AAMAS 2013

Saint Paul
Minnesota
USA

6th - 10th May 2013

twelfth international conference on autonomous agents and multiagent systems

T7 – Tutorial on Self-Interested Decision Making in Sequential Multiagent Settings

May 6 full day

AAMAS-13 Tutorial

Self-Interested Decision Making in Sequential Multiagent Settings

Speakers

Prashant Doshi
Univ. of Georgia

Zinovi Rabinovich
Mobileye, Inc.

UGA

World of catastrophes

- Nature
  - 2004/12/26 – Sumatra-Andaman Earthquake
    - Magnitude estimate between 9.1 and 9.3
    - Triggered tsunamis causing 230,000 fatalities
  - 2005/08 – Hurricane Katrina
    - 1,836 dead
    - $81.2 billion damage
- Human
  - 26 April 1986 – Chernobyl atomic reactor meltdown
  - 11 September 2001 – Twin Towers in New York

Catastrophes: science

- Great Hanshin earthquake (1995). Killed over 6,400 people in and around Kobe, Japan.
- The data served to prototype a rescue simulation: **Robocup Rescue Domain**
  - Captures the dynamics of natural and man factor disasters and civil disorders
    - Includes uncertainty of various parameters
  - Realistically simulates the events: fire, traffic, building collapses, road blockage, etc.

Robocup Rescue - Scenario

- Given a post-event situation
  - Civilians trapped under collapsed buildings, and their life signs weakening with time
  - Some access routs are blocked or destroyed
  - Fires and civil disorder start and spread throughout the event site
- Manage platoons of Fire brigades, Police forces and Ambulance teams
  - Save as many people as possible
  - Recover and preserve site and its infrastructure (buildings, communications, etc.)

Robocup Rescue - Elements

- General capabilities
  - Mobility, communication, partial situation awareness at higher reasoning levels
- Specialisations
  - Ambulance teams rescue civilians from rubble and transport to safety
  - Fire brigades extinguishing fires
  - Police forces for traffic ordering, general order and safety
- Our Target: Provide automated decision and information support for *time critical* and potentially *irreversible* decisions.

Task 1: ambulance allocation

- Multiple ambulance services
  - Business oriented operation
  - Competition for government funds and public opinion
- Given several locations that require medical assistance, how many ambulances from which firm will go to which location?

Task 2: police patrols

- Low ratio of police force vs. operative requirements
- How frequently and with what qualitative force to patrol an area?
- How many safe routs vs their quality can the given police force support? Can and should it be adapted over time?

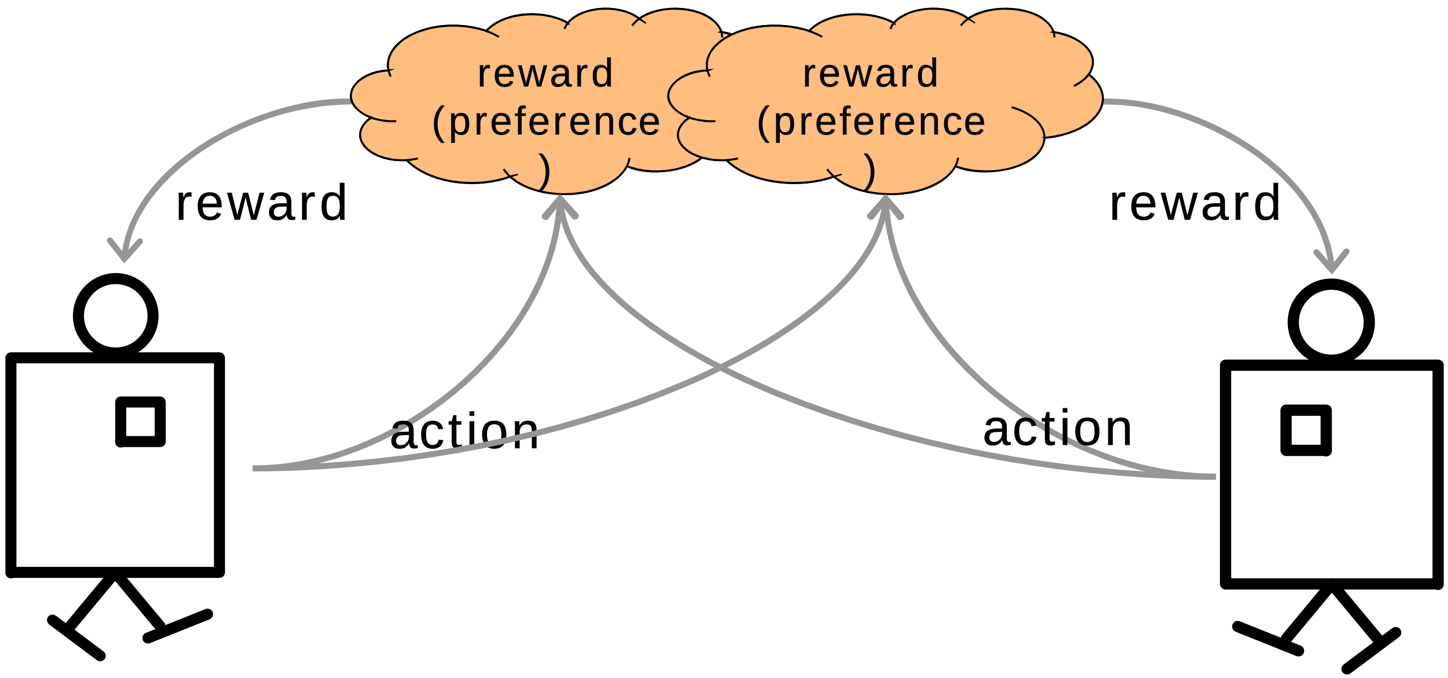
Task 3: firefighters

- Maintain effort toward saving the building or draw back and minimise the spread of fire?
- Concentrate on a multitude of smaller fires or allow controlled unification and deal with only one location?
  - Will transportation routs be endangered?
  - Are there still civilians evacuating from the area/building?
- Push through the fire to victims or save the fire crew and pull out?
  - If multiple crews are on site, which one goes? When?

Roadmap

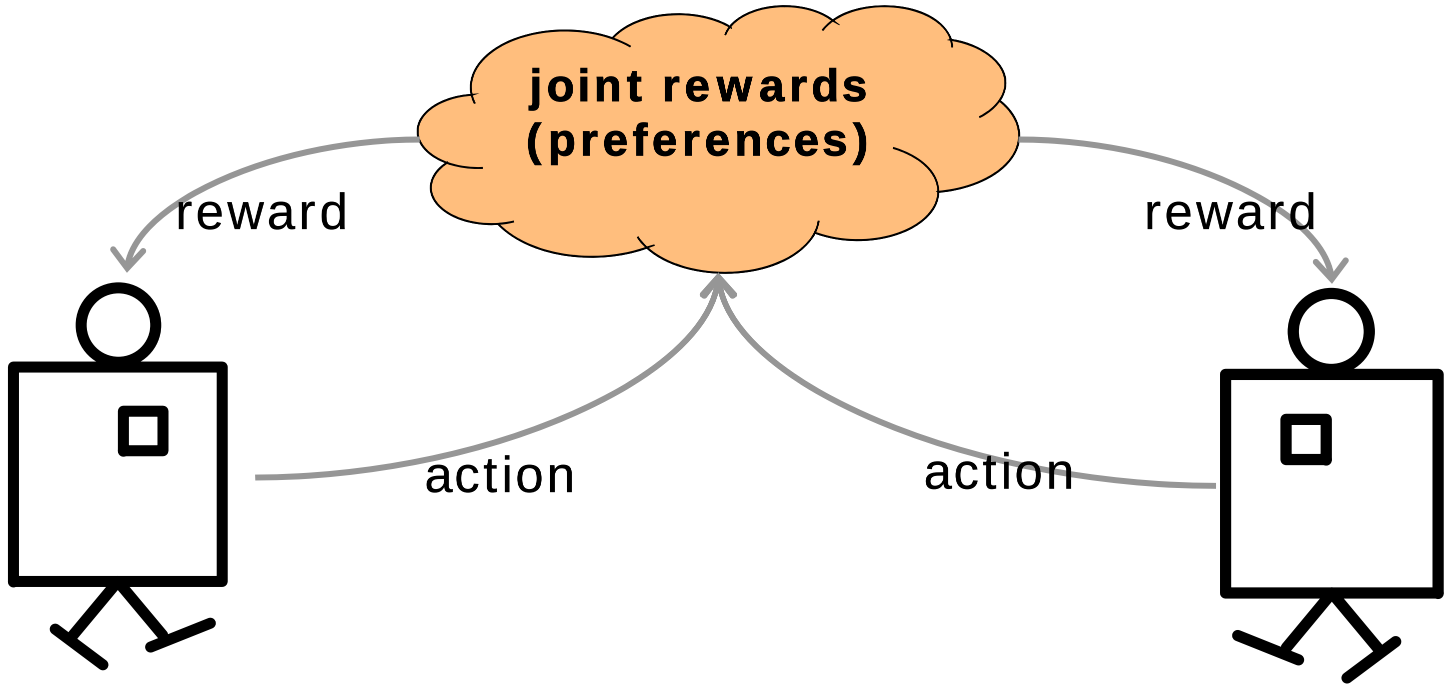
- Driving applications
  - search and rescue
- Multiagent decision making
  - description, requirements, complexity
- Game theory
  - classroom game
  - repeated strategic and Bayesian games
  - fictitious play and regret matching
- Stochastic games
  - DEC-MDP and its specializations
- Partially observable stochastic games
  - I-POMDP framework
  - I-POMDP solution techniques
  - I-POMDP applications
  - DEC-POMDP
- Uncertainty utilization
  - TTD-MDP, Multiagent EMT

Multiagent decision problem



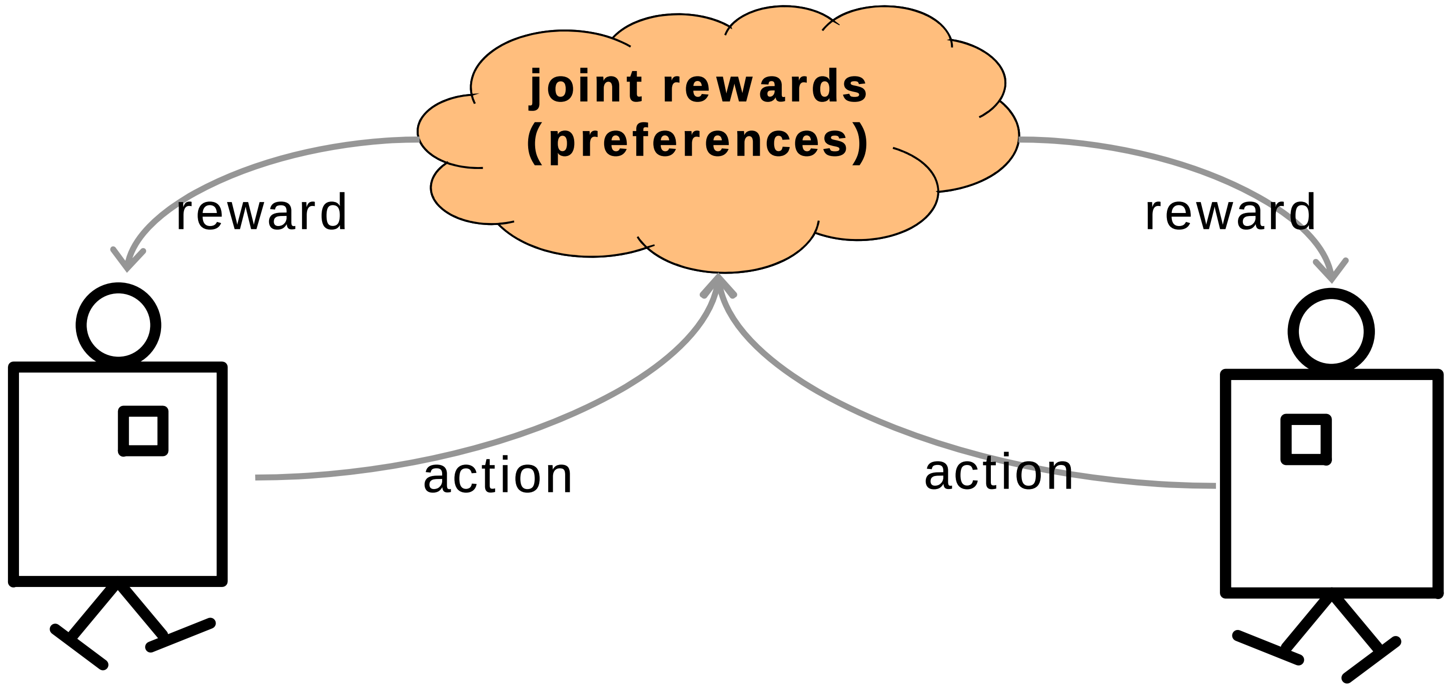
Each agent optimizes its rewards

Multiagent decision problem



Each agent optimizes its rewards

Multiagent decision problem

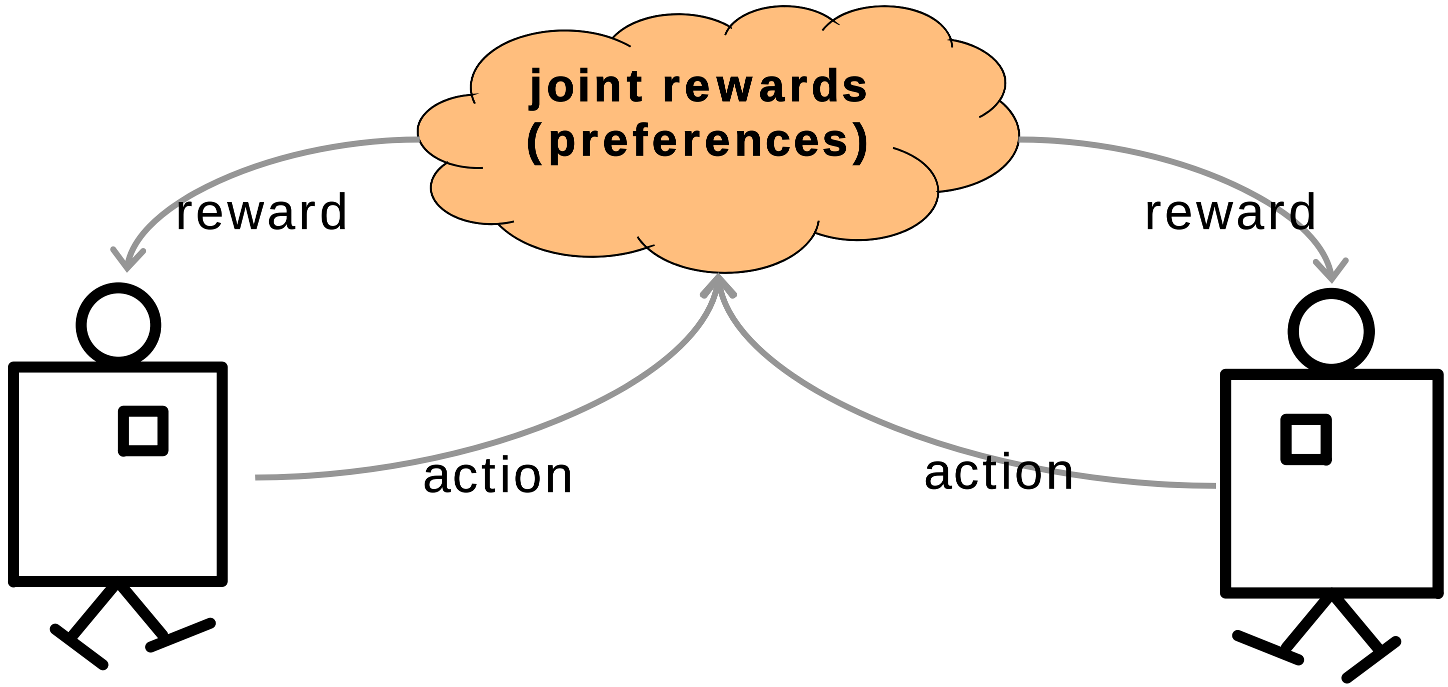


Each agent optimizes rewards

Single interaction *(game)*

Strategy: $\Delta(A)$

Multiagent decision problem

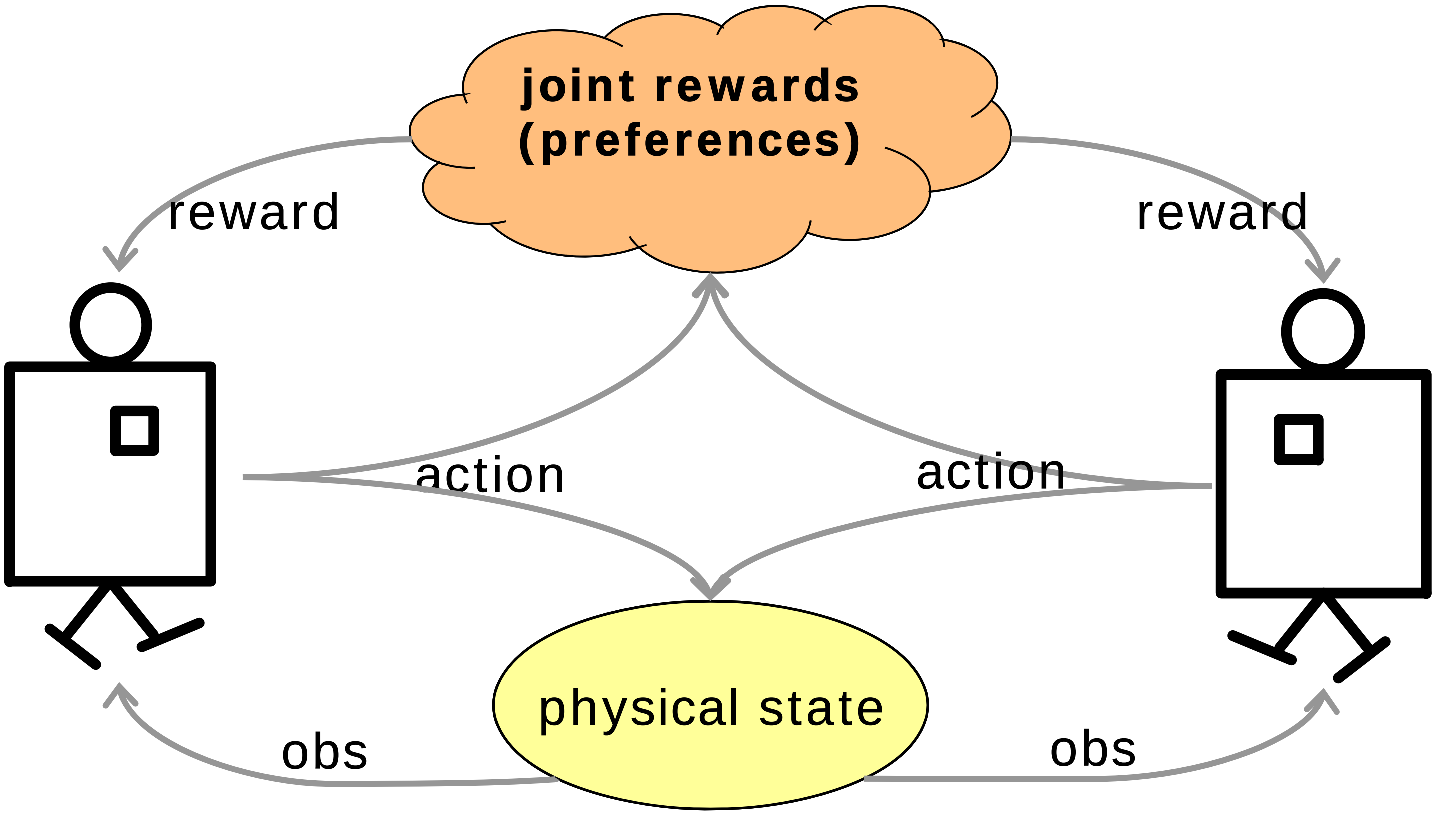


Each agent optimizes rewards

Repeated interactions

Strategy: History of observations $\rightarrow \Delta(A)$

Multiagent decision problem

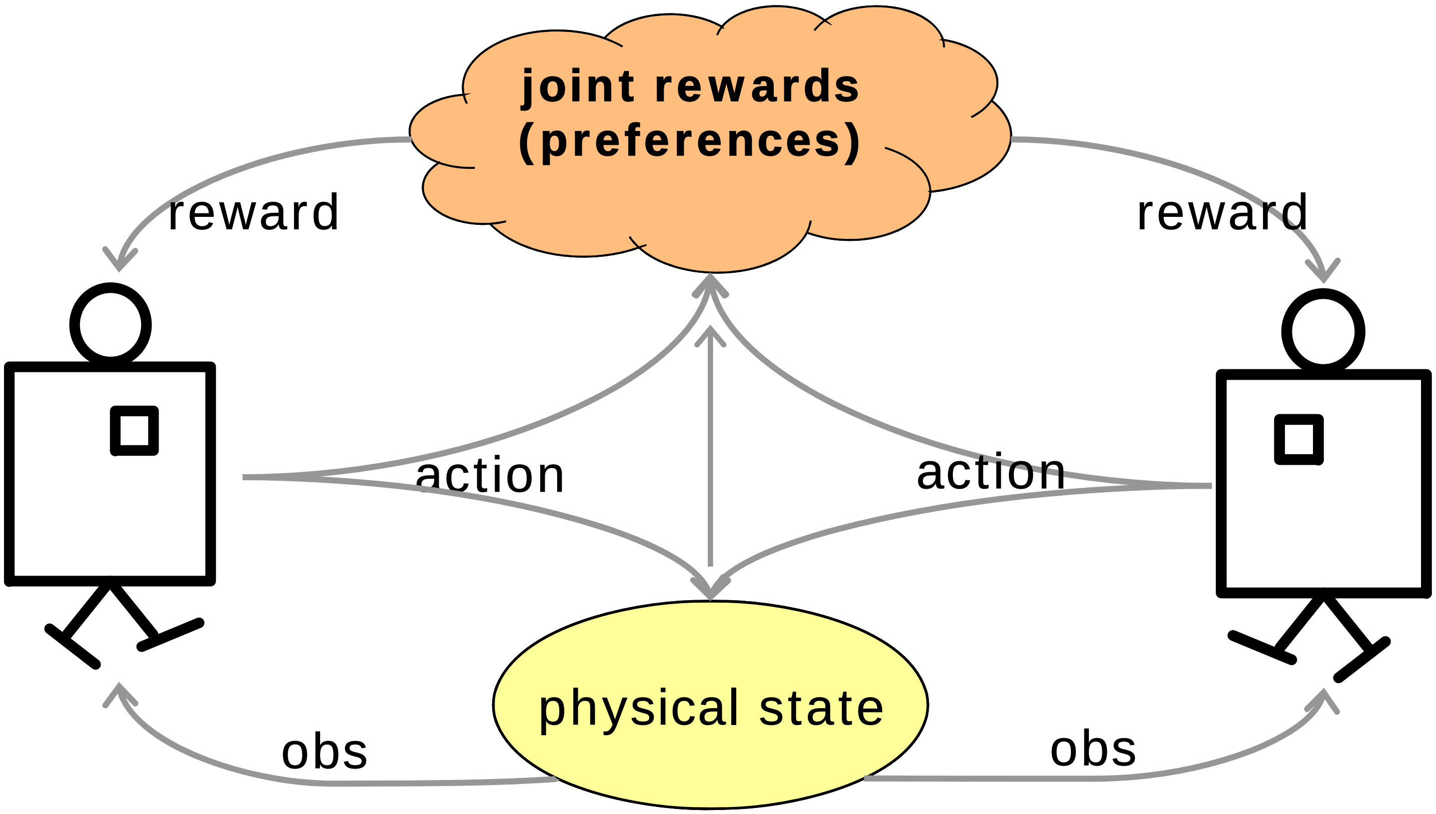


Each agent optimizes rewards

Repeated interactions

Strategy: History of observations $\rightarrow \Delta(A)$

Multiagent decision problem



Each agent optimizes rewards

Repeated interactions

Strategy: History of observations $\rightarrow \Delta(A)$

Dimensions of interaction

- Single or Extended
  - Strategies in extended interactions may be different
    - Extended: Finite or infinite interactions

- Cooperative or Non-cooperative

Dimensions of interaction

- Joint reward or Joint reward and state
  - State is dynamic, influenced by actions
  - State may influence rewards as well
- Perfect or Incomplete information about others

Predictive and epistemological requirements of solution

- In order to maximize rewards, predict actions of others
  - Common knowledge of rationality
    - All agents are rational; All know that all are rational; All know that all know that all are rational; ...
  - Common and perfect knowledge of rewards
    - All know others' rewards; All know that all know others' rewards; ...
  - Common and partial knowledge of rewards
    - Probability distribution over possible rewards is common knowledge

Predictive and epistemological requirements of solution

Epistemological requirements for rational behavior are strict!

