in \mathcal{T}\}$

Online Learning Approach - Algorithm 1

Algorithm 1 Exponentially weighted forecaster for the ad hoc teamwork problem.

1: Initialize $w_\tau^{(0)} = 1, h = \emptyset, t = 0$.
2: **for all** t **do**
3: Let $t \leftarrow t + 1$
4: Let

$$P(h, a) = \frac{\sum_{\tau \in \mathcal{T}} w_\tau^{(t)} E_\tau(h, a)}{\sum_{\tau' \in \mathcal{T}} w_{\tau'}^{(t)}}$$

5: Select action $\hat{A}(t) = \text{argmax}_{a \in \mathcal{A}} P(h, a)$
6: Observe action $A_{-\alpha}(t)$
7: Compute loss $\ell_\tau(h, A_{-\alpha}(t))$ as in (4), $\tau \in \mathcal{T}$
8: Update

$$w_\tau^{(t)} \leftarrow w_\tau^{(t-1)} \cdot e^{-\gamma_t \ell_\tau(h, A_{-\alpha}(t))}$$

9: **end for**

Weight is 1 in the beginning.

exponentially weighted average predictor.

ad hoc agent's action & predicted teammate's action, with highest value of P.

Partially Observable Markov Decision Process (POMDP)

Problems that online learning approach didn’t resolve:

1. Target task information in history actions is not encoded in probabilistic function
    - How much confidence on teammates’ tasks

1. Impacts of the actions that the ad hoc agents took
    - Ad hoc agents’ actions are part of collaboration

POMDP - *belief* Update

- Recall Fictitious Play (Reference: Handout 9)

$$P(a) = \frac{w(a)}{\sum_{a' \in A} w(a')}$$

Observation

History

- *belief* updating in POMDP

$$p_{n+1}(x') = B(p_n, z, a)_{x'} \triangleq \xi \sum_{x \in \mathcal{X}} p_n(x) P(x' \mid x, a) O(z \mid x', a)$$

History

Observation

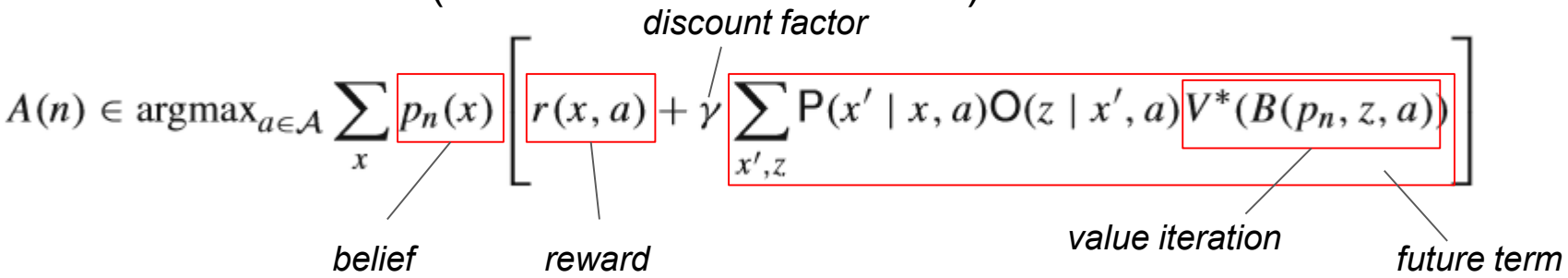
Transition Function

POMDP - Action Selection via Value Iteration

Goal: Maximize rewards over time

Strategy: Select best possible action to each state based on *belief*

Method: Value Iteration (Recall MDP in Handout 11)

