

Planning, Execution & Learning: Multi-Agent Planning II

Reid Simmons
Manuela Veloso

Planning, Execution & Learning: Multi-Agent Planning II 1

Simmons & Veloso : Fall 2008

Multi-Agent Markov Models

- **MMDP** (Multi-Agent MDP)
 - Actions are joint actions of agents
 - State completely observable
 - Complexity same as that of MDP's
- **Dec-POMDP** (Decentralized POMDP)
 - Actions are joint actions of agents
 - Optimal actions based on *joint observations* of agents
 - Complexity is NEXP-complete
- **Dec-MDP** (Decentralized MDP)
 - Actions are joint actions of agents
 - State is *collectively observable*
 - Complexity same as that of DEC-POMDP (!)

Planning, Execution & Learning: Multi-Agent Planning II 2

Simmons & Veloso : Fall 2008

Why are Dec-POMDPs Hard?

- Single-agent POMDP: Agent must plan over all possible observation histories (PSPACE-complete)
- Dec-POMDP: Agents must consider own possible observations histories *as well as* possible observations *and* possible actions of their teammates
 - Policy is mapping from history of (joint) observations to actions
 - Even with communication, Dec-POMDPs remain NEXP
 - Agents must reason about whether to communicate
 - With free communication, complexity reduces to POMDP
 - Optimal policy is to communicate at every time step

Planning, Execution & Learning: Multi-Agent Planning II 3

Simmons & Veloso : Fall 2008

Solving DEC-POMDPs

- Incremental Approximation (Nair & Tambe)
 - Encode distribution of observation histories as a *tree*
 - Generate policies for each agent individually (over observation histories)
 - Fix policies of all but one agent and then solve for that agent
 - Still solving over observation histories of all agents, *but* branching factor is much smaller since most of the joint policy is known
 - Converges to Nash equilibrium
 - Can benefit from restarts
- Assumes *no* Communication

Planning, Execution & Learning: Multi-Agent Planning II 4

Simmons & Veloso : Fall 2008

Transition Independent DEC-POMDPs

(Becker, Zilberstein, Lesser, Goldman)

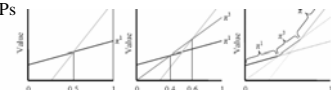
- Transition Independent DEC-POMDPs
 - Actions of one agent cannot affect any other agent's observations or local state
 - Coordinate through a global value function
 - Complementary (superadditive)
 - Redundant (subadditive)
 - Also assumes no communication

Planning, Execution & Learning: Multi-Agent Planning II 5

Simmons & Veloso : Fall 2008

Solving Transition Independent DEC-POMDPs

- (Relatively) Tractable Algorithm for Optimal Joint Policy
 - **Augmented** MDP
 - Depends on local reward to agent, joint reward, and other agent's policy
 - Other policy not known – represent through parameter space
 - **Optimal Coverage Set**
 - Set of all optimal policies corresponding to *any* possible policy of the other agent
 - Can represent compactly using parameter space of augmented MDPs



Planning, Execution & Learning: Multi-Agent Planning II 6

Simmons & Veloso : Fall 2008

Planning to Communicate (Nair & Roth)

- Communication acts to synchronize beliefs of all agents
 - One agent decides when to communicate
 - All agents broadcast observation histories
 - All agents update beliefs
 - Representation: belief *plus* observations
- Communicate *at least* every K time steps
 - Communication is just another action
 - Transition moves to synchronized belief and “resets” observation histories
 - May lead to exponential number of possible synchronized beliefs
- Solve using Nair & Tambe iterative planning method

Planning, Execution & Learning: Multi-Agent Planning II 7 Simmons & Veloso : Fall 2008

Planning to Communicate (Roth)

- **Insight:** Can trade off computation at plan time for computation at run time
- **Approach:**
 - At *plan time*:
 - Create centralized policy for joint-action POMDP
 - At *run time*:
 - Each agent maintains tree of possible joint beliefs
 - Decrease space requirements using particle filter
 - Use heuristic to estimate best action (akin to QMDP)
 - Communicate if by doing so a better action would be chosen

Planning, Execution & Learning: Multi-Agent Planning II 8 Simmons & Veloso : Fall 2008

Multi-Agent Tiger Domain

States: {SL, SR}
Tiger is either behind left door or behind right door

Individual Actions:
 $a_i \in \{\text{OpenL}, \text{OpenR}, \text{Listen}\}$
Robot can open left door, open right door, or listen

Individual Observations:
 $o_i \in \{\text{HL}, \text{HR}\}$
Robot can hear tiger behind left door or hear tiger behind right door
(Observations are noisy and independent)

Joint Rewards:
+20: Both agents open door with treasure
-50: Both agents open door with tiger
-100: Only one agent opens door with tiger
-1: Cost for listening

Planning, Execution & Learning: Multi-Agent Planning II 9 Simmons & Veloso : Fall 2008

Possible Joint Beliefs

Choose joint action by computing expected reward over all leaves

$$Q_{POMDP}(L^i) = \arg \max_{a^i} \sum_{b^i} p(b^i) \times Q(b^i, a^i)$$

Agents will independently select same joint action...
but action choice is very conservative (always <Listen, Listen>)

$$p^i = p^{i-1} \times P(o^i | b^{i-1}, a^{i-1})$$

Planning, Execution & Learning: Multi-Agent Planning II 10 Simmons & Veloso : Fall 2008

DEC-COMM Example

$a_{MC} = Q\text{-POMDP}(L^i) = \langle \text{Listen}, \text{Listen} \rangle$
 L^* = circled nodes
 $a_c = Q\text{-POMDP}(L^*) = \langle \text{OpenR}, \text{OpenR} \rangle$
 $V(a_c) - V(a_{MC}) > \epsilon$

Agent 1 communicates

Planning, Execution & Learning: Multi-Agent Planning II 11 Simmons & Veloso : Fall 2008

DEC-COMM Example (cont'd)

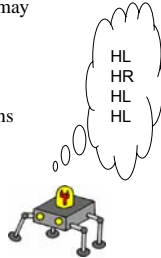
Agent 1 communicates <HL, HL>
Agent 2 communicates <HL, HL>
 $Q\text{-POMDP}(L^2) = \langle \text{OpenR}, \text{OpenR} \rangle$

Agents open right door!

Planning, Execution & Learning: Multi-Agent Planning II 12 Simmons & Veloso : Fall 2008

What to Communicate

- Choose Valuable Observations
 - Observations may be redundant
 - In some domains, some observations may be more informative than others
 - There may be bandwidth limits
- Want to choose only those observations that help change the joint action

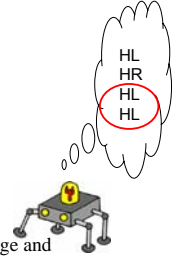


Planning, Execution & Learning: Multi-Agent Planning II 13

Simmons & Veloso : Fall 2008

What to Communicate

- First, Determine if Communication is Necessary
 - Calculate A_{NC} using Q-POMDP
 - Calculate A_C using Dec-Comm
 - If $A_C = A_{NC}$, do not communicate
- Greedily Build Message
 - “Hill-climbing” towards A_C
 - Choose single observation that most increases Q-POMDP value of A_C
 - Continue until either A_{NC} changes or bandwidth limit reached
 - **Alternate:** Start with complete message and eliminate least informative observations



Planning, Execution & Learning: Multi-Agent Planning II 14

Simmons & Veloso : Fall 2008