Models of interactions (first glance)

Single and repeated interactions with joint rewards are the focus of traditional game theory

Interactions involving joint state and reward are the focus of decision theory inspired approaches to game theory. These generally include extensions of single agent decision-theoretic models to multiagent settings

Other applications

- Robotics
  - Planetary exploration

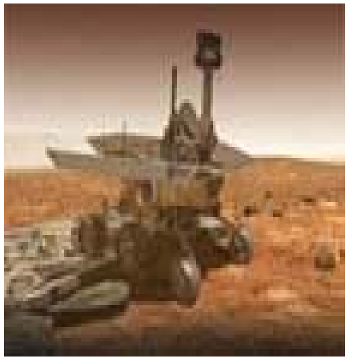
    Surface mapping by rovers

    Coordinate to explore pre-defined region optimally

    Uncertainty due to sensors
  - Robot soccer

    Coordinate with teammates and deceive opponents

    Anticipate and track others' actions



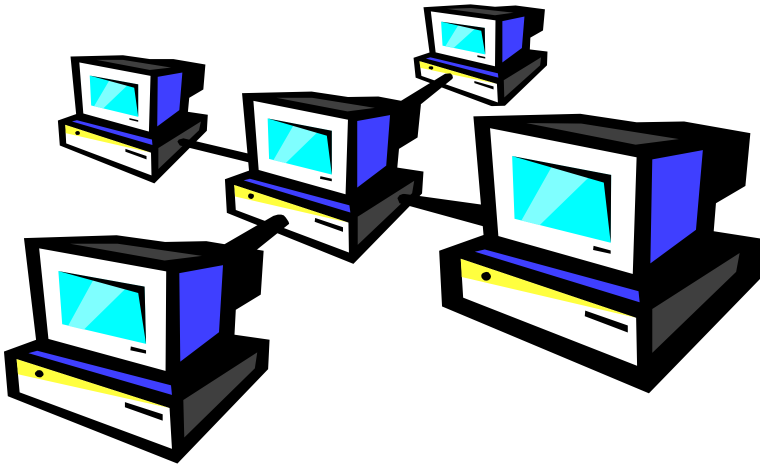
Spirit

Opportunity

RoboCup Competition

Other applications

- Defense
  - Coordinate UAV movements in battlefields
    - Exact “ground situation” unknown
  - Coordinate anti-air defense units
- Distributed Systems
  - Networked Systems
    - Packet routing
  - Sensor networks



Roadmap

- Driving applications
  - search and rescue
- Multiagent decision making
  - description, requirements, complexity
- Game theory
  - classroom game
  - repeated strategic and Bayesian games
  - fictitious play and regret matching
- Stochastic games
  - DEC-MDPs and its specializations
- Partially observable stochastic games
  - I-POMDP framework
  - I-POMDP solution techniques
  - I-POMDP applications
  - Dec-POMDP
- Uncertainty Utilization
  - TTD-MDP, Multiagent EMT

Classroom game: Prisoner's dilemma

- Instructions

We are going to play a card game in which everybody will be matched with someone in the room. I will now give each of you a pair of playing cards, one red card (♥ or ♦) and one black card (♠ or ♣). The numbers or faces on the cards will not matter, just the color. You will be asked to play one of these cards by holding it to your chest . Your earnings are determined by the card that you play and by the card played by the person matched with you.

If you play your red card, then your earnings will increase by $2, and the earnings of the person matched with you will not change. If you play your black card, your earnings do not change and the earnings of the person matched with you go up by $3. If you each play your red card, you will each earn $2. If you each play the black card, you will each earn $3. If you play your black card and the other person plays his or her red card, then you earn zero and the other person earns the $5. If you play red and the other person plays black, you earn the $5, and the other person earns zero. All earnings are hypothetical. After you choose which card to play, hold it to your chest. We then tell you who you are matched with, and you can each reveal the card that you played. Record your earnings in the space below. To make this easier, please write your name: ____________________ .

To begin: Would the people in the row that I designate please choose which card to play and write the color (R or B) in the first column. Show that you have made your decision by picking up the card you want to play and holding it to your chest. Everyone finished? Now, I will pair you with another person, ask you to reveal your choice, and calculate your earnings. Remember to keep track of earnings in the space provided below. Finally, please note that in period 2 you will be matched with a different person, and payoffs will change. In period 3 you will be matched with a different person and payoffs change again, but you get to play with him/her in the last three periods.

Classroom game: Prisoner's dilemma

Your payoff table

Period	Your card (R or B)	Other's card (R or B)	Your earnings
1			
2			
3			
4			
5			

Classroom game: Prisoner's dilemma

Payoff table for Period 1

Player II

Player I

	black	red
black	3,3	0,5
red	5,0	2,2

Payoff table for Period 2

Player II

Player I

	black	red
black	8,8	0,10
red	10,0	2,2

Game in Normal Form

- Defined by a tuple $< I, \{A_i\}_{i \in I}, \{R_i\}_{i \in I} >$
  - I is the set of players, usually $I = \{1, ..., n\}$
  - A_i is the set of actions (*pure strategies*) available to player i.
    - Space of pure strategy profiles $A = \bigotimes_{i \in I} A_i$
    - Let $a = (a_i, a_{-i}) \in A$. Where $a_i \in A_i$ is the action prescribed to agent i, and $a_{-i} \in \bigotimes_{j \in I \setminus \{i\}} A_j = A_{-i}$ portion of profile adopted by other agents.
  - $R_i : A \to \mathcal{R}$ is the reward (*utility*) of the player i, given that players *simultaneously* play their actions
- Each agent *rationally* seeks to maximise its utility

Why game is a game?

- Is there a **guarantee** of utility if I don’t know how others act?
- If I know how others act, how should I?
- If the game is to be repeated, should I act differently?

Guarantees

- “Enemy assumption”: A player assumes that all others collude against it.
  - Essentially a zero sum game
    - $I = 1, 2$, and $R_1 = -R_2$.
  - Guarantee is $\max_{a_1 \in A_1} \min_{a_2 \in A_2} R_1(a_1, a_2)$
- Simplest example: Fire station location

Guarantees: example

- Two plants A and B build a new private fire station
  - Where should it be located?
- Assume fires are deliberate, then time of arrival dictates utility for the Fire Brigade:

		Fire at		
		A	A and B	B
Station	near A	0	-1	-1
	middle	-0.5	-0.5	-0.5
	near B	-1	-1	0

- Minimax value is -0.5 and minimax strategy is *middle*

Equilibria

- Given a partial profile $a_{-i} \in A_{-i}$ the action choice of agents except $i \in I$.
- a_i^* is a best response of agent $i \in I$ to a_{-i} if
  $a_i^* \in \arg\max_{a_i \in A_i} R_i(a_i, a_{-i})$
- A *strategy profile* (joint action) $a \in A$ is a pure Nash equilibria if for all $i \in I$ a_i is a best response to a_{-i}.

Equilibria: example

- Two plants A and B build a new private fire station. Where should it be located?
- Assume fires are deliberate, then time of arrival dictates utility for the Fire Brigade:

	A	A and B	B
near A	0	-1	-1
middle	-0.5	-0.5	-0.5
near B	-1	-1	0

- The pair (*A and B*, *middle*) is a pure Nash equilibria

Non-existence of pure Nash

- Police sends patrols to plant A and plant B to try and catch the saboteurs.
- Utility is determined by the similarity of actions:

	A	B
A	1	-1
B	-1	1

- It is easy to see that no pair $(a_{police}, a_{saboteur})$ is an equilibrium profile.
- Intuition: Surprise factor by randomisation

Mixed profile

- *Mixed strategy* of an agent $i \in I$ is a probability distribution π_i over A_i, where $\pi(a_i)$ is the probability of selecting action a_i.
- Denote Δ_i the set of all probability distributions over A_i. *Mixed strategy profile* (joint mixed strategy) is a distribution $\pi = (\pi_i, \pi_{-i}) \in \bigotimes_{i \in I} \Delta_i$.
  - $\pi(a) = \prod_{i \in I} \pi_i(a_i)$ is the probability that agents will jointly select pure profile $a \in A$.
  - *Expected utility* is then $E_\pi[R_i] = \sum_{a \in A} \pi(a) R_i(a)$

Mixed Nash equilibrium

- Given partial mixed profile π_{-i}. π_i^* is a best response mixed strategy if $\pi_i^* \in \arg\max_{\pi_i \in \Delta_i} E_{(\pi_i, \pi_{-i})}[R_i]$
- A complete mixed profile π is in *mixed Nash equilibrium* if for all $i \in I$, π_i is a best response to π_{-i}.
- For the police patrol example equally probable choice is an equilibrium.

Example

- Two incidents occur in two distinct and remote locations
- Both require attendance by two ambulance teams
- The utilities are

	L_1	L_2
L_1	(2,1)	(-1,-1)
L_2	(-1,-1)	(1,2)

- Problem:
  - Two pure Nash exist (L_1, L_1) and (L_2, L_2)
  - Victims at one location are doomed.
  - Can we improve their chances?

Example (cont)

- Shouldn't mixed Nash do just that?
  - Symmetric mixed Nash exists with each ambulance randomly selecting a location.
  - Expected payoff is $1/4$!! Half of the times the ambulances will miss each other and all victims will perish!

Correlated Nash

- Let P be a joint distribution over the joint profiles A.
- P is a correlated equilibrium if for any agent $i \in I$ holds for all $a_i, \bar{a}_i \in A_i$ that

$$\sum_{a_{-i} \in A_{-i}} P(a_i, a_{-i})(R_i(a_i, a_{-i}) - R_i(\bar{a}_i, a_{-i})) \geq 0$$

- Entire profile is sampled, not composed of random samples
  - Requires a correlated random source
  - E.g. in England: weather at 09:00

Example revisited

	L_1	L_2
L_1	(2,1)	(-1,-1)
L_2	(-1,-1)	(1,2)

- Equilibrium: $P((L_1, L_1)) = P((L_2, L_2)) = \frac{1}{2}$.
- Expected utility is equal for both ambulance teams $\frac{3}{2}$
- Both incidents have equal chance at the treatment

Sad example

- Ambulances are independent business services
  - Cost driven and competitive
- Government funds:
  - Distributed in proportion to saved lives
  - Recognition for success in major events
- Scenario:
  - Two ambulance services
  - Three events: two are minor one major
    - Minor events are local to the services
    - Major event necessitates both services to handle

Sad example (cont)

- Assume that total government funds are 4 units
- If the major event is handled extra 2 units are allocated
- The utilities can be summarised by:

	Major	Minor
Major	(3,3)	(0,4)
Minor	(4,0)	(2,2)

- Problem: It is always best to handle the minor event.
- But in real life they do concentrate on major events. Why?

Repeated games

- Ambulance services “play” this game repeatedly.
  - Long term accumulation of utility
  - For infinite repetition discounting by $\gamma < 1$ or averaging of a single repetition utility, r_i^t, are used.

$$\sum_{t=1}^{\infty} \gamma^t r_i^t \text{ or } \lim_{T \to \infty} \frac{1}{T} \sum_{t=1}^{T} r_i^t$$

- Sequences of actions (or rules composing them) are considered
  - Behaviour rules producing action sequences are termed *policy*
  - In presence of memory new possibilities occur: trust, revenge, reciprocity, etc.

Happy example

- Consider again:

	Major	Minor
Major	(3,3)	(0,4)
Minor	(4,0)	(2,2)

- Assume the following *tit-for-tat* policy:
  - At first attempt to choose “Major”
  - Then mimic the previous action of the other agent
- It is easy to see that TFT is an equilibrium for infinite utility accumulation, and that (Major, Major) is infinitely repeated.

Bayesian games

- Relax the assumption of perfect knowledge of agents' rewards

- Type system
  - Agent's type: Encompasses private information relevant to the agent's behavior
  - Joint probability distribution over types, which is common knowledge

Bayesian games

In Harsanyi's own words:

". . . we can regard the attribute vector c_i *as representing certain physical, social, and psychological attributes of player i himself in that it summarizes some crucial parameters of player i's own payoff function* U_i *as well as main parameters of his beliefs about his social and physical environment . . ."*

Bayesian games – Example

Police Patrol / Criminals

	Enter	Stay out
Enter	0,-1	2,0
Stay out	2,1	3,0

Policing is weak

	Enter	Stay out
Enter	1.5,-1	3.5,0
Stay out	2,1	3,0

Policing is strong

Type space: $\Theta_{Police} = \{R_{Weak}, R_{Strong}\}$

Bayesian games – Example

Criminals

Police Patrol		Enter	Stay out
	Enter	0,-1	2,0
	Stay out	2,1	3,0

Policing is weak

Criminals

	Enter	Stay out
Enter	1.5,-1	3.5,0
Stay out	2,1	3,0

Policing is strong

Let p be the probability that the police is weak

	Enter	Stay out
Enter, Enter	1.5(1-p),-1	2p+3.5(1-p),0
Enter, Stay out	2(1-p), -p+(1-p)	2p + 3(1-p),0
Stay out, Enter	2p + 1.5(1-p), p – (1-p)	3p + 3.5(1-p),0
Stay out, Stay out	2,1	3,0

Bayesian games – Example

Criminals (Police Patrol as row player)

	Enter	**Stay out**
Enter	0,-1	2,0
Stay out	2,1	3,0

Policing is weak

Criminals (Police Patrol as row player)

	Enter	**Stay out**
Enter	1.5,-1	3.5,0
Stay out	2,1	3,0

Policing is strong

For all p ≥ 0, (Enter, Enter) and (Enter, Stay out) is dominated

	Enter	**Stay out**
Enter, Enter	1.5(1-p),-1	2p+3.5(1-p),0
Enter, Stay out	2(1-p), -p+(1-p)	2p + 3(1-p),0
Stay out, Enter	2p + 1.5(1-p), p – (1-p)	3p + 3.5(1-p),0
Stay out, Stay out	2,1	3,0

Bayesian games – Example

Criminals

Police Patrol

	Enter	**Stay out**
Enter	0,-1	2,0
Stay out	2,1	3,0

Policing is weak

Criminals

	Enter	**Stay out**
Enter	1.5,-1	3.5,0
Stay out	2,1	3,0

Policing is strong

For all $p \geq 0$, (Enter, Enter) and (Enter, Stay out) is dominated

so the games collapses into:

	Enter	**Stay out**
Stay out, Enter	2p + 1.5(1-p), p – (1-p)	3p + 3.5(1-p),0
Stay out, Stay out	2,1	3,0

Bayesian games – Example

Criminals (Police Patrol rows)

	Enter	**Stay out**
Enter	0,-1	2,0
Stay out	2,1	3,0

Policing is weak

Criminals (Police Patrol rows)

	Enter	**Stay out**
Enter	1.5,-1	3.5,0
Stay out	2,1	3,0

Policing is strong

	Enter	**Stay out**
Stay out, Enter	1.5 + 0.5p, 2p -1	3.5 – 0.5p, 0
Stay out, Stay out	2,1	3,0

For p > 0.5, Enter is a dominating action for the criminal and {(Stay out, Stay out),Enter} is a Nash equilibrium

For p ≤ 0.5, {(Stay out, Stay out), Enter} and {(Stay out, Enter), Stay out} are Nash equilibria

Bayesian games – Example

Criminals

Police Patrol	Enter	Stay out
Enter	0,-1	2,0
Stay out	2,1	3,0

Policing is weak

Criminals

	Enter	Stay out
Enter	1.5,-1	3.5,0
Stay out	2,1	3,0

Policing is strong

	Enter	Stay out
Stay out, Enter	1.5 + 0.5p, 2p -1	3.5 – 0.5p, 0
Stay out, Stay out	2,1	3,0

EU(Stay out, Enter) = $(1.5+0.5p)x+(1-x)(3.5-0.5p)=3.5-0.5p+x(p-2)$

EU(Stay out, Stay out) = $2x+3(1-x)=3-x$

Police is indifferent when $3.5p-0.5p+x(p-2)=3-x$

$$x=1/2$$

Bayesian games – Example

Policing is weak (Police Patrol rows, Criminals columns)

	Enter	Stay out
Enter	0,-1	2,0
Stay out	2,1	3,0

Policing is strong (Police Patrol rows, Criminals columns)

	Enter	Stay out
Enter	1.5,-1	3.5,0
Stay out	2,1	3,0

	Enter	**Stay out**
Stay out, Enter	1.5 + 0.5p, 2p -1	3.5 – 0.5p, 0
Stay out, Stay out	2,1	3,0

EU(Enter) = $(2p-1)y+1(1-y)=(2p-2)y+1$

EU(Stay out) = 0

Criminal is indifferent when $1+y(2p-2)=0$

$$y=1/2(1-p)$$

Bayesian games – Example

Criminals (Police Patrol rows)

	Enter	Stay out
Enter	0,-1	2,0
Stay out	2,1	3,0

Policing is weak

Criminals (Police Patrol rows)

	Enter	Stay out
Enter	1.5,-1	3.5,0
Stay out	2,1	3,0

Policing is strong

3 Bayesian Nash equilibria

{Stay out, Enter} for any p

{(Stay out, Enter), Stay out} if $p \le 0.5$

$$\left\{\left\langle \frac{1}{2(1-p)}, \frac{1-2p}{2(1-p)} \right\rangle, \left\langle \frac{1}{2}, \frac{1}{2} \right\rangle\right\} \quad \text{if } p \le 0.5$$

Bayesian games

In general, a strategy profile $\{\pi_i, \pi_j\}$ is a Bayesian Nash equilibrium if for each agent i and its type, θ_i ,

$$\pi_i(\theta_i) = \underset{a_i \in A_i}{\operatorname{argmax}} \sum_{\theta_j \in \Theta_j} R_{\theta_i}(a_i, \pi_j(\theta_j)) p(\theta_i, \theta_j)$$

Repeated games

In game theory, two models of decision-making in repeated interactions are popular:

- Fictitious play
- Rational learning

Repeated games – Fictitious play

- Simplest model of decision-making in repeated games
- At each stage, an agent ascribes a mixed strategy to the other, $b_i^t(a_j)$

  Other agent is assumed to act according to this mixed strategy
- The strategy is computed as follows:

$$F^t(a_j) = F^{t-1}(a_j) + \begin{cases} 1 & if \;\; a_j^{t-1} = a_j \\ 0 & if \;\; a_j^{t-1} \neq a_j \end{cases}$$

$$b_i^t(a_j) = \frac{F^t(a_j)}{\sum_{a_j \in A_j} F^t(a_j)}$$

Maintain a frequency count of previous actions

- Agent computes its best response to the mixed strategy of other

Fictitious play – Example

		Police patrol 2	
		Enter	Stay out
Police patrol 1	Enter	0,0	1,1
	Stay out	1,1	0,0

coordination game

2 pure strategy Nash equilibria and one mixed strategy Nash equilibrium

{Enter, Stay out} {Stay out, Enter}

$$\{\langle 0.5, 0.5 \rangle, \langle 0.5, 0.5 \rangle\}$$

Fictitious play – Example

Police patrol 2

Police patrol 1		Enter	Stay out
	Enter	0,0	1,1
	Stay out	1,1	0,0

coordination game

Round	Patrol 1	Patrol 2	1's belief	2's belief
0			(1,0.5)	(1,0.5)
1	Stay out	Stay out	(1,1.5)	(1,1.5)
2	Enter	Enter	(2,1.5)	(2,1.5)
3	Stay out	Stay out	(2,2.5)	(2,2.5)
4	Enter	Enter	(3,2.5)	(3,2.5)
…	…	…	…	…

Fictitious play – Example

Police patrol 2

Police patrol 1	Enter	Stay out
Enter	0,0	1,1
Stay out	1,1	0,0

coordination game

Round	Patrol 1	Patrol 2	1's belief	2's belief
0			(1,0.5)	(1,0.5)
1	Stay out	Stay out	(1,1.5)	(1,1.5)
2	Enter	Enter	(2,1.5)	(2,1.5)
3	Stay out	Stay out	(2,2.5)	(2,2.5)
4	Enter	Enter	(3,2.5)	(3,2.5)
…	…	…	…	…

Nash equilibrium!

Fictitious play

Interesting properties

- If an action vector is a strict Nash equilibrium of a stage game, it is the steady state of fictitious play in the repeated game
- If the empirical distribution of each agent's strategies converges in fictitious play, then it converges to a Nash equilibrium
- Fictitious play in repeated games converges if the game is a 2x2 game with generic payoffs or is a zero-sum game

Regret: Emotion

- Regret is a realisation of a missed opportunity
  - Action choice suboptimal in retrospective
- In common practise people use it to improve their future decisions.
  - The current strategy is modified, to *match* the optimal retrospective
- Given that we play a game repeatedly, can this concept be utilised to rationally improve utility accumulation?