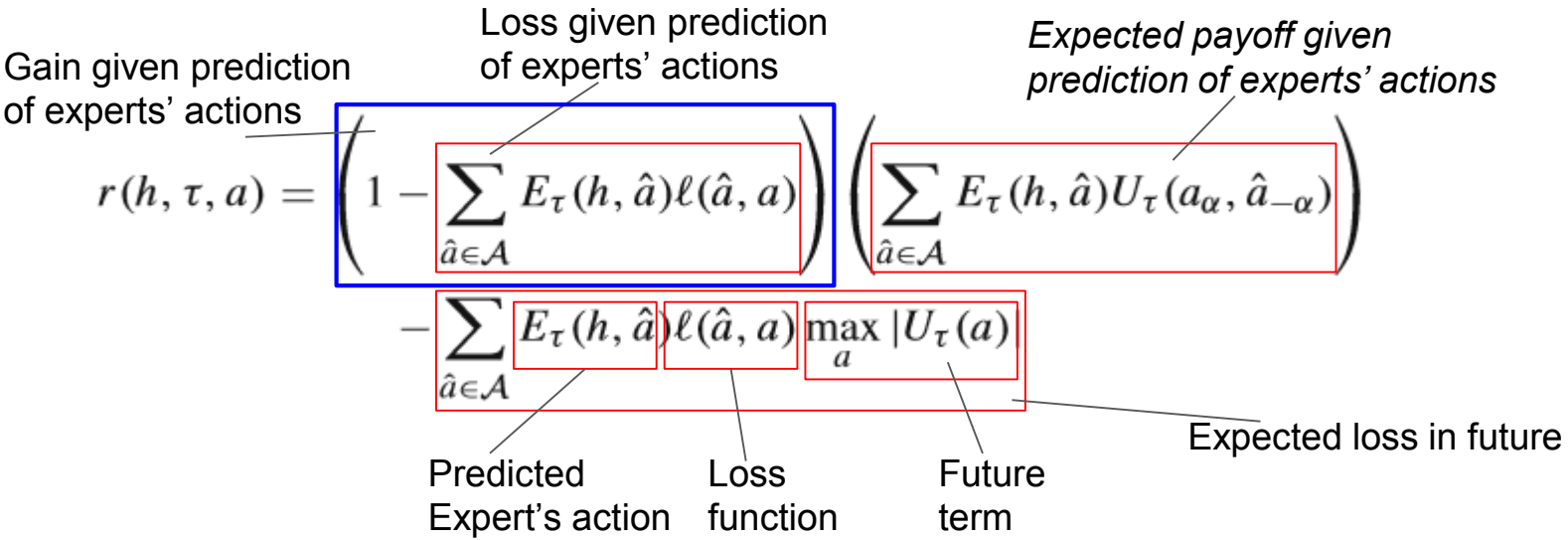


$$V^*(p_n) = \max_{a \in \mathcal{A}} \sum_x p_n(x) \left[r(x,a) + \gamma \sum_{x',z} \mathsf{P}(x' \mid x,a) \mathsf{O}(z \mid x',a) V^*(B(p_n,z,a)) \right]$$

Recursively calculate the utility of each action (*Recall Q-Learning*)

POMDP - Reward Function

- Goal: encode the environment payoff which is not observable by ad hoc agent.



POMDP - Tradeoff

Performance vs. Complexity

- The complexity of approximation of value iteration V^* is related to the length of history.

- The longer the history length, the harder the approximation.

Empirical Evaluation

- 3 experiments
  - Simple e-commerce scenario of length 100
  - Randomly generated payoff matrices with varying number of:
    - Tasks
    - Agents
    - Actions
  - Applicability to pursuit domain

Experiment Basics

- One or more “legacy agents”
  - Know the target task
  - Programmed to adapt actions to ad hoc agent
- One ad hoc agent
  - Must infer target task (from set of tasks)
  - Coordinate actions with other agents
- Performance measured in loss and payoff

Agents

- Online Learning (OL)
- Partially Observable Markov Decision Problem (POMDP)
- Online Learning with known target task (OL k.t)
- Markov Decision Problem (MDP)
- Reinforcement Learning (RL)

E-Commerce Scenario

Table 3 Performance of the different approaches in the *e*-commerce scenario for different horizon lengths. The results are averages over 1, 000 independent Monte Carlo runs

	Agent	$H = 1$	$H = 2$	$H = 3$
Loss	POMDP	1.468 ± 1.403	1.365 ± 1.181	1.255 ± 1.060
	OL	1.500 ± 1.565	1.389 ± 1.269	1.294 ± 1.026
	OL (known task)	1.510 ± 1.399	1.395 ± 1.173	1.298 ± 0.946
Payoff	POMDP	571.0 ± 36.2	571.0 ± 31.3	572.0 ± 32.0
	OL	505.7 ± 112.0	503.6 ± 111.8	497.9 ± 114.2
	MDP (known task)	522.8 ± 82.7	531.0 ± 83.00	541.1 ± 79.7
	RL	426.6 ± 116.4	273.2 ± 185.6	-113.1 ± 364.2