Regret: Formalisation

- Given a game $G =< I, \{A_i\}, \{R_i\} >$
  - History of play at time t is a sequence of joint profiles $h^t = (a^\tau)_{\tau=1}^t \in \bigotimes_{\tau=1}^t A$, where $A = \bigotimes_{i \in I} A_i$
- Consider the following definition:
  - Lost opportunity:
    $$L_i^t(a_i^*, a_i') = \frac{1}{t} \sum_{\tau \leq t : a_i^\tau = a_i^*} [R^i(a_i', a_{-i}^\tau) - R_i(a^\tau)]$$
  - Disappointment (regret):
    $$D_i^t(a_i^*, a_i') = \max\{L_i^t(a_i^*, a_i'), 0\}$$

Regret Matching

- Assume at time t agent i took action a_i^*
- Denote $p_i^{t+1}(a_i)$ the probability of agent i choosing action a_i at time step $t+1$
- To *match regret* one should:
  - Compute the regret matrix $D_i^t(a_i^*, a_i')$
  - At time $t+1$ the agent should select its action w.r.t.:

$$p_i^{t+1}(a_i) = \begin{cases} \frac{1}{\mu} D_i^t(a_i^*, a_i) & a_i \neq a_i^* \\ 1 - \sum\limits_{a_i' \in A_i} p_i^{t+1}(a_i') & otherwise \end{cases}$$

Example

	l_1	l_2
l_1	(2,1)	(-1,-1)
l_2	(-1,-1)	(1,2)

$$L_i^t(a_i^*, a_i') = \frac{1}{t} \sum_{\tau \le t : a_i^\tau = a_i^*} [R^i(a_i', a_{-i}^\tau) - R_i(a^\tau)]$$

$$D_i^t(a_i^*, a_i') = \max\{L_i^t(a_i^*, a_i'), 0\}$$

t	a_1	a_2	R	D_1		D_2		$p_1(l_1)$	$p_2(l_1)$
				$l_1 \to l_2$	$l_2 \to l_1$	$l_1 \to l_2$	$l_2 \to l_1$		
1	l_2	l_1	(-1,-1)	0	3	3	0	$\frac{3}{10}$	$\frac{7}{10}$
2	l_2	l_2	(1,2)	0	1	$\frac{3}{2}$	0	$\frac{1}{10}$	0
3	l_1	l_2	(-1,-1)	$\frac{2}{3}$	$\frac{1}{3}$	1	0	$\frac{1}{15}$	0
4	l_2	l_2	(1,2)	$\frac{2}{4}$	0	$\frac{3}{4}$	0	0	0

Regret: behaviour convergence

- Let $h^t = (a^\tau)_{\tau=1}^t$ be the history of play up to time t.
  - Empirical distribution of joint profiles is:
  $$z^t(a) = \frac{1}{t} \left|\{\tau \leq t : a^\tau = a\}\right|$$
- **Theorem (Hart-MasColell, Hart)**:
  If every agent matches its regret, then the sequence z^t converges almost surely to the set of correlated equilibria.

Stochastic process

- *Stochastic process* is a sequence of random variables $X_1, \ldots, X_t, \ldots$.
  - Completely described by the conditional probability $Pr(X_t = x_t | X_1 = x_1, ..., X_{t-1} = x_{t-1})$
  - For Markovian process $Pr(X_t = x_t | X_1 = x_1, ..., X_{t-1} = x_{t-1}) = Pr(X_t = x_t | X_{t-1} = x_{t-1}) = T_t(x_t | x_{t-1})$
  - Process is homogeneous if exists *transition matrix* $T(x'|x) = Pr(X_t = x' | X_{t-1} = x)$ for all t

Regret Matching: Markov chain

- Random choice of next action given the current one
- Transition matrix for agent i at time t:

$$T_i^{t+1}(a_i'|a_i^*) = \begin{cases} \frac{1}{\mu} D_i^t(a_i^*, a_i) & a_i \neq a_i^* \\ 1 - \sum_{a_i' \in A_i} T_i^{t+1}(a_i'|a_i^*) & otherwise \end{cases}$$

- Exists stationary probability q_i^{t+1} so that $q^{t+1} = T_i^t q_i^{t+1}$

Regret Matching: Markov chain

- **Theorem (Hart-MasColell, Hart):**
  - If player i chooses action at time $t+1$ according to q_i^{t+1}, then its regrets $D_i^t(\cdot|\cdot)$ converge to zero
  - If every player adopts the above then z^t converges almost surely to the set of correlated equilibria

Roadmap

- Driving applications
  - search and rescue
- Multiagent decision making
  - description, requirements, complexity
- Game theory
  - classroom game
  - repeated strategic and Bayesian games
  - fictitious play and regret matching
- Stochastic games
  - DEC-MDP and its specializations
- Partially observable stochastic games
  - I-POMDP framework
  - I-POMDP solution techniques
  - I-POMDP applications
  - Dec-POMDP
- Uncertainty Utilization
  - TTD-MDP, Multiagent EMT

Stochastic (Markov) Games

- Regret Matching had some very interesting elements
  - Agent concentrates exclusively on its regret. It is essentially the *state* of the world.
  - Agent's utility was essentially determined by the change in the state
  - Probabilistically selected *actions* w.r.t. state
  - Action sequence was generated by a concise rule, or *policy*
- Can this situation be explicitly modelled?
- Can a "game" be described with these properties?

Markovian Environment

- Consider the tuple $< S, s_0, A, T >$
  - S set of agent's world states, with s_0 being the initial one
  - A is the set of actions available to the agent
  - $T : S \times A \times S \rightarrow [0, 1]$ is the transition matrix. $T(s', a, s)$ is the probability that the world will change from state $s \in S$ to state $s' \in S$ if agent performs $a \in A$
- What a rational agent would do with such a setting?

How does it work?

- At time $t = 0$ the world starts at state s_0
- Then decision loop is repeated
  - Agent chooses an action $a_t \in A$
  - Action a_t is applied
  - The world changes its state. s_{t+1} is chosen w.r.t. $T(\cdot|s_t, a_t)$
  - Time step occurs $t \leftarrow t + 1$
- How does an agent choose its action?

Example

- For example the crime rate is weakly responsive to the police presence
- Modelled by a Markovian environment
  - $S = \{high, medium, low\}$ is the crime rate
  - $A = \{large, small\}$ is the police force size

$T(\cdot, a, \cdot)$	$a = large$			$a = small$		
	high	medium	low	high	medium	low
high	0	0.7	0.3	1	0	0
medium	0	0.5	0.5	0.5	0.5	0
low	0	0	1	0.1	0.3	0.6

Markov Decision Problem

- The tuple $< S, s_0, A, T >$ is only the *environment*
- Rational agents needs a performance measure to decide on an action (sequence)
- Markov Decision Problem (MDP) is a tuple $< S, s_0, A, T, r >$
  - Given a utility function $r : S \times A \times S \longrightarrow \mathbf{R}$
  - Utility based performance measure
    - Finite horizon $T < \infty$: $\mathbf{E}\left(\sum_{t=0}^{T} r(s_{t+1}, a_t, s_t)\right)$
    - Infinite horizon $\gamma < 1$: $\mathbf{E}\left(\sum_{t=0}^{\infty} \gamma^t r(s_{t+1}, a_t, s_t)\right)$
    - Infinite Average: $\lim_{T \to \infty} \mathbf{E}\left(\frac{1}{T} \sum_{t=0}^{T} r(s_{t+1}, a_t, s_t)\right)$

Action sequence by policy

- Formally infinite performance measures would require strategies to be infinite sequences of actions
- Instead we define a *policy*
  - Repeatedly applied rule to construct the sequence
  - We'll focus on $\pi : S \rightarrow \Delta(A)$, where $\Delta(A)$ is the space of distributions over A
- Sufficiency of policy space
  - The sufficient statistics set for previous activity is the domain
  - Performance may not be improved by a more complex policy
  - $\pi : S \rightarrow \Delta(A)$ is sufficient for single agent MDPs

How good is a policy?

- Denote $V^\pi(s)$ the utility accumulated by an agent following policy π if the system starts in state s.
$$V^\pi(s) = \sum_a \pi(s,a) \sum_{s'} \left(R(s',a,s) + \gamma V^\pi(s')\right) T(s'|s,a)$$
- Define auxiliary quality of action $Q^\pi(s,a)$
  - Denotes the utility gained by an agent by applying $a \in A$ in state s and then following policy π
$$V^\pi(s) = \sum_a \pi(s,a) Q^\pi(s,a)$$
$$Q^\pi(s,a) = \sum_{s'} \left(R(s',a,s) + \gamma V^\pi(s')\right) T(s'|s,a)$$
- Notice that given π, V^π is the solution to a system of linear equations

Example

- Crime rate model:
  - $S = \{high, medium, low\}$ is the crime rate
  - $A = \{large, small\}$ is the police force size

$T(\cdot, a, \cdot)$	$a = large$			$a = small$		
	high	medium	low	high	medium	low
high	0	0.7	0.3	1	0	0
medium	0	0.5	0.5	0.5	0.5	0
low	0	0	1	0.1	0.3	0.6

- Police chief will receive:
  - A reprimand if the crime rate increases
  - A frown from his neighbour if it remains the same
  - A medal if it drops
  - A bad reputation if he uses too much force

Example

- Crime rate model:
  - $S = \{high, medium, low\}$ is the crime rate
  - $A = \{large, small\}$ is the police force size

$T(\cdot, a, \cdot)$	$a = large$			$a = small$		
	high	medium	low	high	medium	low
high	0	0.7	0.3	1	0	0
medium	0	0.5	0.5	0.5	0.5	0
low	0	0	1	0.1	0.3	0.6

- Police chief utility is:

$R(\cdot, a, \cdot)$	$a = large$			$a = small$		
	high	medium	low	high	medium	low
high	-1.5	0	0	-0.5	1	1
medium	-2	-1.5	0	-1	-0.5	1
low	-2	-2	-1.5	-1	-1	-0.5

Example

- A policy $\pi : S \rightarrow \Delta(A)$ for the chief would be to decide how many people he send out every day with what probability depending on that day's situation.
- Assume that he always send out large force $\pi(s) = (1, 0)$
- Assume also that he likes to say "Tomorrow is another day" and assigns $\gamma = 0.5$
- What would be his benefit?

Example

	$T(\cdot, a = large, \cdot)$			$R(\cdot, a = large, \cdot)$		
	high	medium	low	high	medium	low
high	0	0.7	0.3	-1.5	0	0
medium	0	0.5	0.5	-2	-1.5	0
low	0	0	1	-2	-2	-1.5

$$
\begin{aligned}
V^\pi(s) &= \sum_{s'} (R(s', a, s) + \gamma V^\pi(s')) T(s'|s, a) \\
V^\pi(h) &= 0.0 * (..) + 0.7 * (0.0 + 0.5 V^\pi(m)) + \ldots \\
&\quad 0.3 * (0.0 + 0.5 V^\pi(l)) \\
V^\pi(m) &= 0.0 * (..) + 0.5 * (-1.5 + 0.5 * V^\pi(m)) + \ldots \\
&\quad 0.5 * (0.0 + 0.5 V^\pi(l)) \\
V^\pi(l) &= 0.0 * (..) + 0.0 * (..) + 1.0 * (-1.5 + 0.5 V^\pi(l))
\end{aligned}
$$

Example

	$T(\cdot, a = large, \cdot)$			$R(\cdot, a = large, \cdot)$		
	high	medium	low	high	medium	low
high	0	0.7	0.3	-1.5	0	0
medium	0	0.5	0.5	-2	-1.5	0
low	0	0	1	-2	-2	-1.5

$$
\begin{aligned}
V^\pi(s) &= \sum_{s'} (R(s', a, s) + \gamma V^\pi(s')) T(s'|s, a) \\
V^\pi(h) &= 0.35 V^\pi(m) + 0.15 V^\pi(l) \\
V^\pi(m) &= -0.75 + 0.25 V^\pi(m) + 0.25 V^\pi(l) \\
V^\pi(l) &= -1.5 + 0.5 V^\pi(l)
\end{aligned}
$$

Example

	$T(\cdot, a = large, \cdot)$			$R(\cdot, a = large, \cdot)$		
	high	medium	low	high	medium	low
high	0	0.7	0.3	-1.5	0	0
medium	0	0.5	0.5	-2	-1.5	0
low	0	0	1	-2	-2	-1.5

$$\begin{aligned} V^\pi(h) &= -1.15 \;\; (\max \approx -0.59) \\ V^\pi(m) &= -2 \;\; (\max \approx -1.13646) \\ V^\pi(l) &= -3 \;\; (\max \approx -1.285714) \end{aligned}$$

Optimal policy

- Rational agent would like to find $\pi^* \in \arg\max_\pi V^\pi(s_0)$
- Bellman-Ford Equation:
  - Exists V^* so that:

$$V^*(s) = \max_\pi \sum_a \pi(s,a) \sum_{s'} \left(R(s',a,s) + \gamma V^*(s')\right) T(s'|s,a)$$

  - $V^* = \max_\pi V^\pi$, and exists π^* so that $V^* = V^{\pi^*}$

$$\pi^*(s,\cdot) = \arg\max_{\pi(s,\cdot)} \sum_a \pi(s,a) \sum_{s'} \left(R(s',a,s) + \gamma V^*(s')\right) T(s'|s,a)$$

- But how do we find V^*??

Value Iteration

- Dynamic Programming solution
  - Start from some arbitrary small $V_0(\cdot)$
  - Propagate back in time:

$$V_{t+1}(s) = \max_{\pi} \sum_{a} \pi(s,a) \sum_{s'} \left(R(s',a,s) + \gamma V_t(s')\right) T(s'|s,a)$$

- Propagation step is a γ-contraction mapping
  - Procedure converges to V^*

Policy Iteration

- But we can have an intermediate policy:
  - Start with some arbitrary $Q_0(\cdot, \cdot)$
  - Loop the following:
    - Compute a greedy policy w.r.t. Q_t:
      $$\pi(s, a) = \arg\max_a Q_t(s, a)$$
    - Compute policy value V^π
    - Compute
      $$Q_{t+1}(s, a) = \sum_{s'} \left(R(s', a, s) + \gamma V^\pi(s')\right) T(s'|s, a)$$
- Converges being a contraction mapping as well

Markov games

- State may be subject to effects by more than one agent
- Multiagent Markovian Environment $< S, s_0, \{A_i\}_{i=1}^N, T >$
  - S and $s_0 \in S$ are the state space and initial state
  - A_i is the space of i'th agent actions
  - $T : S \times A \times S \rightarrow [0, 1]$, where $A = \bigotimes A_i$.
    $T(s', a, s)$ is the probability that state will change from s to s' if joint action $a = (a_1, ..., a_N)$ is taken
- Markov Game is then $< S, s_0, \{A_i\}_{i=1}^N, T, \{R_i\}_{i=1}^N >$
  - $R_i : S \times A \rightarrow \mathbf{R}$, where $A = \bigotimes A_i$
  - Usually discount accumulated

Policy profile

- For regular games we had a mixed strategy profile $\pi = (\pi_1, ..., \pi_N)$
  - $\pi(a) = \prod \pi_i(a_i)$
- For Markov games we define a joint policy profile $\pi = (\pi_1, ..., \pi_N)$
  - $\pi(s, a) = \prod \pi_i(s, a_i)$
- Notice that a policy of an individual agent may be "pure"
  - For each $s \in S$ exists a single $a_i \in A_i$ so that $\pi(s, a_i) = 1$

Minimax solution

- For $N = 2$ and $R_1 = -R_2$ we can formulate a minimax solution
  - Let $V(s)$ be expected reward for the optimal policy starting at state $s \in S$
  - Let $Q(s, a_1, a_2)$ the expected reward for the optimal policy if at first agents perform (a_1, a_2)
- Then system of equations holds:
  - $V(s) = \max_{\pi} \min_{a_2} \sum_{a_1 \in A_1} Q(s, a_1, a_2)\pi(a_1)$
  - $Q(s, a_1, a_2) = R(s, a_1, a_2) + \gamma \sum_{s' \in S} T(s', a_1, a_2, s)V(s')$

Equilibrium solution

- Given the estimate of quality $Q(s, a)$ one can define equilibrium
- Policy profile $\pi = (\pi_1, ..., \pi_N)$ is an equilibrium if for any $\pi' = (\pi'_i, \pi_{-i})$
$$\sum_{a \in A} \pi(s, a) Q_i(s, a) \geq \sum_{a \in A} \pi'(s, a) Q_i(s, a)$$

Dec-MDP

- Dec-MDP is a Markov Game with identical utilities
- Models a cooperating (team) group of agents
  - Find a joint policy profile to maximise the (common) utility
  - Notice that the individual sub-policies are executed independently
- Extremely hard to solve!!
- Frequently the state is a composition of:
  - Individual properties of agents
  - Intrinsic agent independent property

TI-Dec-MDP

- Let $< S = \bigotimes_{i=0}^{N} S_i, \{A_i\}, T, R >$ be a Dec-MDP with decomposable state space
- *Transition independence* implies that:
  - Exist $T_i : S_i \times A_i \times S_i \to [0, 1]$ for $1 \leq i \leq N$
  - Exists $T_0 : S_0 \times S_0 \to [0, 1]$
  - For $s' = (s'_1, ..., s'_N), s = (s_1, ..., s_N) \in S$ and $a = (a_1, ..., a_N) \in A$ holds:

$$T(s', a, s) = T_0(s'_0, s_0) \prod_{i=1}^{N} T_i(s'_i, a_i, s_i)$$

- Notice that the utility function R has not been factored.

TI-Dec-MDP (cont)

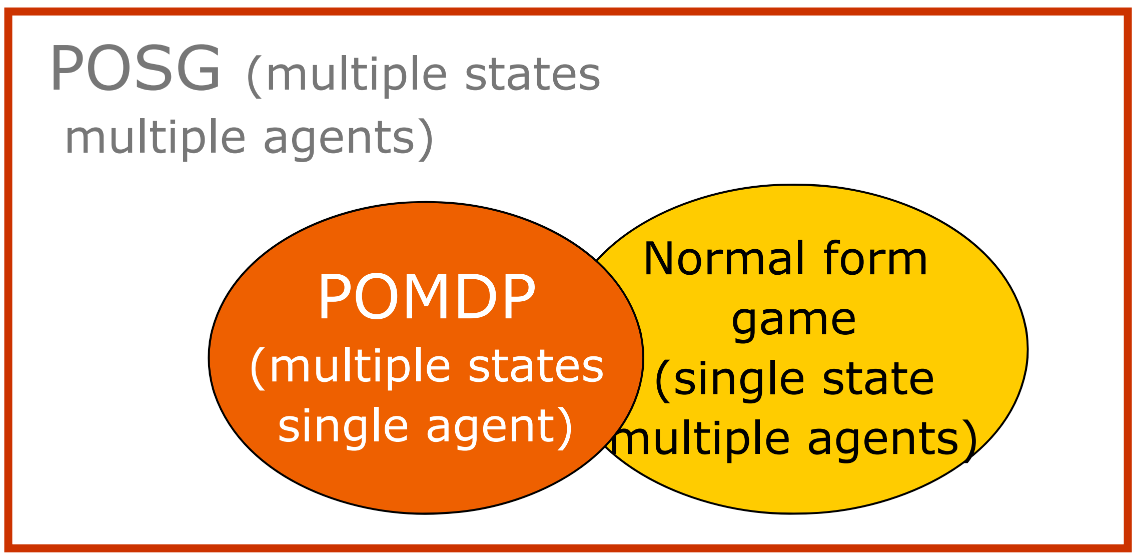
- TI-Dec-MDPs are hard to solve, but exponentially simpler than Dec-MDPs
  - $\pi_i : S \times A \to [0, 1]$ are sufficient policies
  - Solved by a dedicated Coverage Set Algorithm for structured utilities
  - Approximated by an iterative procedure
- Asynchronous policy iteration
  - Begin from an arbitrary policy profile $\pi = (\pi_1, ..., \pi_N)$
  - Each agent i *in its turn*
    - Compute the reduced utility $R_i = E[R]$ w.r.t. π_{-i}
    - Compute optimal policy π_i^* for the resulting MDP $< S_0 \times S_i, A_i, T_0 \cdot T_i, R_i >$
  - Let $\pi \leftarrow (\pi_i^*, \pi_{-i})$

Roadmap

- Driving applications
  - search and rescue
- Multiagent decision making
  - description, requirements, complexity
- Game theory
  - classroom game
  - repeated strategic and Bayesian games
  - fictitious play and regret matching
- Stochastic games
  - DEC-MDPs and its specializations
- Partially observable stochastic games
  - I-POMDP framework
  - I-POMDP solution techniques
  - I-POMDP applications
  - Dec-POMDP
- Uncertainty Utilization
  - TTD-MDP, Multiagent EMT

Partially observable stochastic game

POSGs are a generalization of POMDPs and normal form games to multiple states and multiple agents



Multiagent POMDPs

Multiagent POMDP frameworks generalize POMDPs to multiagent settings

- Decentralized POMDPs (DEC-POMDPs)
  - Objective view of the interaction
    (What should all agents do?)
  - Applicable to team problems
  - Initial beliefs of agents are common knowledge

- Interactive POMDPs (I-POMDPs)
  - Subjective view of the interaction
    (What should a particular agent do?)
  - Applicable to cooperative and non-cooperative problems
  - Beliefs of other agents are unknown

Background: POMDP

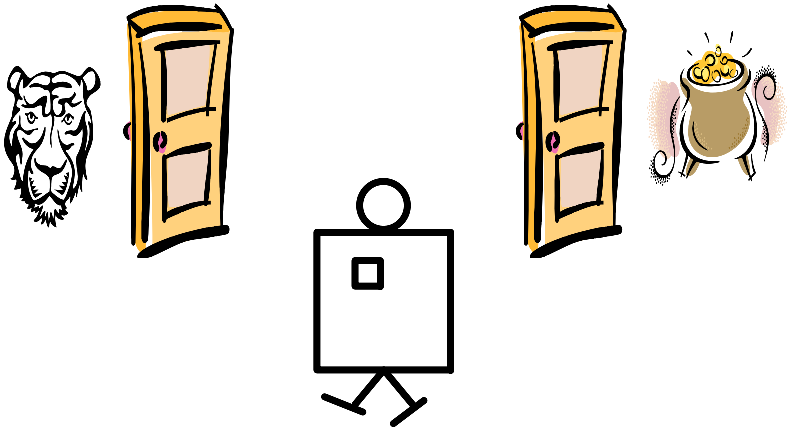
Decision-making in single agent complex domains:
Partially **O**bservable **M**arkov **D**ecision **P**rocess

Single agent Tiger problem (digression from search & rescue)

Task: Maximize collection of gold over a finite or infinite number of steps while avoiding tiger

Tiger emits a growl periodically (GL or GR)

Agent may listen or open doors (L, OL, or OR)



Background: POMDP

- Question 1: How rich should S be?
  Answer: As much as you can

- Question 2: What if other agents are present?

- Problem

"... there is currently no good way to combine game theoretic and POMDP control strategies."

- Russell and Norvig
AI: A Modern Approach, 2nd Ed.