Results

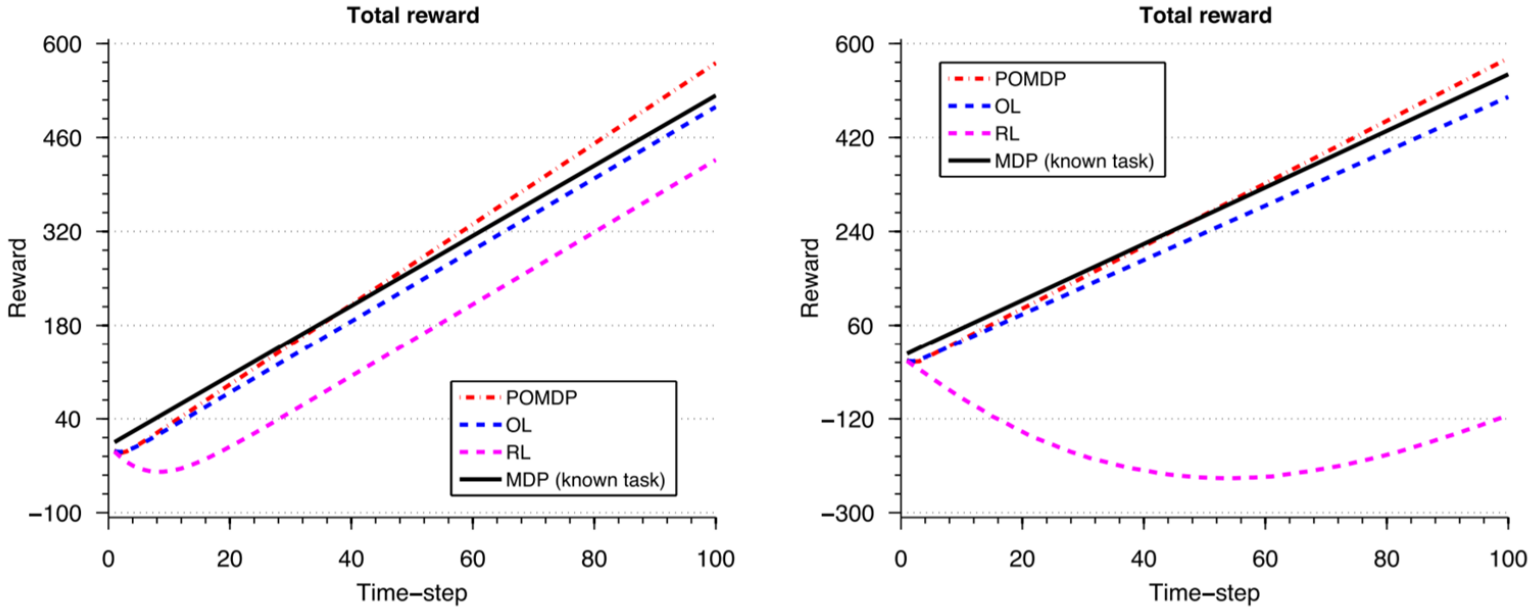
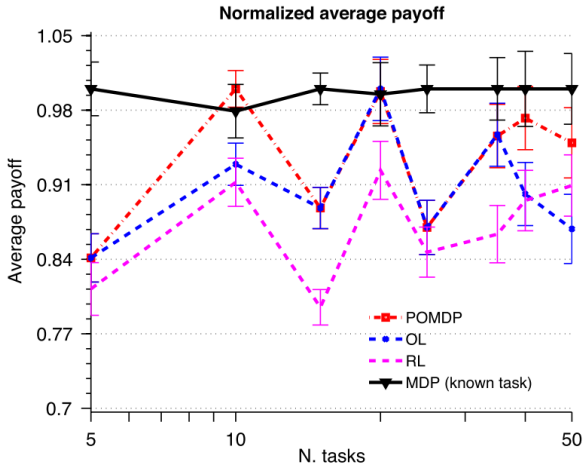
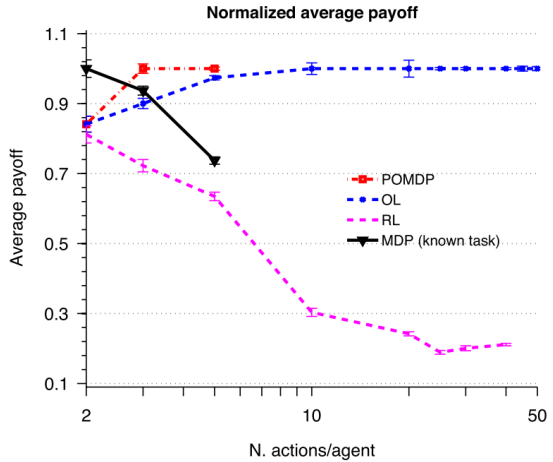
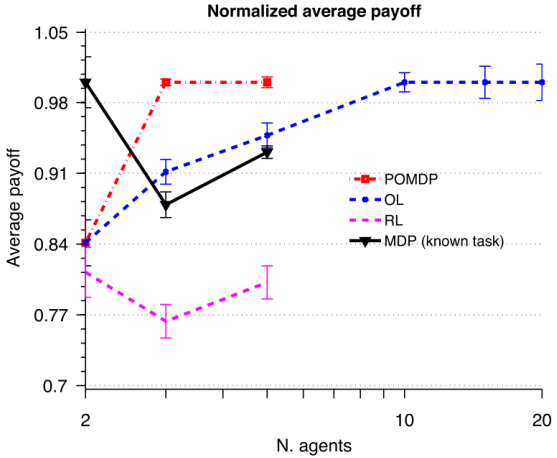


Fig. 8 Average payoff of the different approaches in the *e*-commerce scenario, for different horizon lengths. The results are averages over 1,000 independent Monte Carlo runs

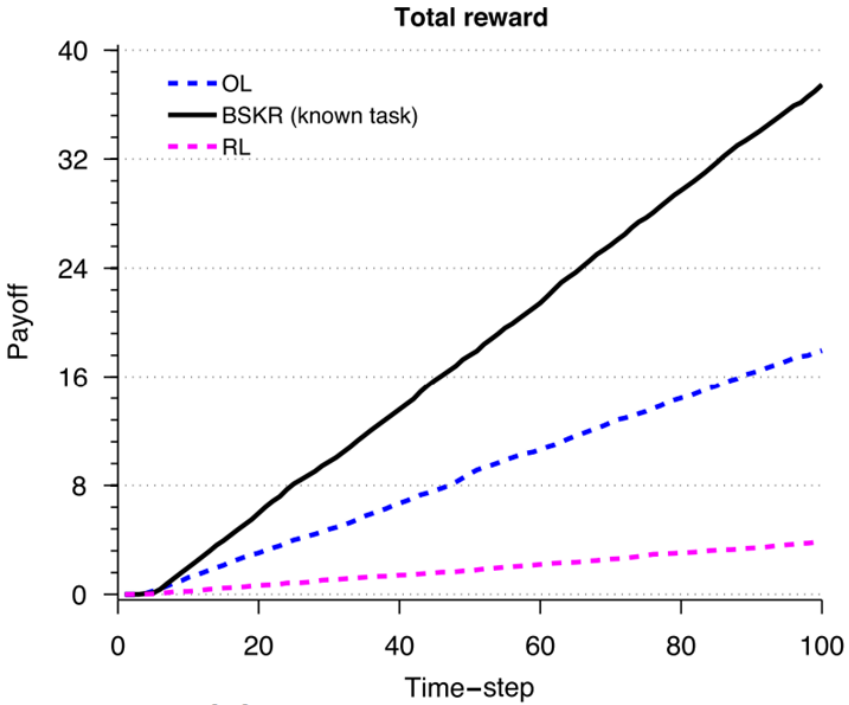
Scalability of Ad Hoc Teamwork



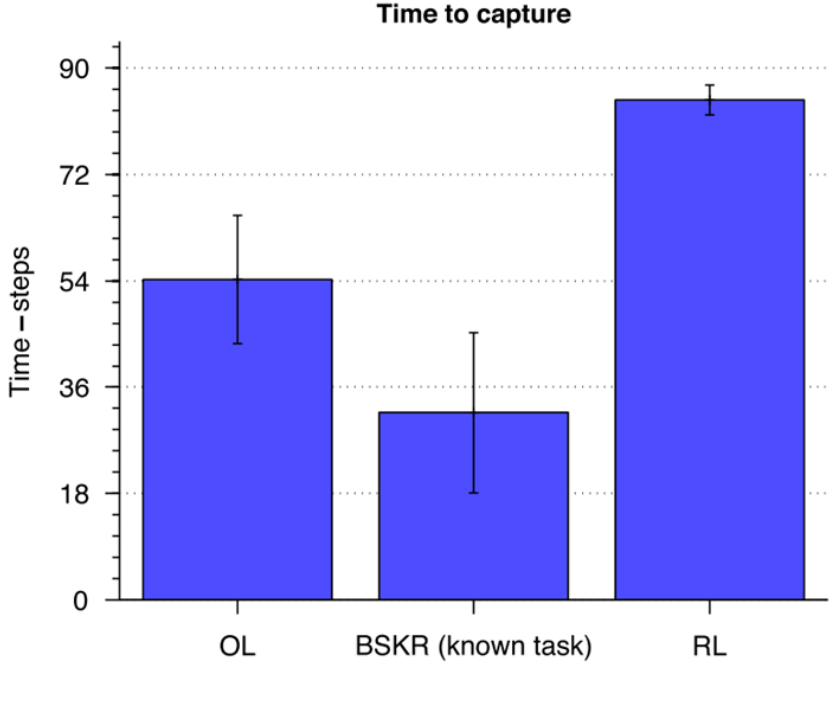




Pursuit Domain



(a) Total average payoff



(b) Average time to capture the prey

Conclusions

Key Contributions

- Novel perspective of the ad hoc teamwork problem, focusing on task and teammate identification influencing planning.
- Formalized the ad hoc teamwork problem into a sequential decision problem.
- Proposes two novel approaches for identifying tasks and teammates.

Both approaches heavily rely on Bounded Rationality, as it controls the decision mechanisms.

Our Conclusions

Comprehensive literature review, including optimality of ad hoc agents and reactivity of teammates to ad hoc agents' actions.

Good reformulation and perspective shift of the ad hoc teammate problem. More fully analyzes the problem as a whole.

Approaches are useful for agent teammates that rely on most recent memories, but not for those whose memories are simplified models with (near) infinite memory, e.g. Bayesian Tables, Neural Networks, etc.

Questions

Online Learning Approach - Example

- τ_1 = Replace the agent optimized to build LCD Monitors
- τ_2 = Replace the agent optimized to build desktop computers
- $\mathcal{A}_\alpha = \mathcal{A}_{-\alpha} = \{$(A, LCD), (B, LCD), (A, MB), (B, MB)$\}$