Background: POMDP

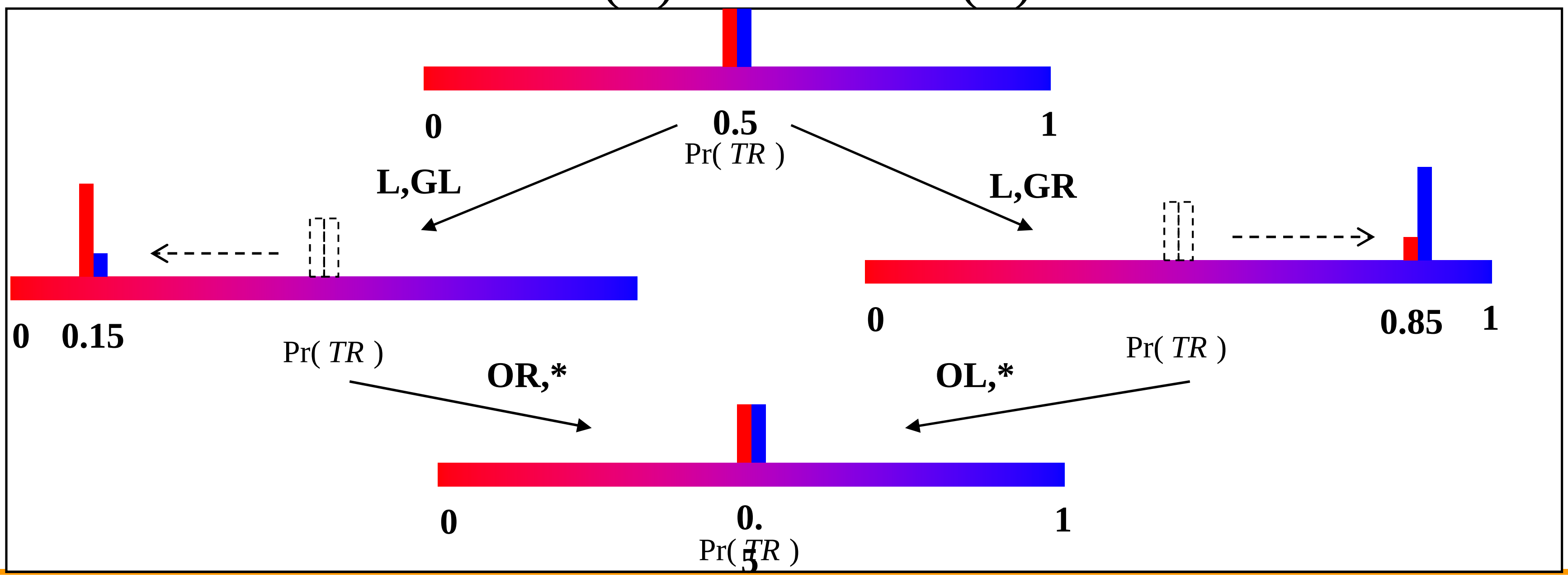
Steps to compute a strategy (policy)

1. Model of the decision making situation:

$$\left\langle S, A_i, \Omega_i, O_i, T_i, R_i, OC_i \right\rangle$$

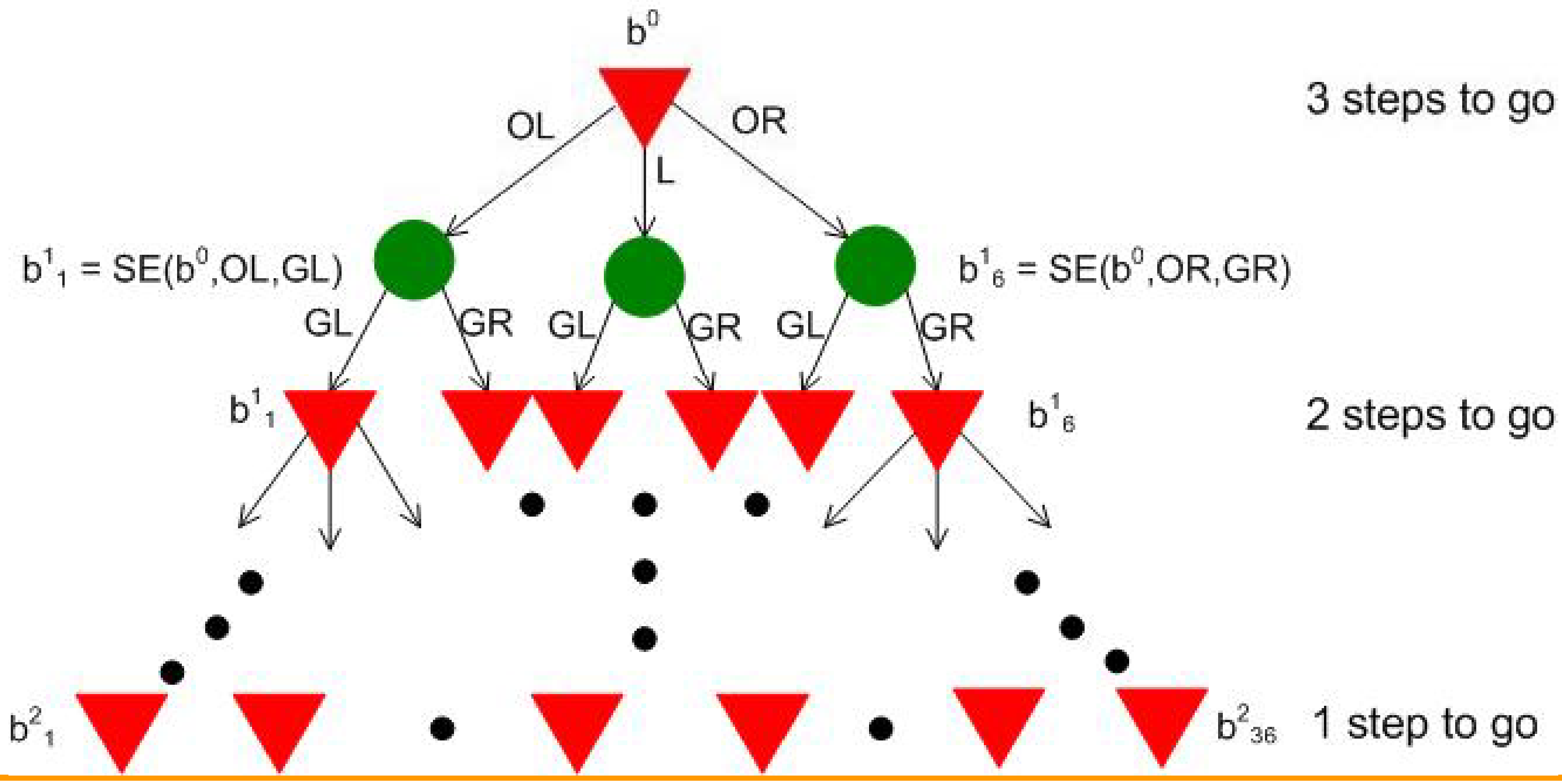
2. Update beliefs:

$$SE : \Delta(S) \times A \times \Omega \rightarrow \Delta(S)$$



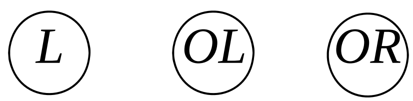
Background: POMDP

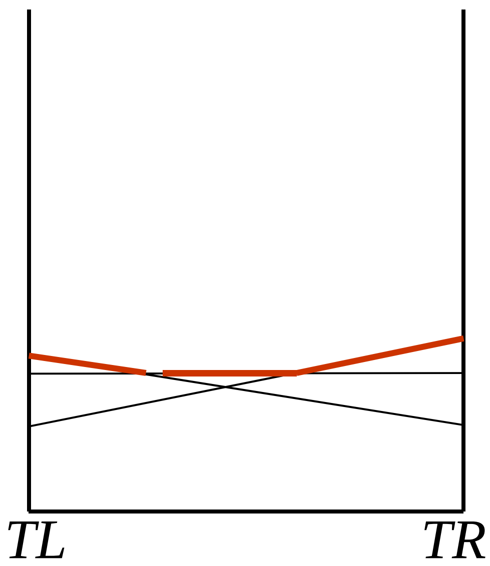
3. Optimal policy computation:
   - Build the look ahead reachability tree
   - Dynamic programming (DP)



Background: POMDP

Dynamic Programming in POMDPs

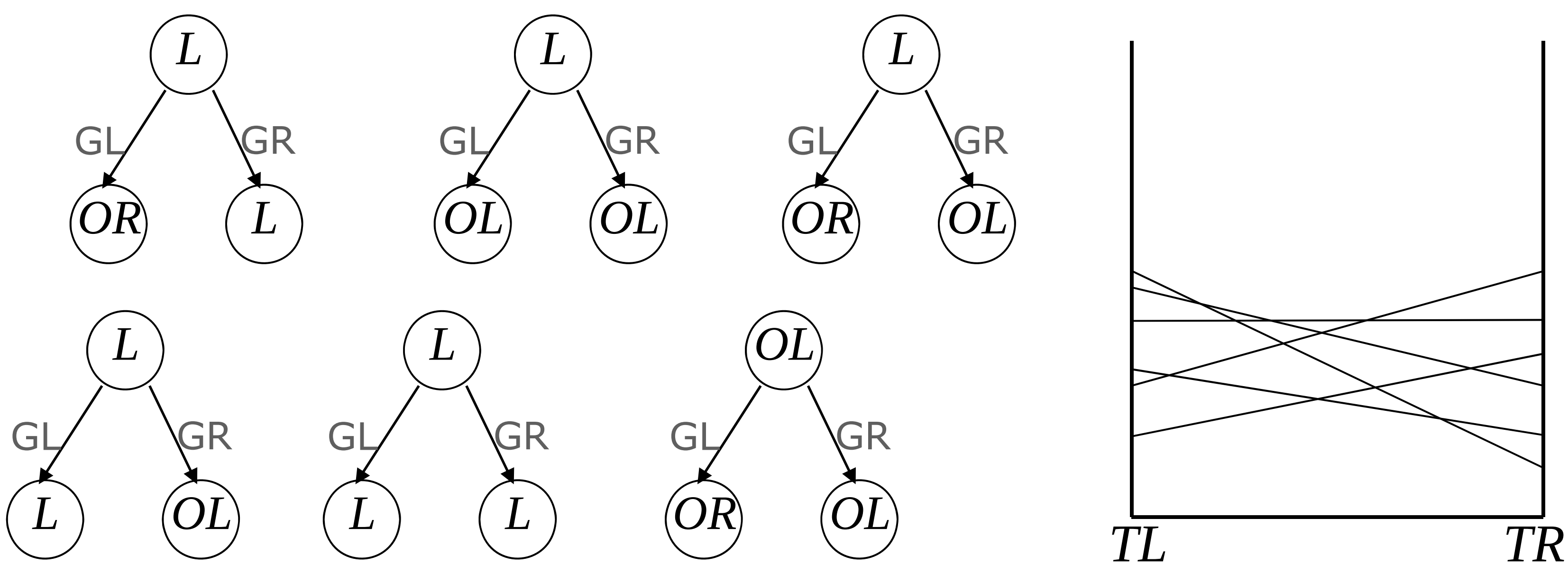




Background: POMDP

DP in POMDPs

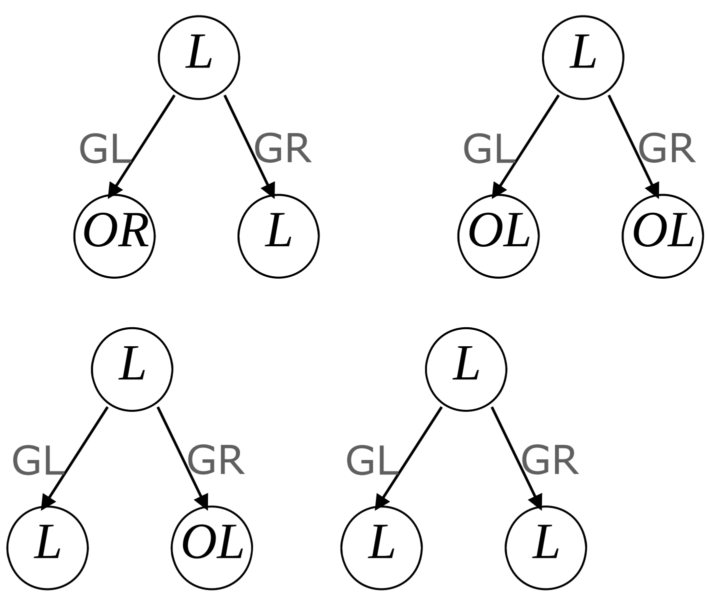
Number of policy trees is exponential in observations and doubly exponential in horizons!

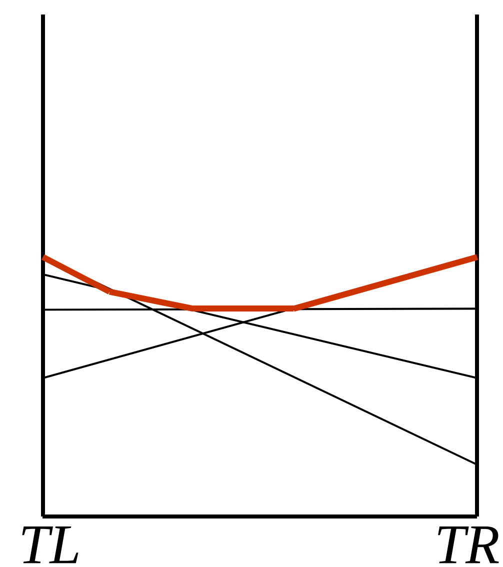


Background: POMDP

DP in POMDPs

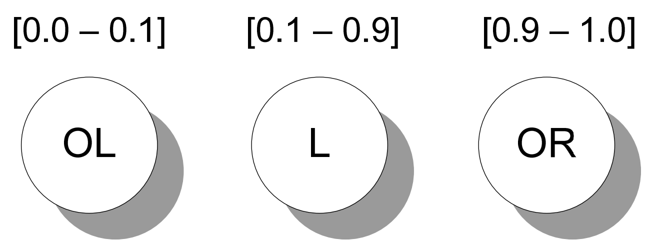
Prune suboptimal policy trees



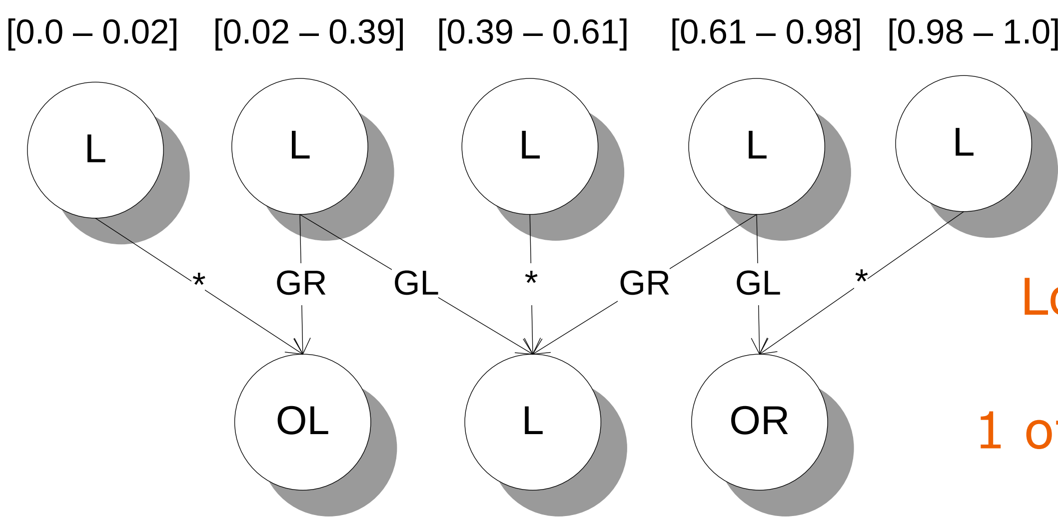


Background: POMDP

Policies in the tiger problem



Look ahead 1 step
(horizon 1)



Look ahead 2 steps
(horizon 2)
1 of 4 different policies

I-POMDP

Key ideas

- Include possible behavioral models of other agents in the state space. Agent's beliefs are distributions over the physical state and models of others
  - Intentional (types) and subintentional models
- Intentional models contain beliefs. Beliefs over models give rise to interactive belief systems
  - Interactive epistemology, recursive modeling
- Finitely nested belief system as a computable approximation of the interactive belief system
- Compute best response to agent's belief (subjective rationality)

Potential applications

- Robotics
  - Planetary exploration

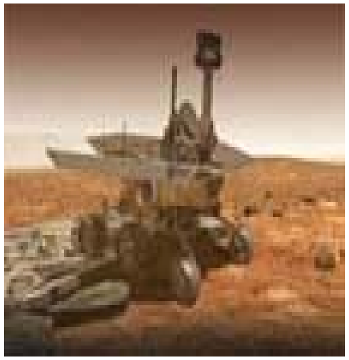
    Surface mapping by rovers

    Coordinate to explore pre-defined region optimally

    Uncertainty due to sensors
  - Robot soccer

    Coordinate with teammates and deceive opponents

    Anticipate and track others' actions



Spirit

Opportunity

RoboCup Competition

I-POMDP

Definition of a finitely nested I-POMDP of strategy level l for agent *i* in a 2 agent setting

$$\left\langle IS_{i,l}, A, T_i, \Omega_i, O_i, R_i, OC_i \right\rangle$$

$IS_{i,l}$ is the set of interactive states

$$IS_{i,l} = S \times M_{j,l-1} \quad where \quad M_{j,l-1} = \Theta_{j,l-1} \cup SM_j$$

$$\theta_{j,l-1} = \left\langle b_{j,l-1}, A, T_j, \Omega_j, O_j, R_j, OC_j \right\rangle$$ and Bayes rational

I-POMDP

Definition of a finitely nested I-POMDP of strategy level l for agent *i* in a 2 agent setting

$$\left\langle IS_{i,l}, A, T_i, \Omega_i, O_i, R_i, OC_i \right\rangle$$

$IS_{i,l}$ is the set of interactive states

A is the set of joint actions

T_i is the transition function defined on the physical state (beliefs of others cannot be directly manipulated)

Ω_i is the set of observations of agent *i*

O_i is the observation function (beliefs of others are not directly observable)

R_i is the reward function of agent *i*

Interactive beliefs in I-POMDP

- *"In interactive contexts [...], it is important to take into account not only what the players believe about substantive matters [...] but also what they believe about the beliefs of other players."*
- *"One specifies what each player believes about the substantive matters, about the beliefs of others about these matters, about the beliefs of others about the beliefs of others, and so on ad infinitum."*

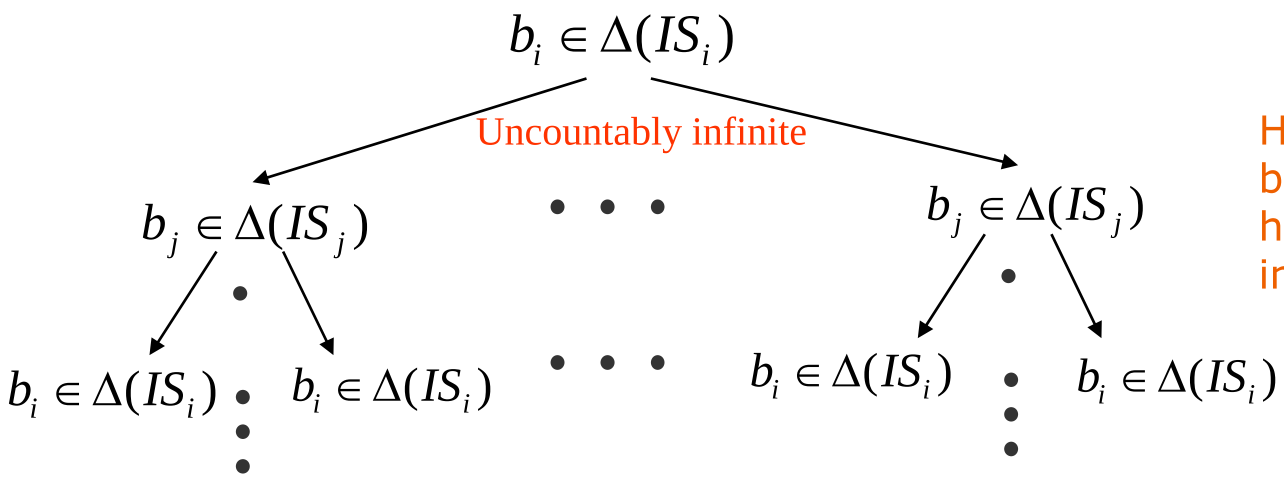
  - Robert J. Aumann

- New concept: Interactive beliefs
- New approach to game theory: Epistemic, decision analytic

Interactive beliefs in I-POMDP

Agent *i*'s belief is a distribution over the physical state and models of j

$$
\begin{aligned}
b_i \in \Delta(IS_i) &= \Delta(S \times M_j) \\
&= \Delta(S \times \{\langle B_j \times \widehat{\Theta}_j \rangle \cup SM_j\})
\end{aligned}
$$



Hierarchical belief systems have been explored in game theory

Observation

- Amount of information in interactive belief hierarchy is finite
  - Information content decreases asymptotically with the number of levels

- Question 1: How many levels should we include?

  Answer: As many as we can

- Can one work with infinite levels?