- In the beginning, the history is empty and Ad hoc agent will predict actions uniformly at random.
- Ad hoc agent chooses action $A_1(1) =$ (B, LCD) and predicts action of the teammate $\hat{A}_2(1) =$ (A, MB)
- Teammate actually choose: $A_2(1) =$ (A, LCD)
- The history is h_1 = {(B, LCD), (A, LCD)}.
- Also, $L_{\tau_1}(h_1) = 1,$ $L_{\tau_2}(h_1) = 0.5,$ $L_P(h_1) = 0.75$
  And, $R_0(P, \mathcal{E}) = 0.25$

Online Learning Approach - Example (continued)

Second Step:

$$\hat{V}^{\alpha}_{\tau_1}(h_1, (\text{A, LCD})) = -22, \qquad \hat{V}^{\alpha}_{\tau_1}(h_1, (\text{B, LCD})) = -24,$$
$$\hat{V}^{\alpha}_{\tau_1}(h_1, (\text{A, MB})) = 4, \qquad \hat{V}^{\alpha}_{\tau_1}(h_1, (\text{B, MB})) = -1,$$
$$\hat{V}^{\alpha}_{\tau_2}(h_1, (\text{A, LCD})) = -22, \qquad \hat{V}^{\alpha}_{\tau_2}(h_1, (\text{B, LCD})) = -24,$$
$$\hat{V}^{\alpha}_{\tau_2}(h_1, (\text{A, MB})) = 6, \qquad \hat{V}^{\alpha}_{\tau_2}(h_1, (\text{B, MB})) = 1,$$

Ad hoc agent selects action $A_1(2) = (\text{A, MB})$ and predicts $\hat{A}_2(2) = (\text{B, MB})$.

$$\hat{V}^{-\alpha}_{\tau_1}(h_1, (\text{A, LCD})) = -24, \qquad \hat{V}^{-\alpha}_{\tau_1}(h_1, (\text{B, LCD})) = -19,$$
$$\hat{V}^{-\alpha}_{\tau_1}(h_1, (\text{A, MB})) = 4, \qquad \hat{V}^{-\alpha}_{\tau_1}(h_1, (\text{B, MB})) = 6,$$
$$\hat{V}^{-\alpha}_{\tau_2}(h_1, (\text{A, LCD})) = -24, \qquad \hat{V}^{-\alpha}_{\tau_2}(h_1, (\text{B, LCD})) = -19,$$
$$\hat{V}^{-\alpha}_{\tau_2}(h_1, (\text{A, MB})) = 2, \qquad \hat{V}^{-\alpha}_{\tau_2}(h_1, (\text{B, MB})) = 4.$$

Again, $R_1(P, \mathcal{E}) = 0.25$