  Answer: Yes, in some special cases

Observation

- Minimax in Chess game
  - Model of agent’s possible moves
  - Model the other player’s possible responses
    - Assume she is rational (is she?)
  - Model the other player modeling the agent’s possible responses
    - Assume she believes agent is rational (does she?)
  - Model further ...
    - Assume that she believes that agent believes that she is rational ...
- Include as much detail and levels as you can

I-POMDP

- Integrate models of others in a decision-theoretic framework
  - An important model is a POMDP describing an agent – it includes all factors relevant to agent's decision making. These are **intentional models (BDI)**
  - Represent uncertainty by maintaining beliefs over the state and models of other agents. This gives rise to interactive belief systems
    - interactive epistemology
  - When no other agents are present beliefs become "flat" and classical POMDP results
  - Computable approximation of the interactive beliefs: finitely nested belief systems
    - infinitely nested beliefs are computable if there is common knowledge – Nash equilibria

Belief update in I-POMDP

- Formalization

$$
\begin{aligned}
&Pr(is^t|a_i^{t-1}, b_{i,l}^{t-1}) = \beta \sum_{IS^{t-1}:\widehat{m}_j^{t-1}=\widehat{\theta}_j^t} b_{i,l}^{t-1}(is^{t-1}) \\
&\times \sum_{a_j^{t-1}} Pr(a_j^{t-1}|\theta_{j,l-1}^{t-1}) O_i(s^t, a_i^{t-1}, a_j^{t-1}, o_i^t) \\
&\times T_i(s^{t-1}, a_i^{t-1}, a_j^{t-1}, s^t) \sum_{o_j^t} O_j(s^t, a_i^{t-1}, a_j^{t-1}, o_j^t) \\
&\times \tau(SE_{\widehat{\theta}_j^t}(b_{j,l-1}^{t-1}, a_j^{t-1}, o_j^t) - b_{j,l-1}^t)
\end{aligned}
$$

Belief update in I-POMDP

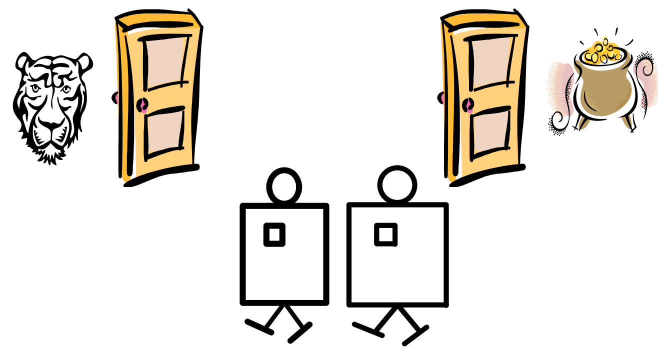
Multiagent Tiger problem

Task: Maximize collection of gold over a finite or infinite number of steps while avoiding tiger

Each agent hears growls as well as creaks (S, CL, or CR)

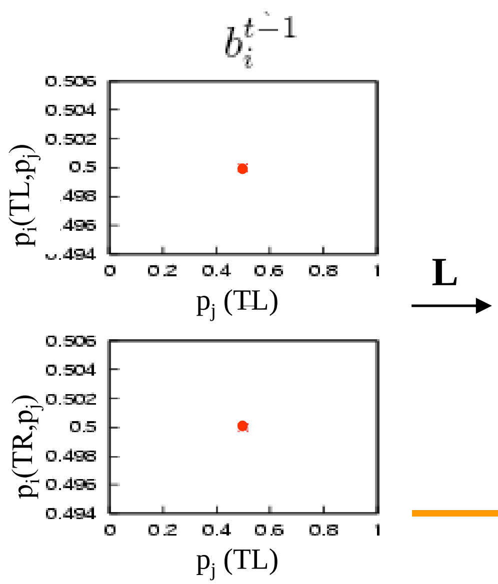
Each agent may open doors or listen

Each agent is unable to perceive other's observation



agents *i* & *j*

Understanding the I-POMDP (level 1) belief update



Belief update in I-POMDP

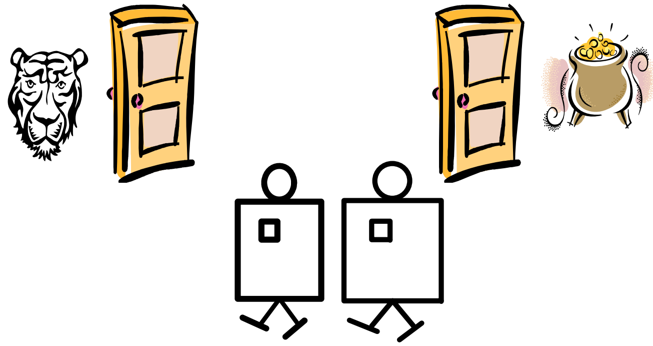
Multiagent Tiger problem

Task: Maximize collection of gold over a finite or infinite number of steps while avoiding tiger

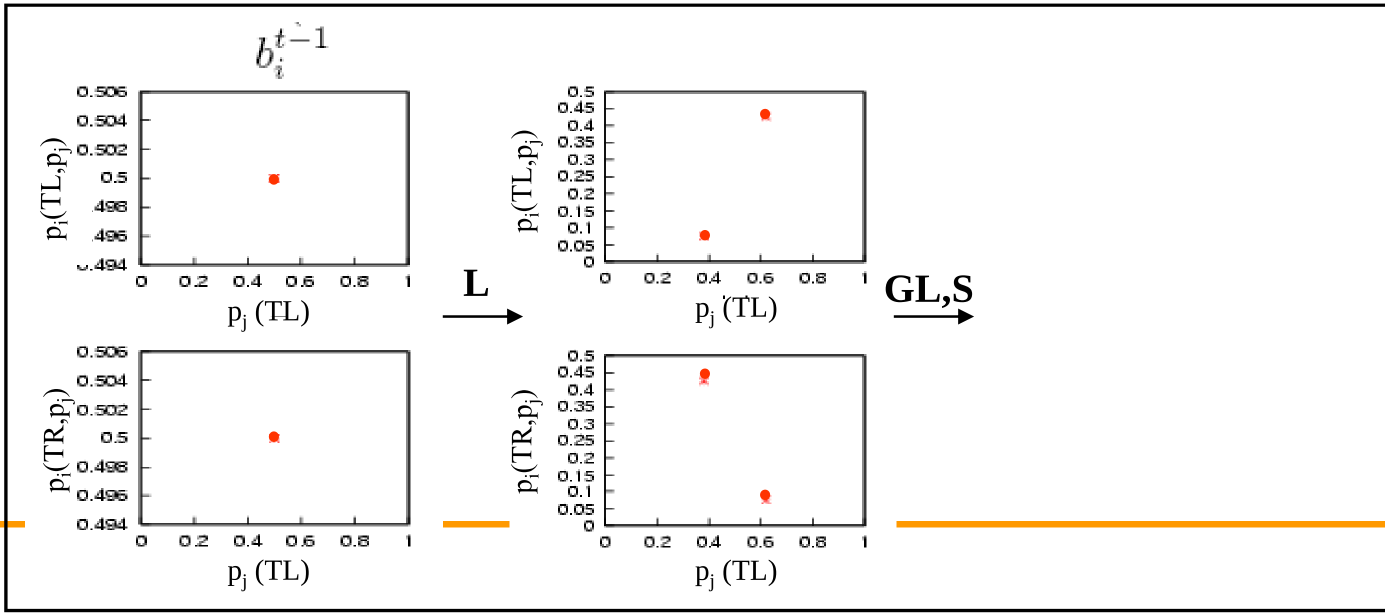
Each agent hears growls as well as creaks (S, CL, or CR)

Each agent may open doors or listen

Each agent is unable to perceive other's observation



Understanding the I-POMDP (level 1) belief update



Belief update in I-POMDP

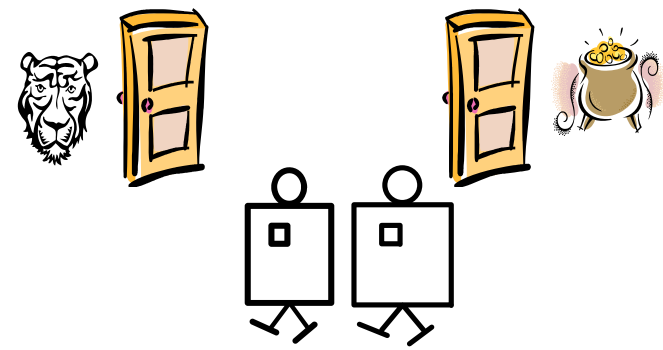
Multiagent Tiger problem

Task: Maximize collection of gold over a finite or infinite number of steps while avoiding tiger

Each agent hears growls as well as creaks (S, CL, or CR)

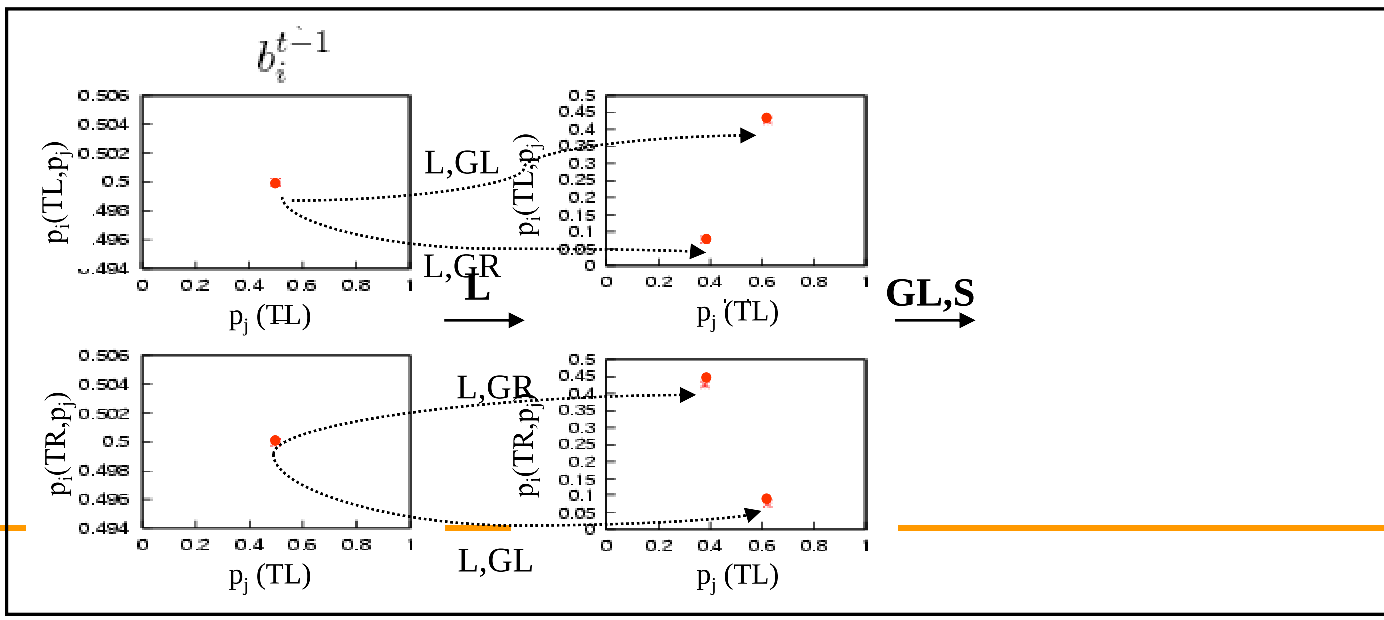
Each agent may open doors or listen

Each agent is unable to perceive other's observation



agents *i* & *j*

Understanding the I-POMDP (level 1) belief update



Belief update in I-POMDP

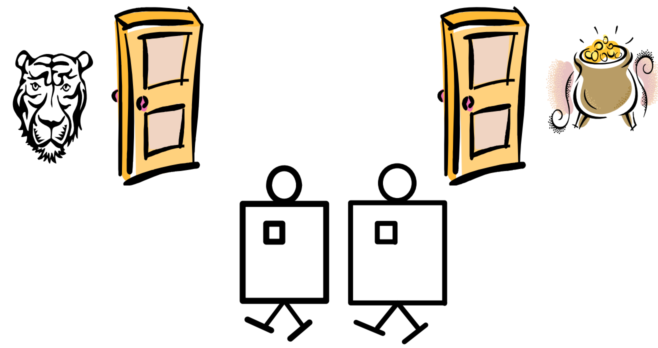
Multiagent Tiger problem

Task: Maximize collection of gold over a finite or infinite number of steps while avoiding tiger

Each agent hears growls as well as creaks (S, CL, or CR)

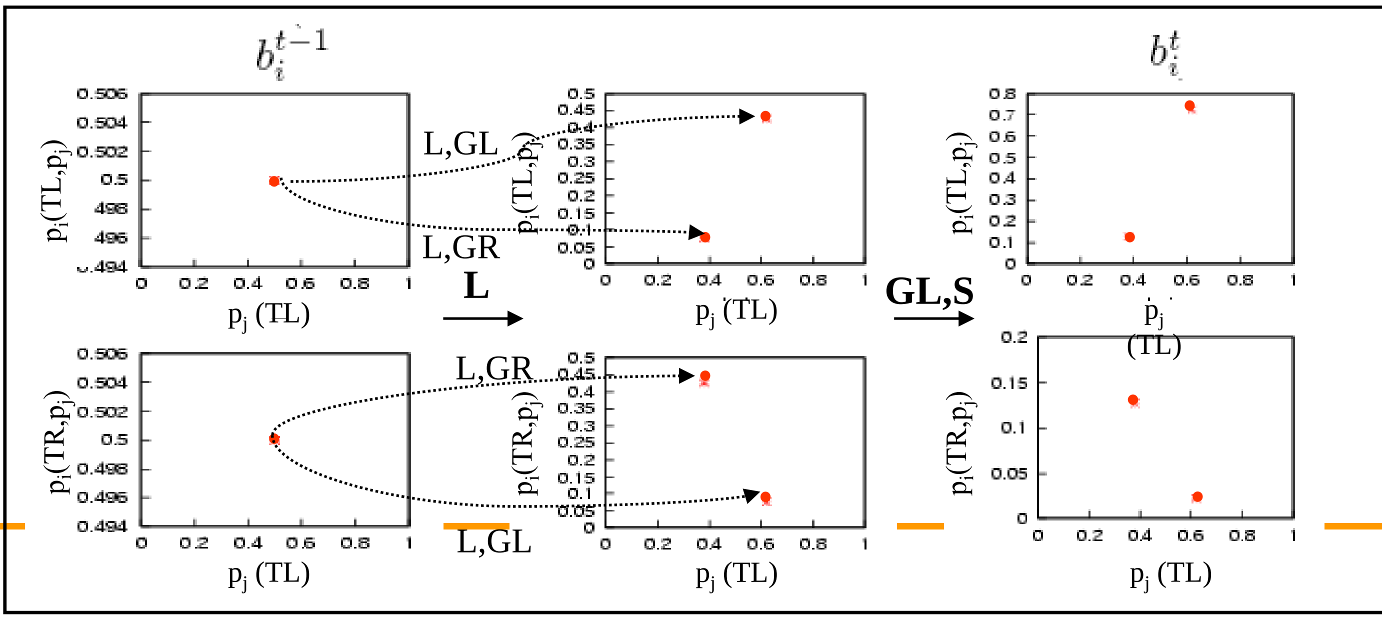
Each agent may open doors or listen

Each agent is unable to perceive other’s observation



agents *i* & *j*

Understanding the I-POMDP (level 1) belief update



DP in I-POMDP

Recurse through levels beginning with level 0

Agent j
level 0 models of horizon 1
(assumes agent *i* is noise)

a_2 a_1 a_1 a_1 a_2 a_1 a_2

DP in I-POMDP

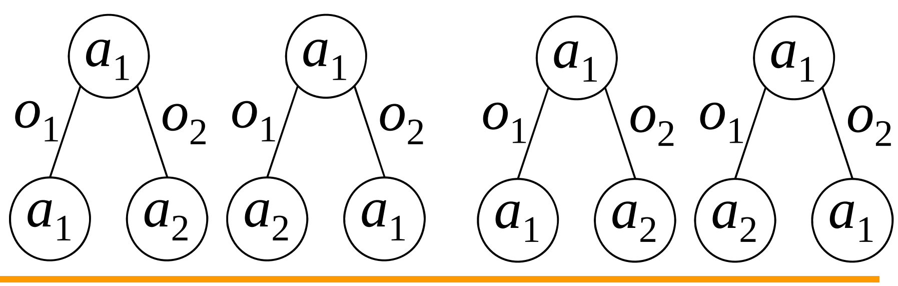
Best response to level 1 belief at horizon 1

Agent i level 1	Agent j level 0 models of horizon 1
a_1	a_2 a_1 a_1 a_1 a_2 a_1 a_2

DP in I-POMDP

Agent i
level 1

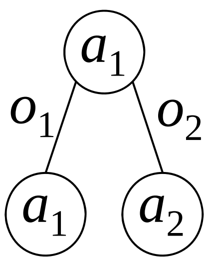
Agent j
level 0 models of horizon 2



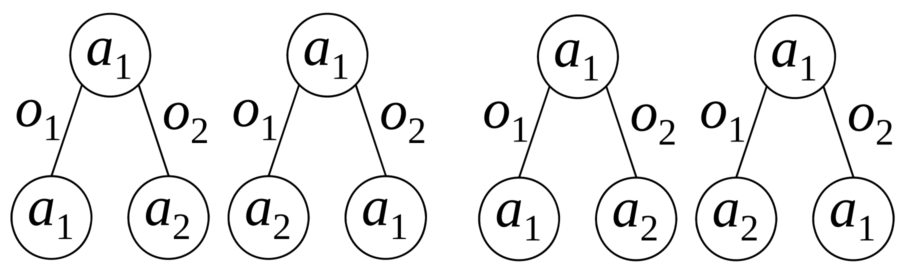
DP in I-POMDP

Best response to level 1 belief at horizon 2

Agent i
level 1



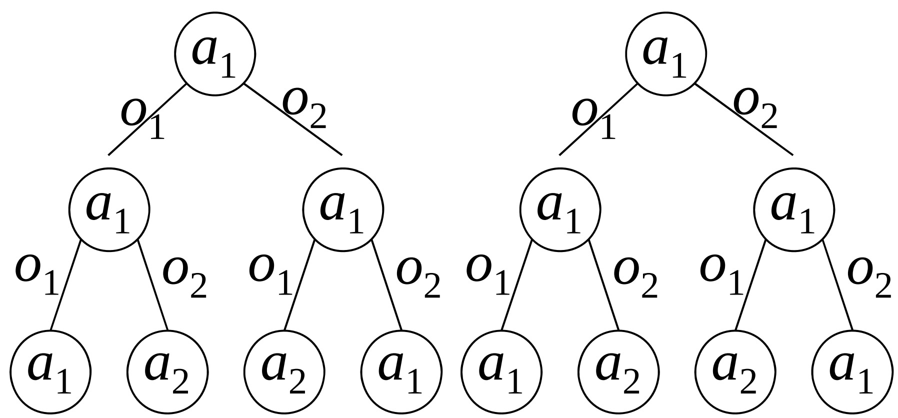
Agent j
level 0 models of horizon 2



DP in I-POMDP

Agent i
level 1

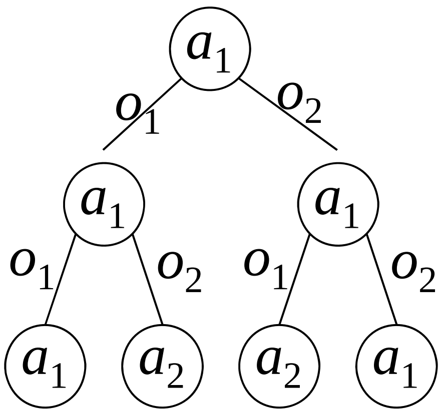
Agent j
level 0 models of horizon 3



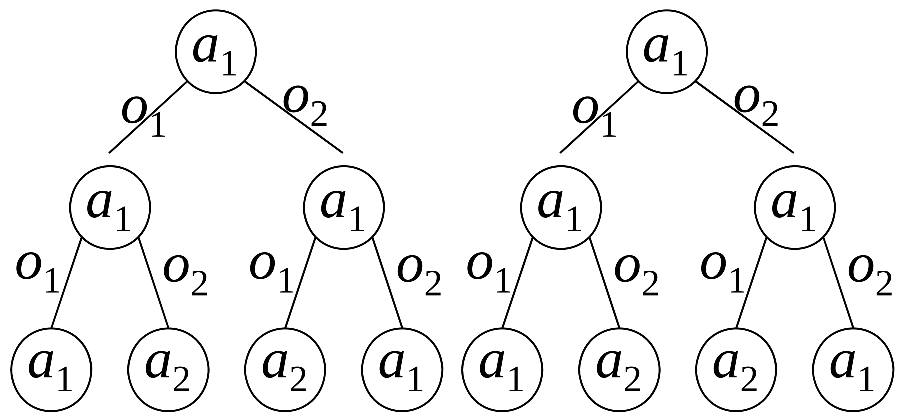
DP in I-POMDP

Best response to level 1 belief at horizon 3

Agent i
level 1



Agent j
level 0 models of horizon 3



POMDPs and I-POMDPs

- Beliefs – probability distributions over states are sufficient statistics
  - They fully summarize the information contained in any sequence of observations
- Solving POMDPs is hard (P-space)
  - We need approximations (e.g., particle filtering)
- Solving I-POMDPs is at least as hard
  - An approximation: interactive particle filtering
- If recursion does not terminate, look for fixed points

Improving DP in I-POMDP

The interactive state space is very large because it includes models of other agents. Theoretically, the space of computable models is countably infinite

- The curse of dimensionality is especially potent for I-POMDP
- I-POMDP faces the curse of history afflicting both agents

Can we reduce the size of the interactive state space and thereby mitigate the curse of dimensionality?

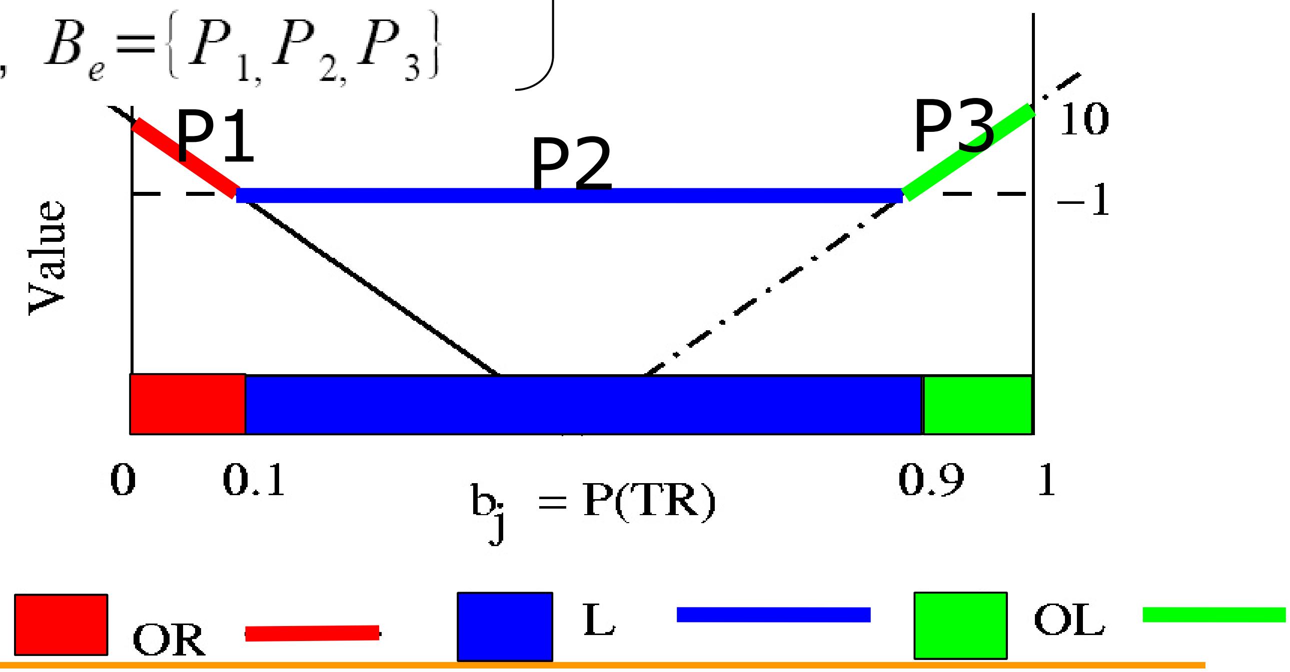
Improving DP in I-POMDP

Can we reduce the size of the interactive state space and thereby mitigate the curse of dimensionality without loss in value?

Behaviorally equivalent models

- $B_e = \{P_i \mid P_i \subseteq \Delta(S), \cup_i P_i = \Delta(S)\}$
- $\forall_{b_m, b_n \in P_i} OPT(b_m \in P_i) = OPT(b_n \in P_i)$
- In the previous example, $B_e = \{P_{1,} P_{2,} P_3\}$

Equivalence Classes of Beliefs



Equivalence classes of interactive states

- Definition
  - Combination of a physical state and an equivalence class of models

$$ECIS_{i,l} = \Big\{ (s, M_{j,k}^{l-1}) \mid M_{j,k}^{l-1} \subseteq \{\Theta_{j,l-1} \cup SM_j\},$$

$$\forall_{m_{j,l-1}, m'_{j,l-1} \in M_{j,k}^{l-1}} OPT(m_{j,l-1}) = OPT(m'_{j,l-1})$$

$$\Big\}$$

Lossless aggregation

- In a finitely nested I-POMDP, a probability distribution over $ECIS_{i,l}$, $\bar{b}_{i,l} \in \Delta(ECIS_{i,l})$, provides a sufficient statistic for the past history of *i*'s observations
- Transformation of the interactive state space into behavioral equivalence classes is value-preserving
- Optimal policy of the transformed finitely nested I-POMDP remains unchanged

Solving I-POMDPs exactly

Procedure Solve-IPOMDP ($AGENT_i$, Belief Nesting L) :

Returns Policy

If L = 0 Then

Return { Policy : = Solve-POMDP ($AGENT_i$) }

Else

For all $AGENT_j$ < > $AGENT_i$

$Policy_j$: = Solve-IPOMDP($AGENT_j$, L-1)

End

M_j := Behavioral-Equivalence-Models($Policy_j$)

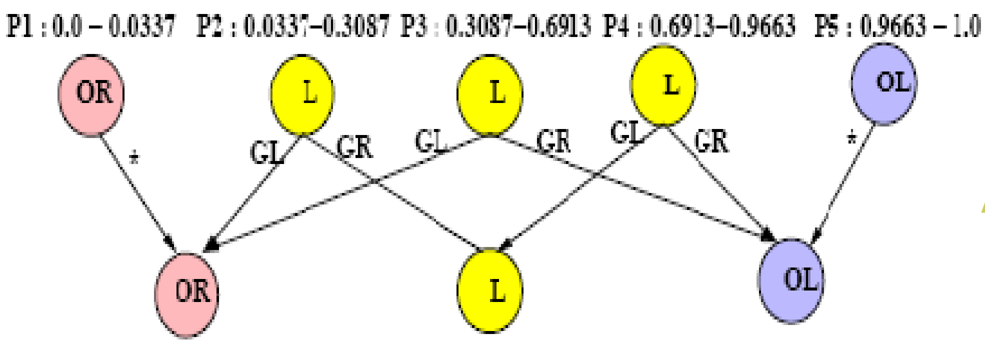
$ECIS_i$: = S x **{ x_j M_j }**

Policy : = Modified-GIP($ECIS_i$, A_i , T_i , Ω_i , O_i , R_i)

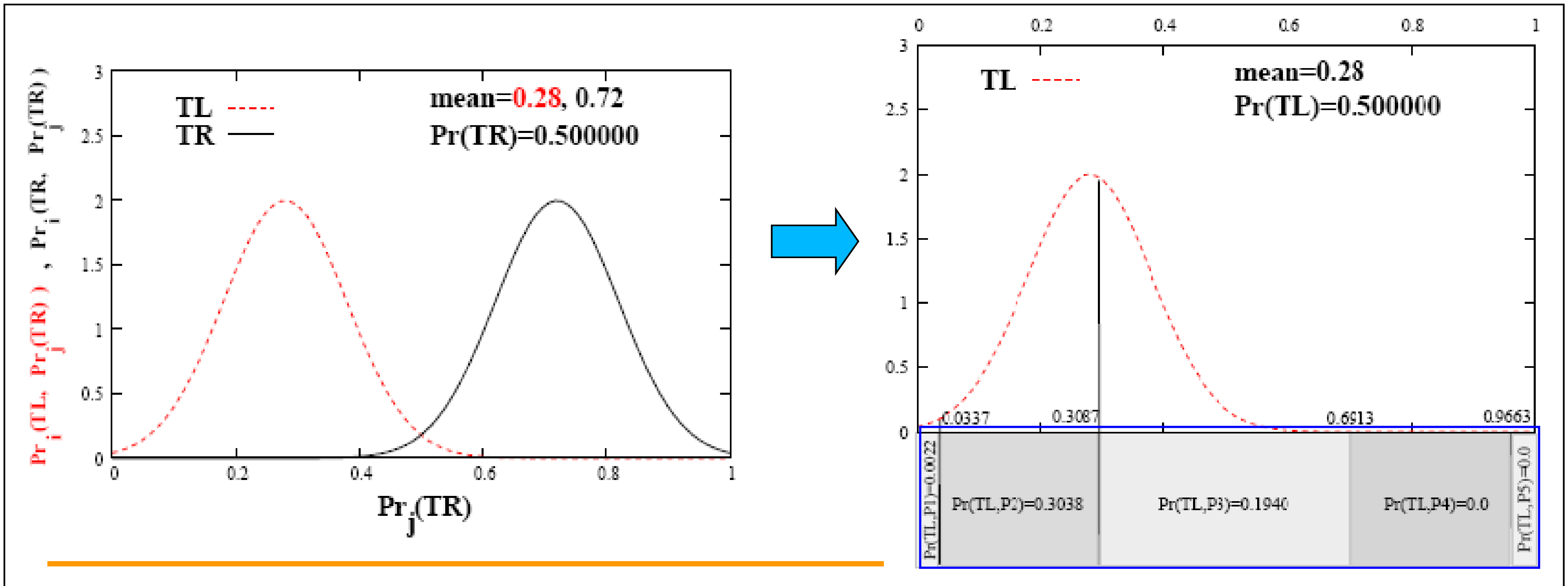
Return Policy

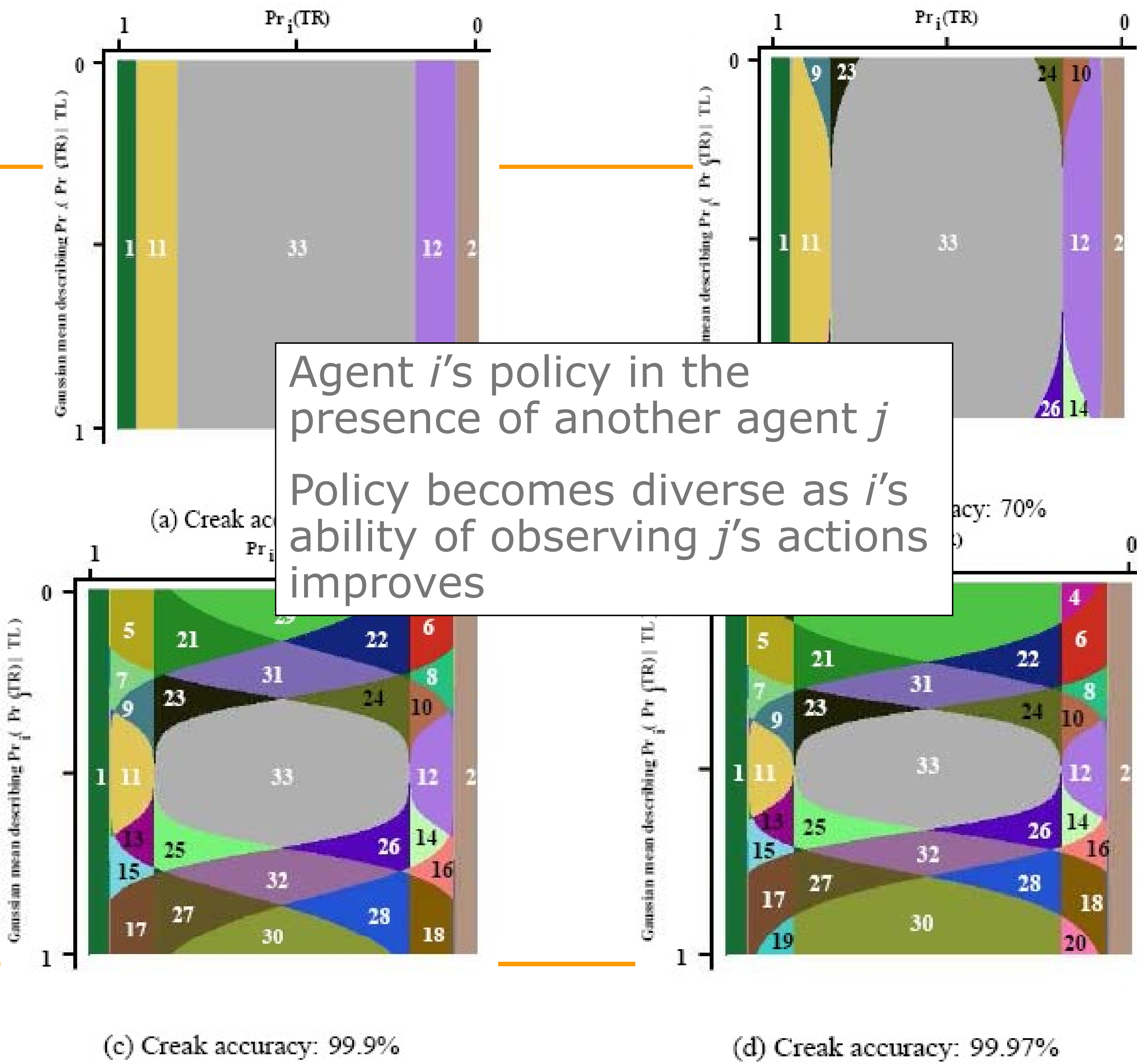
End

Beliefs on ECIS



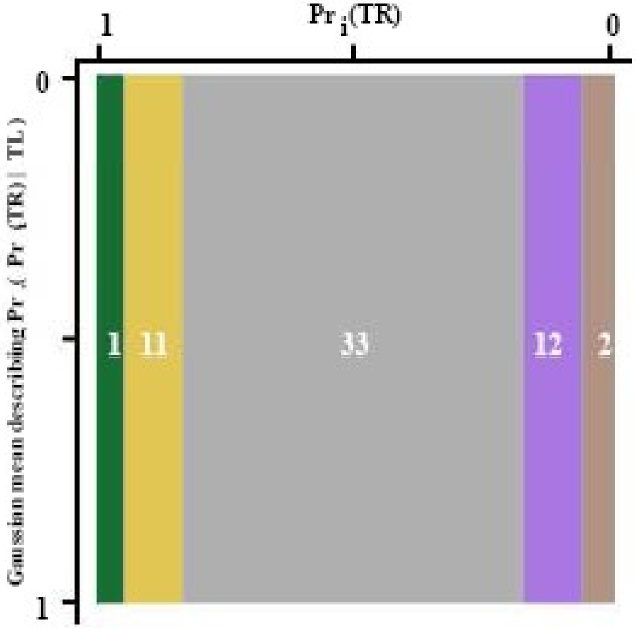
Agent j's policy



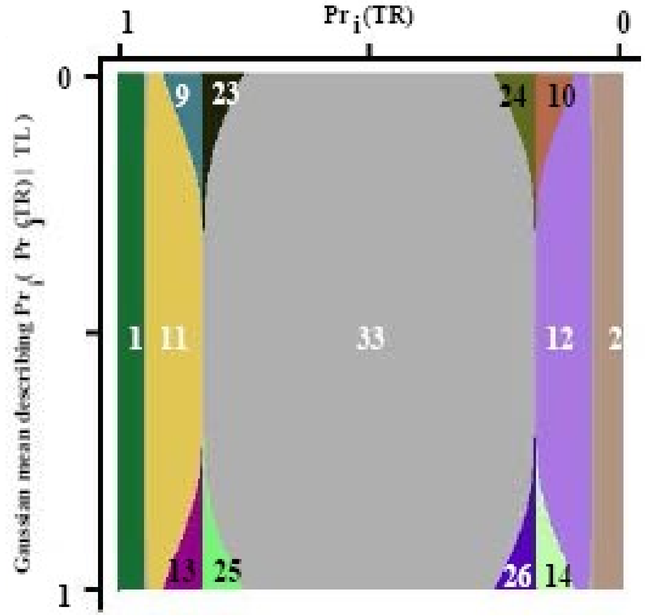


Agent *i*'s policy in the presence of another agent *j*

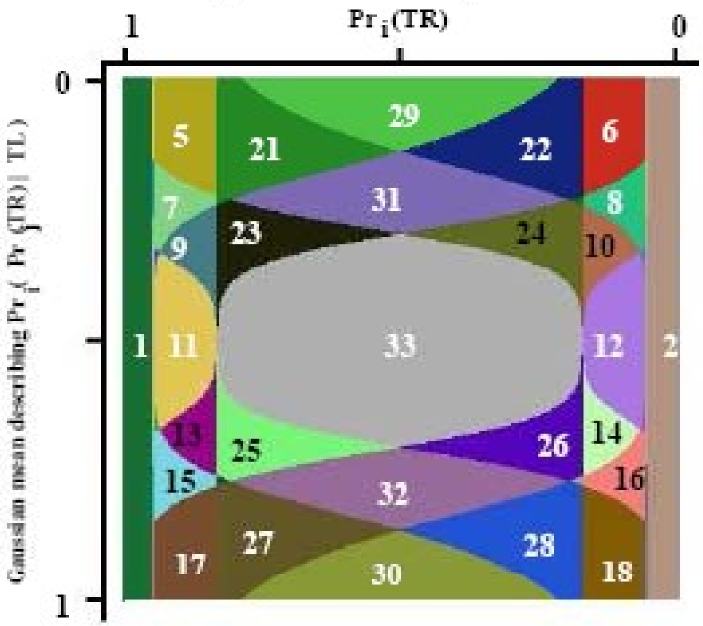
Policy becomes diverse as *i*'s ability of observing *j*'s actions improves



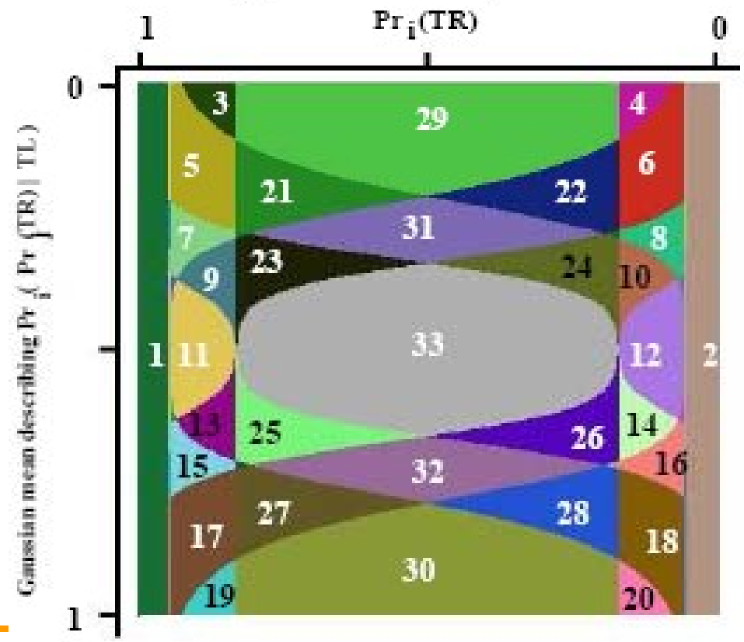
(a) Creak accuracy: 33%



(b) Creak accuracy: 70%



(c) Creak accuracy: 99.9%



(d) Creak accuracy: 99.97%

Discussion on ECIS

- A method that enables exact solution of finitely nested interactive POMDPs
- Aggregate agent models into behavioral equivalence classes
  - Discretization is lossless
- Interesting behaviors emerge in the multi-agent Tiger problem

Improving DP in I-POMDP

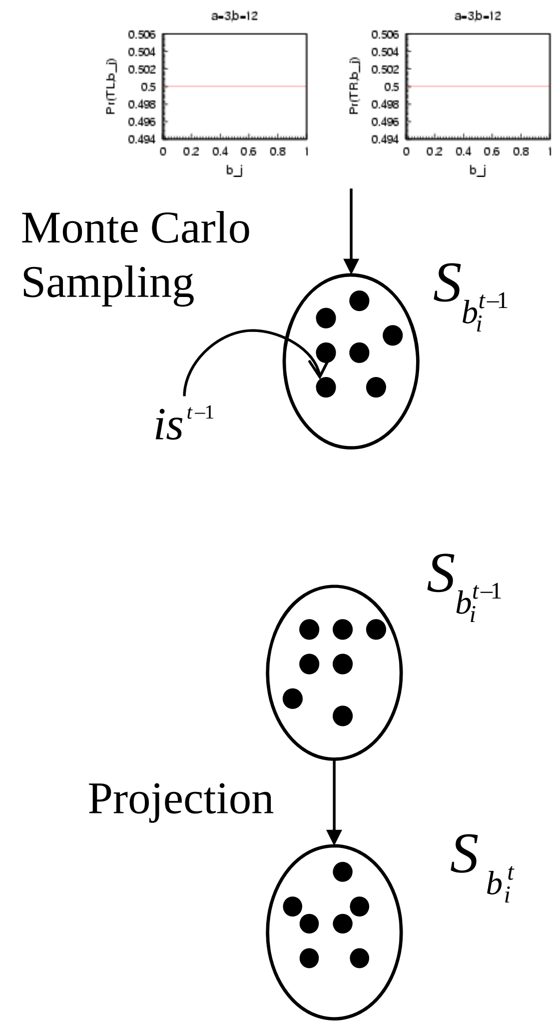
Can we reduce the size of the interactive state space and thereby mitigate the curse of dimensionality permitting loss in value?

Monte Carlo sampling in I-POMDP

1. Sample interactive states using agent's belief as the sampling distribution

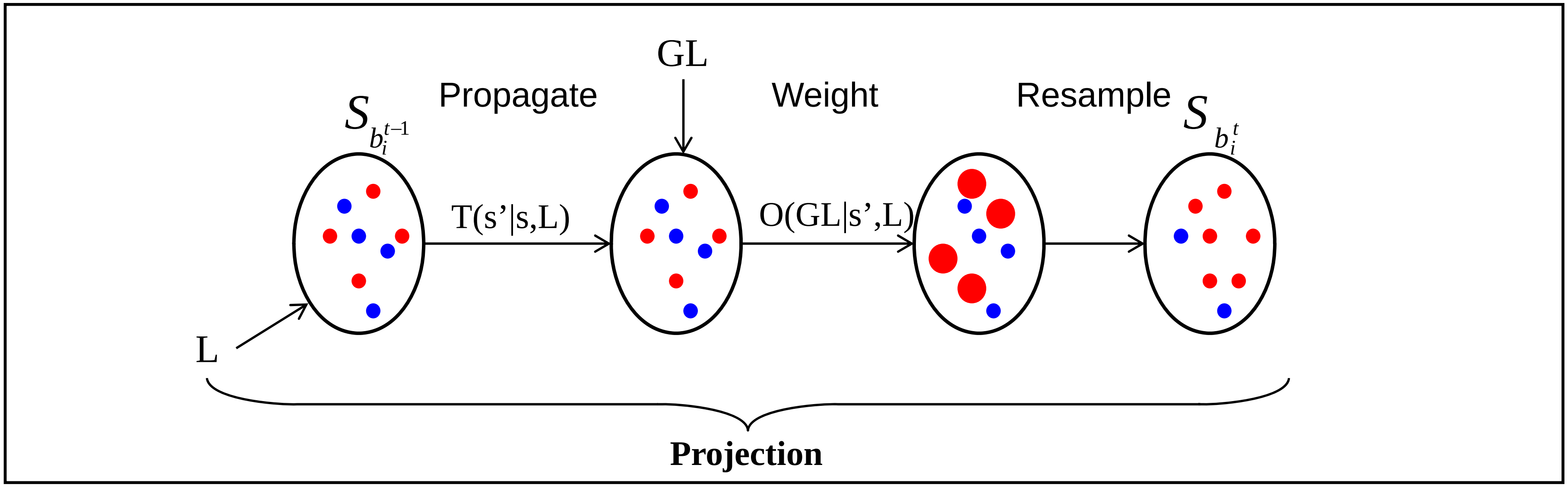
2. Project the set of samples over time

3. Perform DP using sampled set



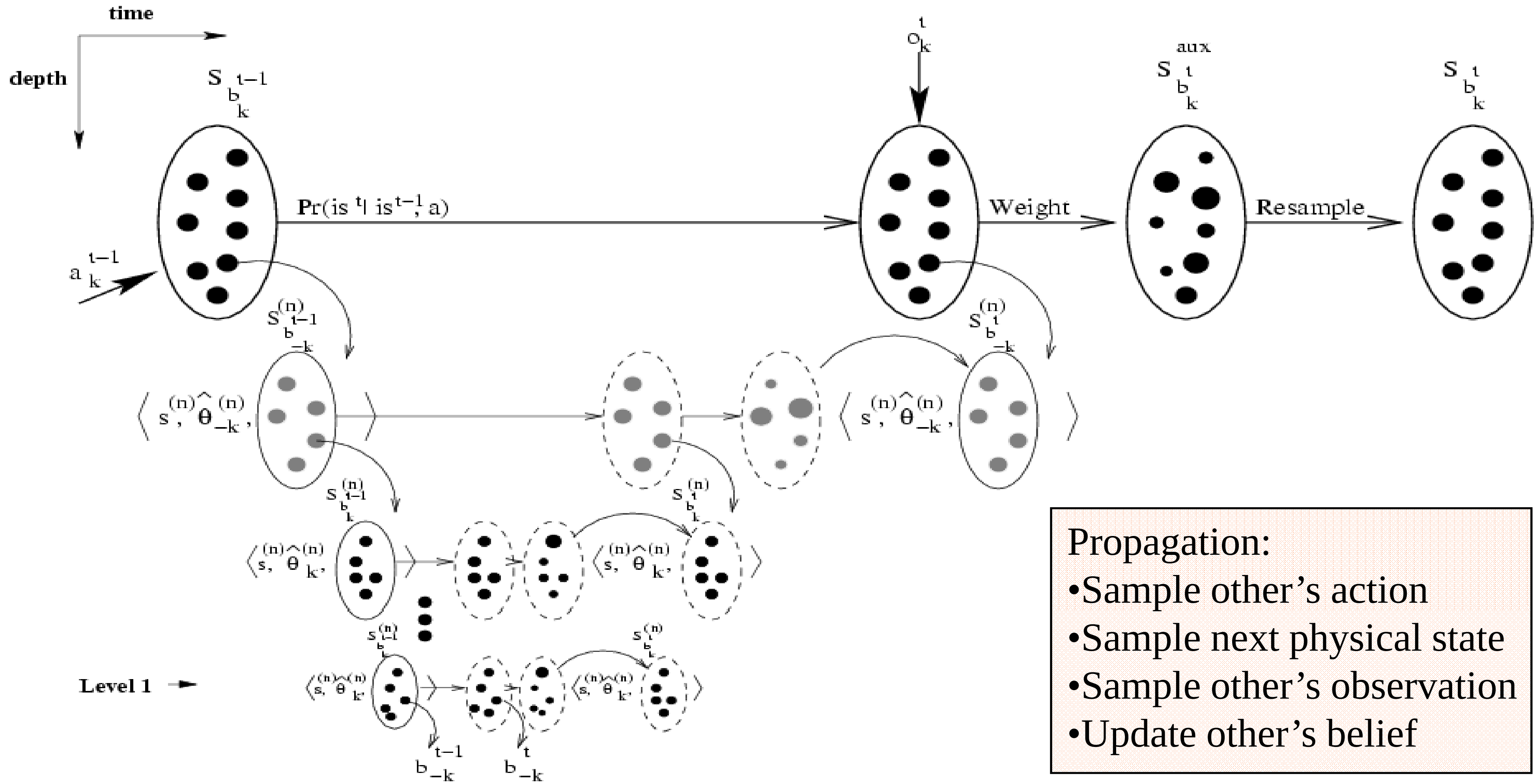
Particle filtering

Single-agent Tiger problem



Interactive particle filtering

Approximating the I-POMDP belief update



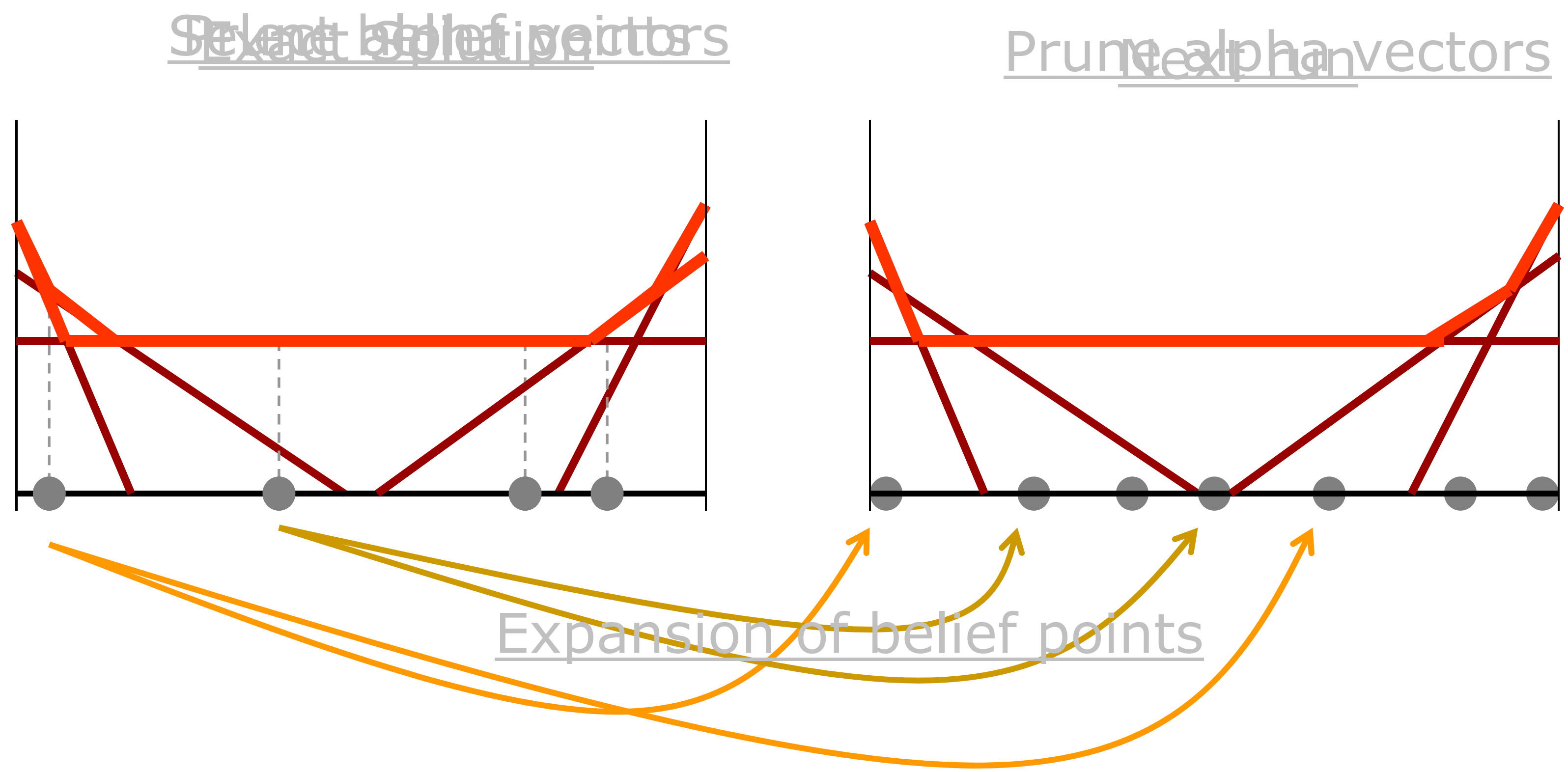
Propagation:
- Sample other's action
- Sample next physical state
- Sample other's observation
- Update other's belief

Improving DP in I-POMDP

Can we mitigate the curse of history by avoiding planning for all the observation histories for both agents?

Point Based Value Iteration (PBVI)

- Potentially scalable approach for solving POMDPs approximately (Pineau et al., ’03,‘06)



Point Based Value Iteration

- Many different belief expansion strategies
  - Stochastic trajectory generation
  - Greedy error minimization
  - Gain based methods (Samples et al. ’07)
- Improvements on PBVI
  - Randomly backing up vectors at select points (Perseus; Spaan&Vlassis, ’05)
  - Prioritized vector backup (Shani et al. ’06)

Interactive PBVI (I-PBVI)

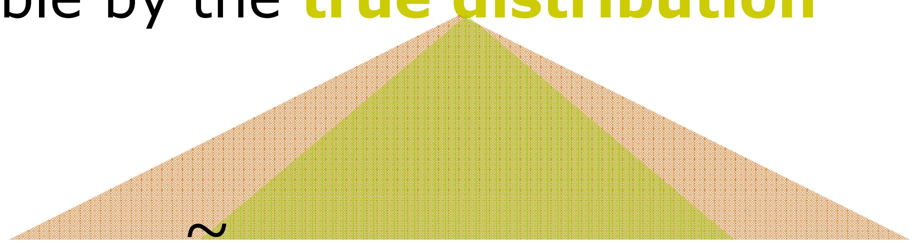
- Hypothesis: Extending PBVI approach to I-POMDPs results in a scalable approximation for I-POMDPs

- Generalizing PBVI to multiagent settings is not trivial
  - Research challenges:
    1. Space of agent models is countably infinite
    2. Parameterized representation of nested beliefs is difficult
    3. Other agents' actions need to be predicted suggesting a recursive implementation

Issue 1: Space of agent models is infinite

<u>Approach</u>

- Analogous to PBVI in POMDPs, select a few initial models of the other agent
  - Need to ensure that the true model is within this set, otherwise the belief update is inconsistent
- Select models so that the *Absolute Continuity Condition* is satisfied
  - **Subjective distribution** over future observations (paths of play) should not rule out the observation histories considered possible by the **true distribution**



- How to satisfy ACC?
  - Cautious beliefs
  - Select a finite set of models, $\tilde{\Theta}_{i/j}$, with the partial (domain) knowledge that the true or an equivalent model is one of them

Issue 2: Representing nested beliefs is difficult

- Level 0 beliefs are standard discrete distributions (vectors of probabilities that sum to 1)
- Level 1 beliefs could be represented as probability density functions over level 0 beliefs
- Probability density functions over level 1 beliefs may not be computable in general
  - Parameters of level 1 beliefs may not be bounded (e.g., a polynomial of any degree)
  - Level 2 beliefs are strictly partial recursive functions

Approach

- We previously limited the set of models, $\tilde{\Theta}_{i/j}$
- Level *l* belief becomes a *discrete* probability distribution

$$\widetilde{IS}_{i,l} = S \times \tilde{\Theta}_{j,l-1} \qquad \tilde{b}_{i,l} \in \Delta(\widetilde{IS}_{i,l})$$

Issue 3: Predict other agent's actions

Approach

- Candidate agent models grow over time and must be tracked
  - Define a complete interactive state space

Reach($\tilde{\Theta}_{j,l-1}$,0) = $\tilde{\Theta}_{j,l-1}$

Reach($\tilde{\Theta}_{j,l-1}$, *H*) = Set of models of agent *j* in the course of H steps

$$\tilde{IS}_{i,l} = S \times \mathrm{Reach}(\tilde{\Theta}_{j,l-1}, H)$$

- Solve other agent's models at each level to predict actions
  - Recursively invoke I-PBVI to solve models

Interactive PBVI

- Back project alpha vectors for I-POMDPs (see paper)
- Retain alpha vectors optimal at selected belief points

I-PBVI (Initial beliefs: $\langle \tilde{B}^N_{k,l}, \tilde{B}^N_{-k,l-1}, \ldots, \tilde{B}^N_{k,0} \rangle$, Horizons: $H > 0$, Strategy level: $l \geq 0$)

1: $\tilde{\Gamma}^H \leftarrow$ Initial-AlphaVectors ()
2: **for** $t \leftarrow H - 1$ to 0 **do**
3:     **if** $l = 0$ **then**
4:         $\tilde{\Gamma}^t \leftarrow$ PBVI Backup($\tilde{B}^N_{k,0}, \tilde{\Gamma}^{t+1}, H - t$)
5:     **else**
6:         $\tilde{\Gamma}^t \leftarrow$ I-PBVI Backup($\tilde{B}^N_{k,l}, \ldots, \tilde{B}^N_{k,0}, \tilde{\Gamma}^{t+1}, H - t, l$)
7:     Expand the previous set of beliefs at all levels recursively
8:     Add the expanded beliefs to the existing sets
9: **return** $\tilde{\Gamma}^0$

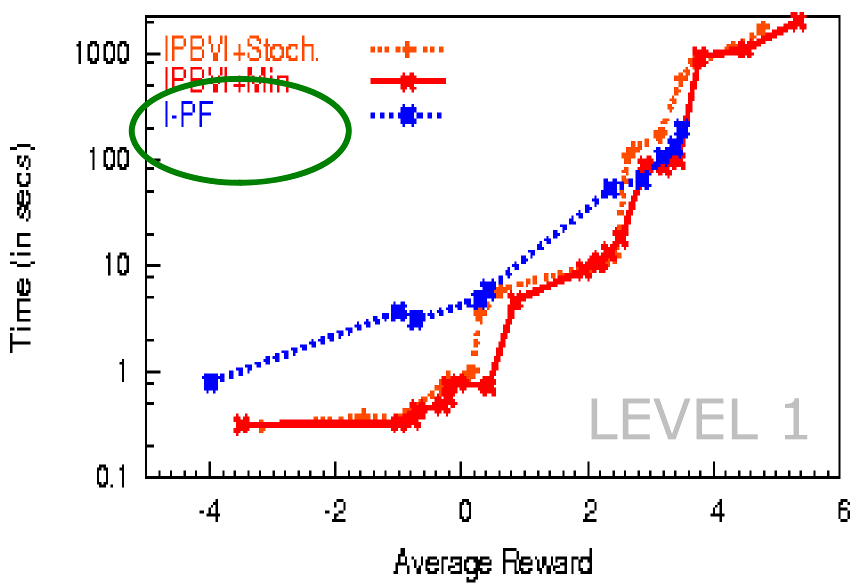
- Computational Savings

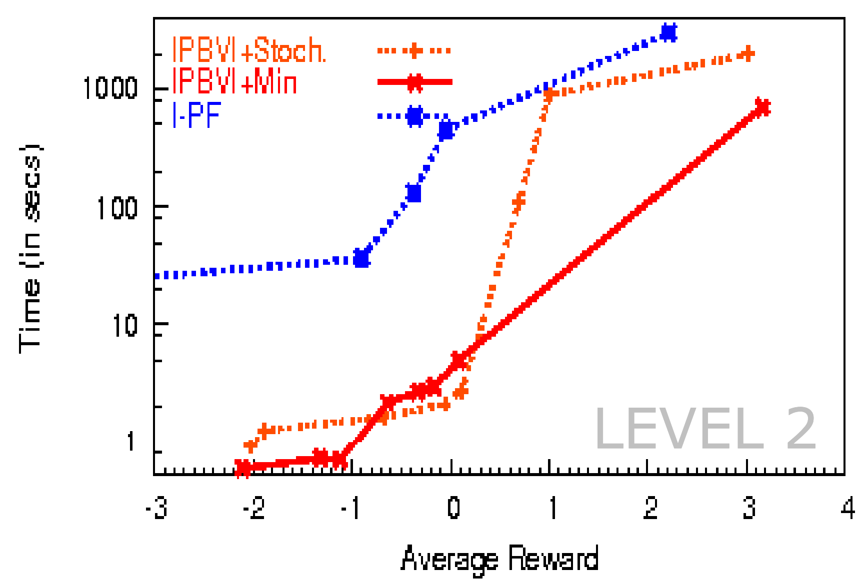
$$\mathcal{O}(N(l+1)) \qquad \mathcal{O}(|A_i||\mathcal{V}^{t+1}|^{|\Omega_i|} + |M|l)$$

Experimental Results

- Measured the least time taken in reaching a particular performance in terms of the rewards
  - Function of belief points, number of models and horizons
  - Compared with Interactive Particle Filter (I-PF)

Multiagent Tiger Problem





(Dual Processor Xeon, 3.4GHz, 4GB RAM, Linux)

Discussion on I-PBVI

- Interactive PBVI generalizes PBVI to multiagent settings
  - The generalization is not trivial
- I-PBVI demonstrates scalable results on toy problems
  - Further testing on realistic applications is within reach
- Further improvement is possible by carefully limiting the set of models in Reach()
  - True or equivalent model must remain in the set otherwise the belief update may become inconsistent

DP in I-POMDP for stationary policies

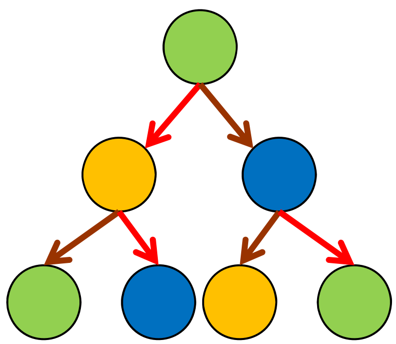
Can we directly improve I-POMDP policies instead of first improving the value function and then obtaining a better policy?

Policy Iteration

- Class of solution algorithms – search policy space
  - Exponential growth in solution size
- Bounded Policy Iteration (Poupart&Boutilier,03)
  - Fixed solution size (controlled growth)
  - Applied in POMDP & Dec-POMDP
    - Dec-BPI (Bernstein,Hansen&Zilberstein,05) -- optional correlation device may not be feasible in non-cooperative settings
- Contribution:
  - *Policy iteration algorithm* (approximate) for I-POMDPs : generalization of BPI
  - Shows scalability to larger problems

Policy Representation

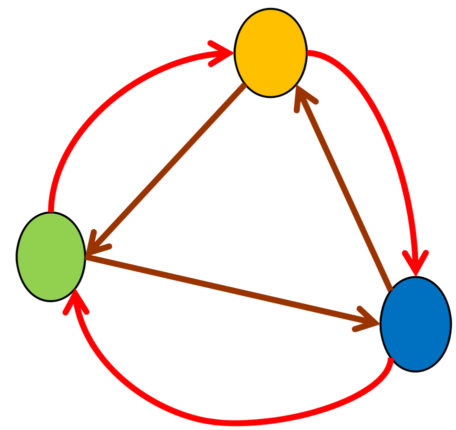
- Possible representation of policy



Tree representation

Node → action

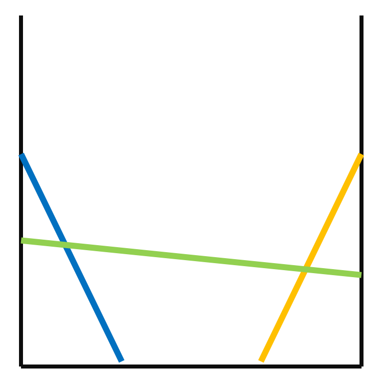
Edge → obs



Finite state controllers (Hansen, 1998)

- Node has an infinite horizon policy rooted at it
- Node has a value vector associated with it which is a linear vector over the entire belief space
- Beliefs are mapped to a node (n) that optimizes the expected reward from that belief:

$$\text{argmax}_n \; b \cdot V^n$$



Finite State Controller

- A finite state controller may be defined as:

$$\mathcal{F}_i = < \mathcal{N}_i, \mathcal{E}_i, \mathcal{L}_i, \mathcal{T}_i >$$

where:

$\mathcal{N}_i$ is the set of nodes in the FSC of agent *i*

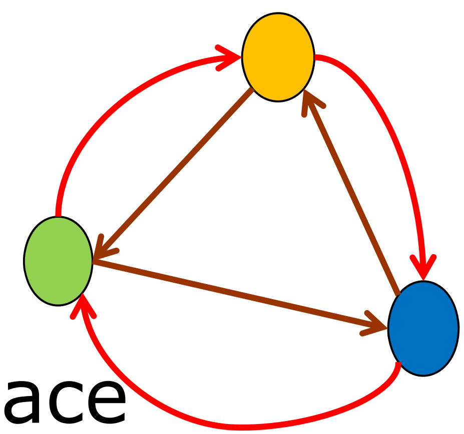
$\mathcal{E}_i$ is the set of edge labels (Ω_i)

$\mathcal{L}_i : \mathcal{N}_i \rightarrow A_i$

$\mathcal{T}_i : \mathcal{N}_i \times A_i \times \Omega_i \rightarrow \mathcal{N}_i$

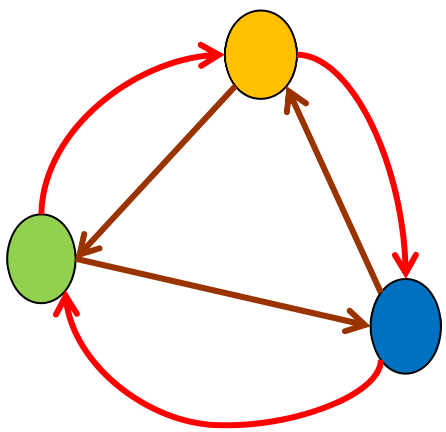
Let: $\mathcal{M}_i : B_i \rightarrow \mathcal{N}_i$

$\mathcal{N}_i$ partitions the entire belief space



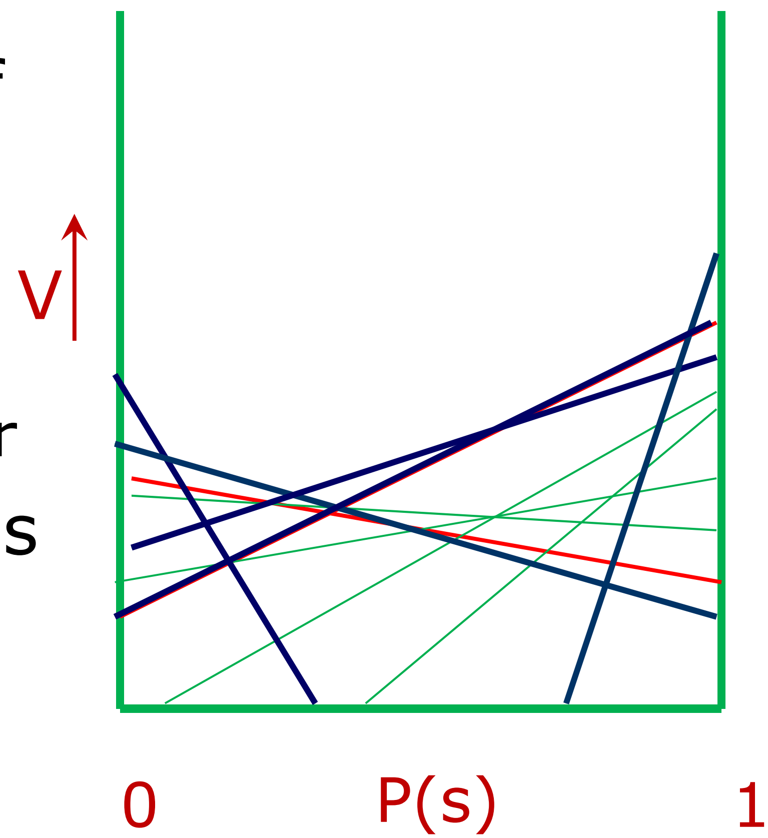
Policy Iteration

- Starting with an initial controller, iterate over two steps until convergence:
  - Policy Evaluation:
    - Evaluate V^n for each node
    - Solve system of linear equations
  - Policy Improvement:
    - Construct a better controller
    - Possibly by adding new nodes



Policy Improvement (Hansen,98)

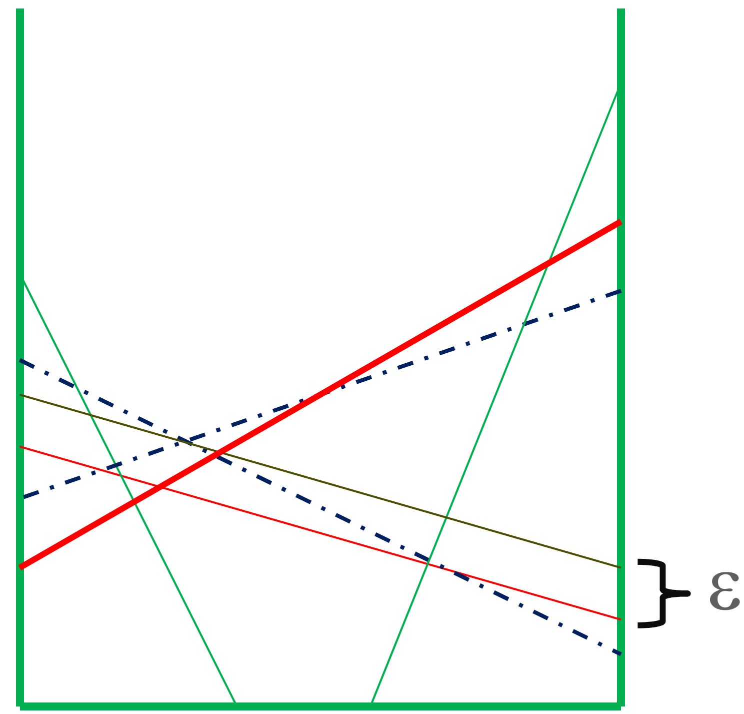
- Apply Backup operator, i.e. construct new nodes with all possible values of action and transition on observation
  - $|A||N|^{|\Omega|}$ new nodes
  - Add them to the controller
- Prune all dominated nodes
- Drawback: Leads to exponential growth in controller size



Example of policy iteration for a POMDP

Bounded Policy Iteration (BPI)
(Poupart&Boutilier,03)

- Instead of performing a complete back up, replace a node with a better node
  - Linear program for partial backup
  - New node is a convex combination of two backed up nodes
  - Changes in controller:



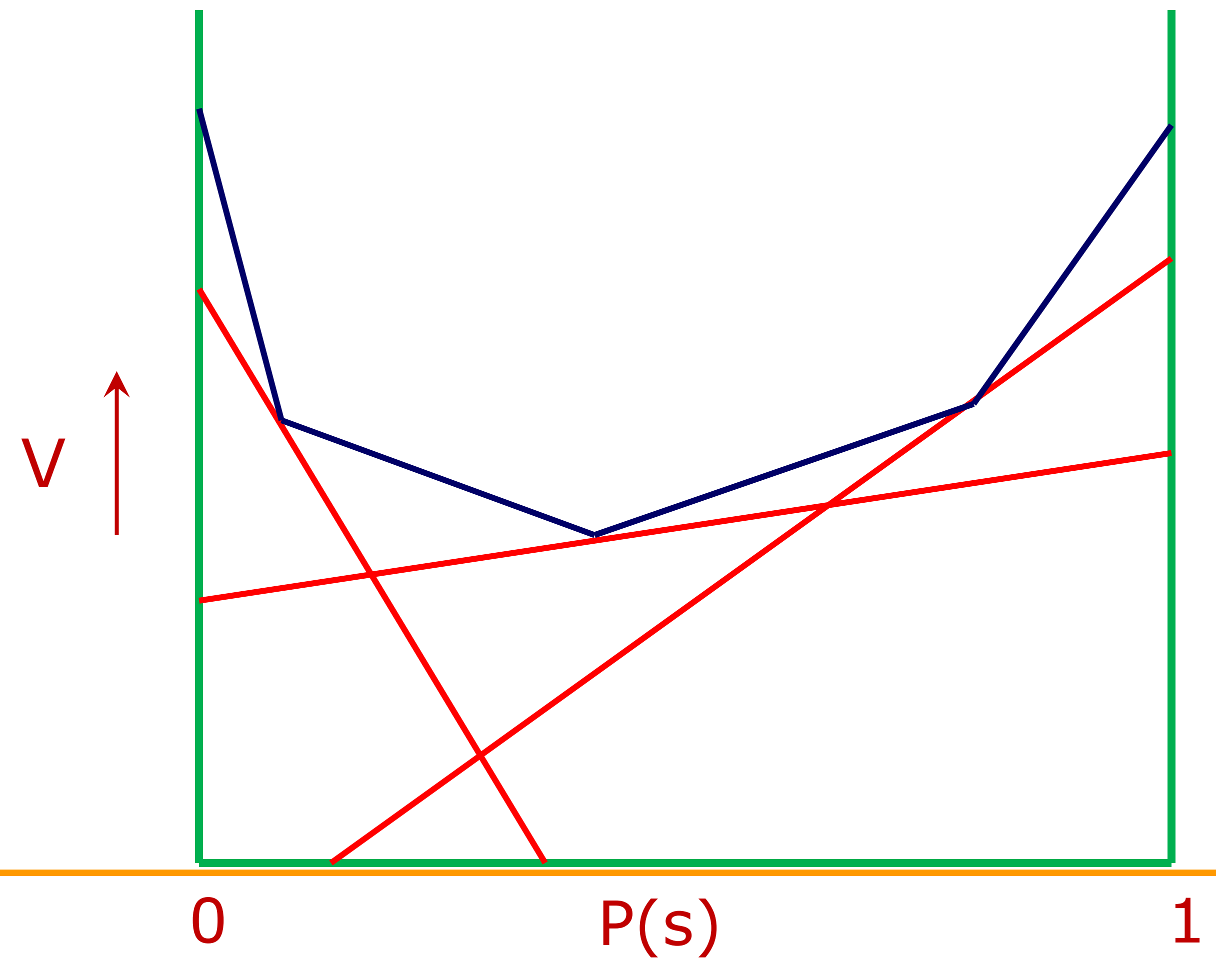
$$\mathcal{F}_i = < \mathcal{N}_i, \mathcal{E}_i, \mathcal{L}_i, \mathcal{T}_i >$$

$$\mathcal{L}_i : \mathcal{N}_i \rightarrow \Delta(A_i)$$ stochastic action policy

$$\mathcal{T}_i : \mathcal{N}_i \times A_i \times \Omega_i \rightarrow \Delta(\mathcal{N}_i)$$ stochastic observation policy

Local Optima

- This form of policy improvement is prone to converging to *local optima*
- When all nodes are *tangents* to backed up nodes: $\varepsilon = 0$, no improvement
- Escape technique suggested by Poupart & Boutilier (2003) in BPI

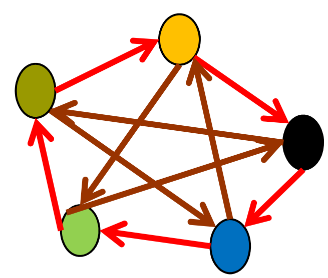


I-POMDP Generalization: Nested Controllers

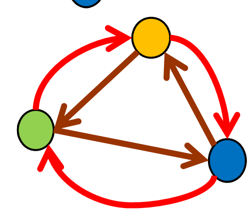
- Nested Controllers: Analogous to nested beliefs
  - Embed recursive reasoning
- Starting from level 0 upwards, for each level *l*, construct a *Finite state controller* for each frame of each agent ($\mathcal{F}_{i(j),l}$)
  - For convenience of representation, let's assume two agents and each one frame for an agent at each level

$$\mathcal{F}_{i,l}: \qquad IS_{i,l} = S \times N_{j,l-1}$$

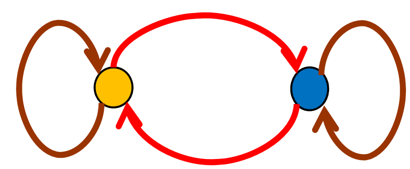
Agent i's level 2 controller:



Agent j's level 1 controller:



Agent i's level 0 controller:



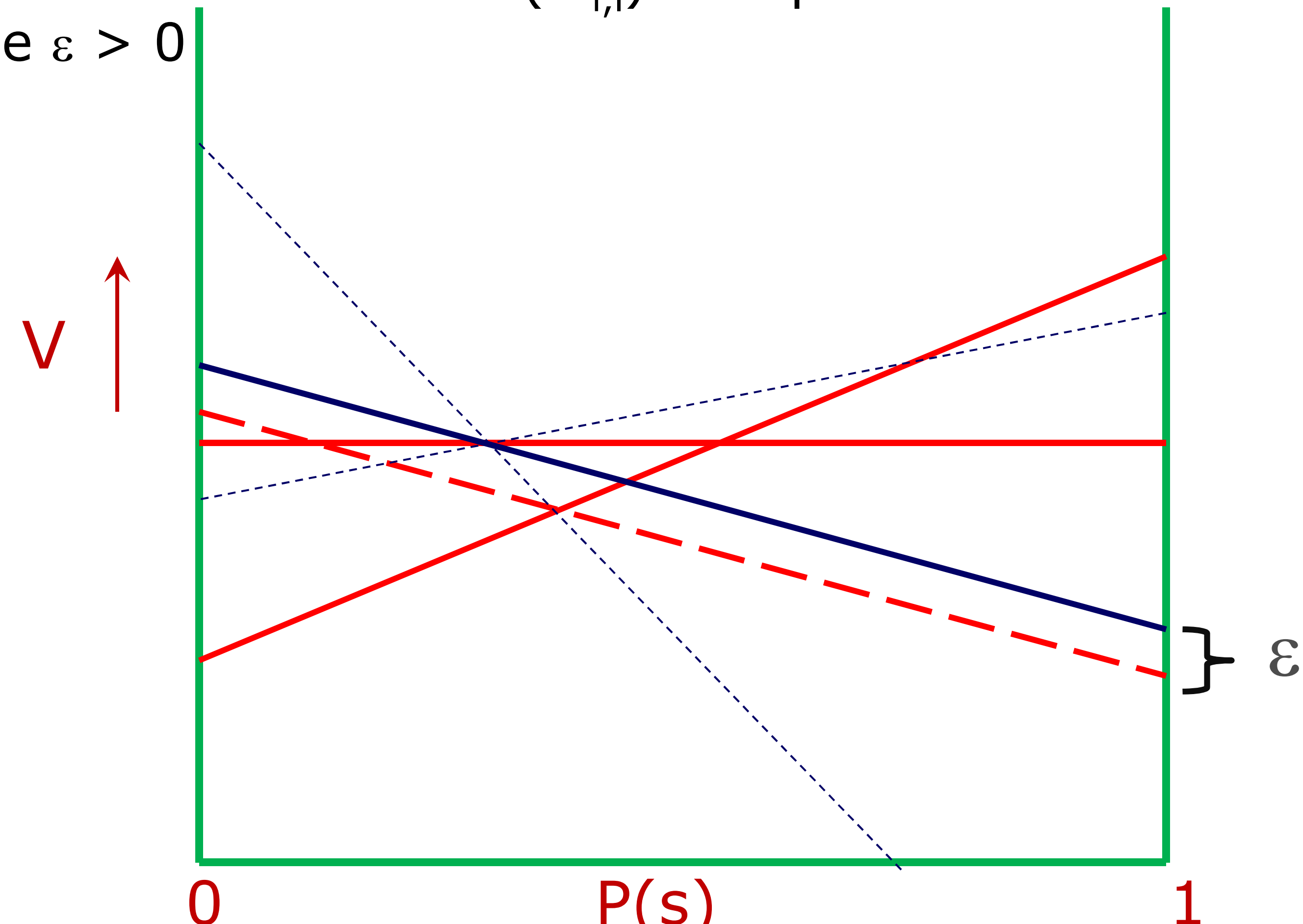
Interactive BPI: Policy Evaluation

- Compute the value vector of each node using the estimate of other agent's model by solving a system of linear equations:
- For each $n_{i,l}$, and interactive state, is=(s, $n_{j,l-1}$), solve:

$$V^{n_{i,l}}(s, n_{j,l-1}) = \sum_{a_i} Pr(a_i|n_{i,l}) \sum_{a_j} Pr(a_j|n_{j,l-1}) \Big\{ R(s, a_i, a_j) + \gamma \sum_{o_i} \sum_{s'} \sum_{n'_{j,l-1}} T_i(s, a_i, a_j, s') O_i(s', a_i, a_j, o_i) \times \sum_{o_j} O_j(s', a_i, a_j, o_j) Pr(n'_{j,l-1}|n_{j,l-1}, a_j, o_j) \times \sum_{n'_{i,l}} Pr(n'_{i,l}|n_{i,l}, a_i, o_i) V^{n'_{i,l}}(s', n'_{j,l-1}) \Big\}$$

I-BPI: Policy Improvement

- Pick a node ($n_{i,l}$) and perform a *partial backup* using LP to construct another node ($n'_{i,l}$) that pointwise dominates $n_{i,l}$ by some $\varepsilon > 0$



New vector dominates old vector by ε and hence replaces it

I-BPI: Policy Improvement

- Pick a node ($n_{i,l}$) and perform a *partial backup* using LP to construct another node that point-wise dominates $n_{i,l}$ by some $\varepsilon > 0$

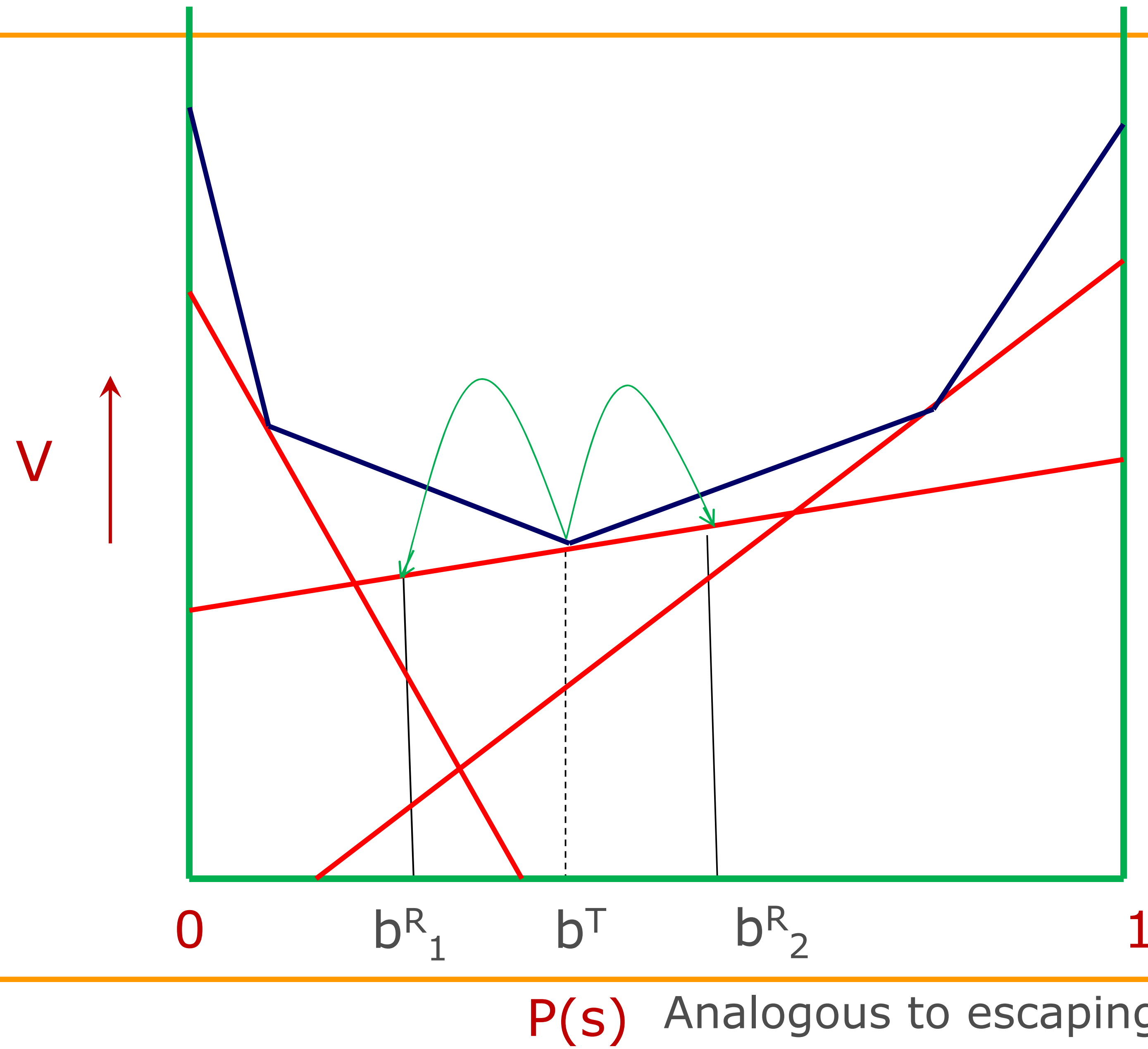
Objective Function: maximize ε

Variables: $Pr(a_i)\ \forall a_i; Pr(n'_{i,l}|a_i, o_i)\ \forall a_i, o_i, n'_{i,l} \in \mathcal{N}_{i,l} - \{n_{i,l}\}$

Constraints: $\sum_{a_i} Pr(a_i) = 1 \quad \forall a_i, o_i : \sum_{n'_{i,l}} Pr(n'_{i,l}|a_i, o_i) = Pr(a_i)$

$$V^{n_{i,l}}(s, n_{j,l-1}) + \epsilon \le \sum_{a_i} Pr(a_i) \sum_{a_j} Pr(a_j|n_{j,l-1}) \Big\{ R(s, a_i, a_j) + \gamma \sum_{o_i} \sum_{s'} \sum_{n'_{j,l-1}} T_i(s, a_i, a_j, s') O_i(s', a_i, a_j, o_i) \times \sum_{o_i} O_j(s', a_i, a_j, o_j) Pr(n'_{j,l-1}|n_{j,l-1}, a_j, o_j) \times \sum_{n'_{i,l}} Pr(n'_{i,l}|a_i, o_i) V^{n'_{i,l}}(s', n'_{j,l-1}) \Big\}$$

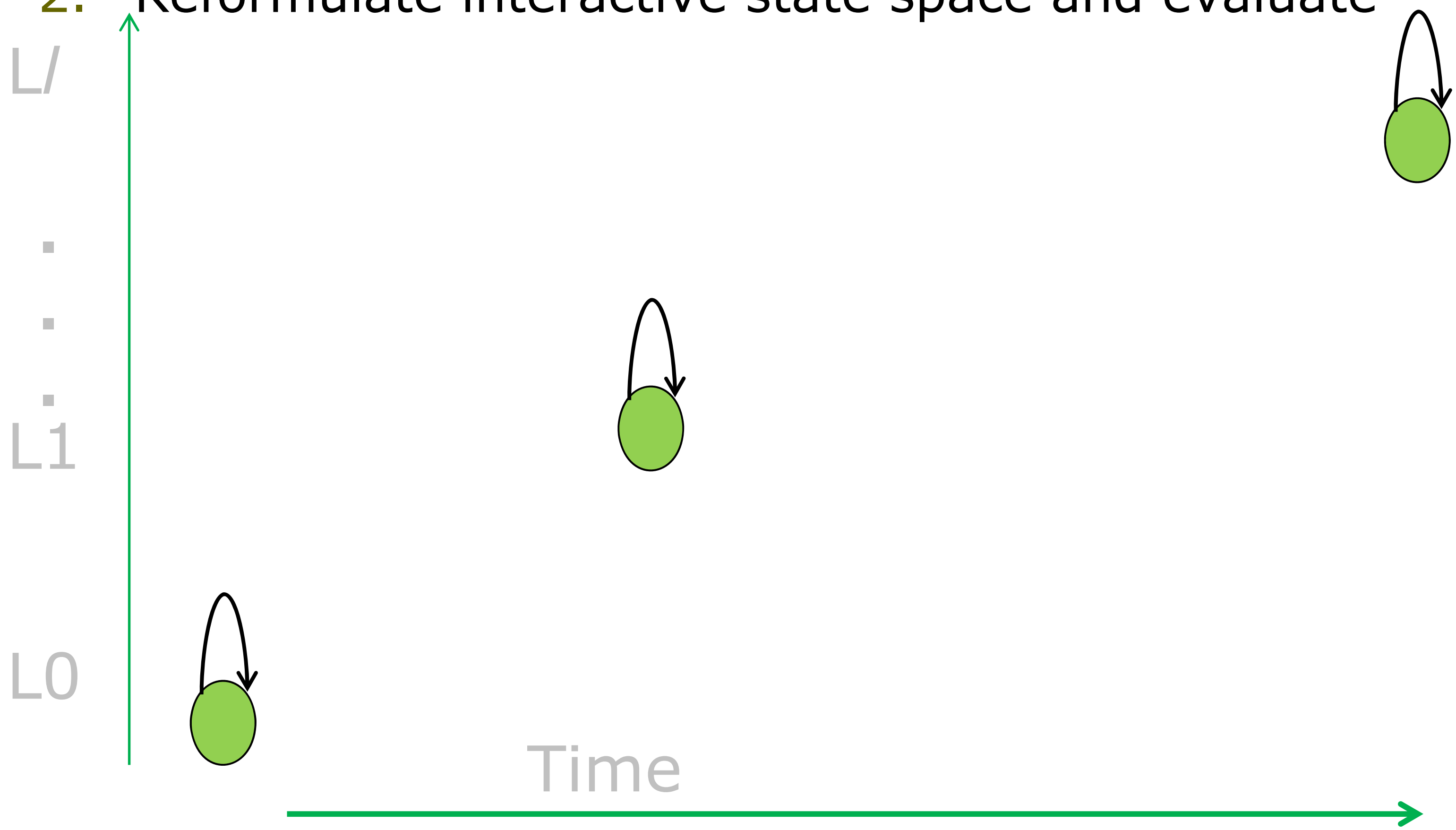
Escaping Local Optima



Analogous to escaping for POMDPs

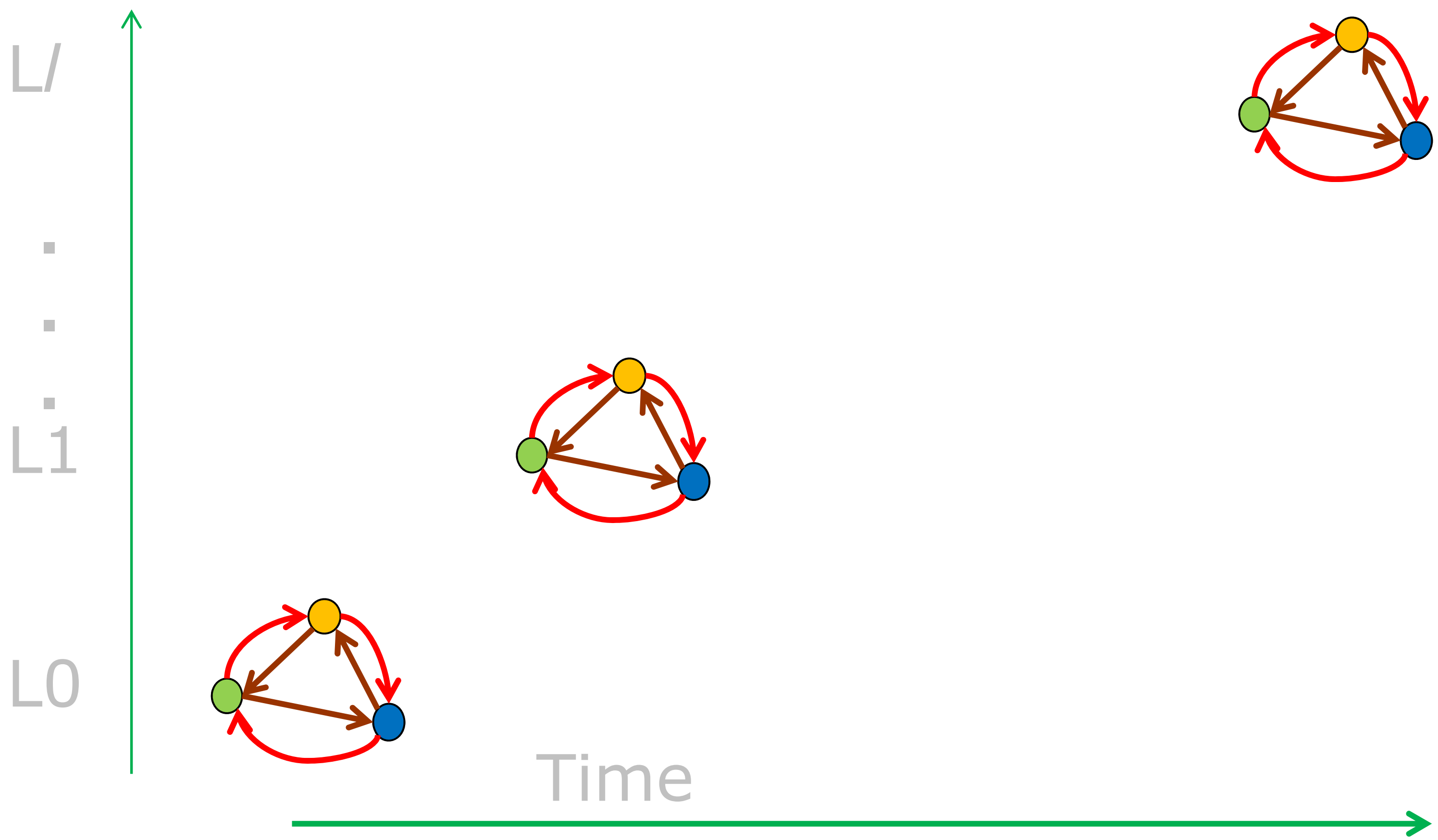
Algorithm: I-BPI

1. Starting from Level 0 up to Level *l*, construct a 1 node controller for each level with a random action and transition to itself.
2. Reformulate interactive state space and evaluate



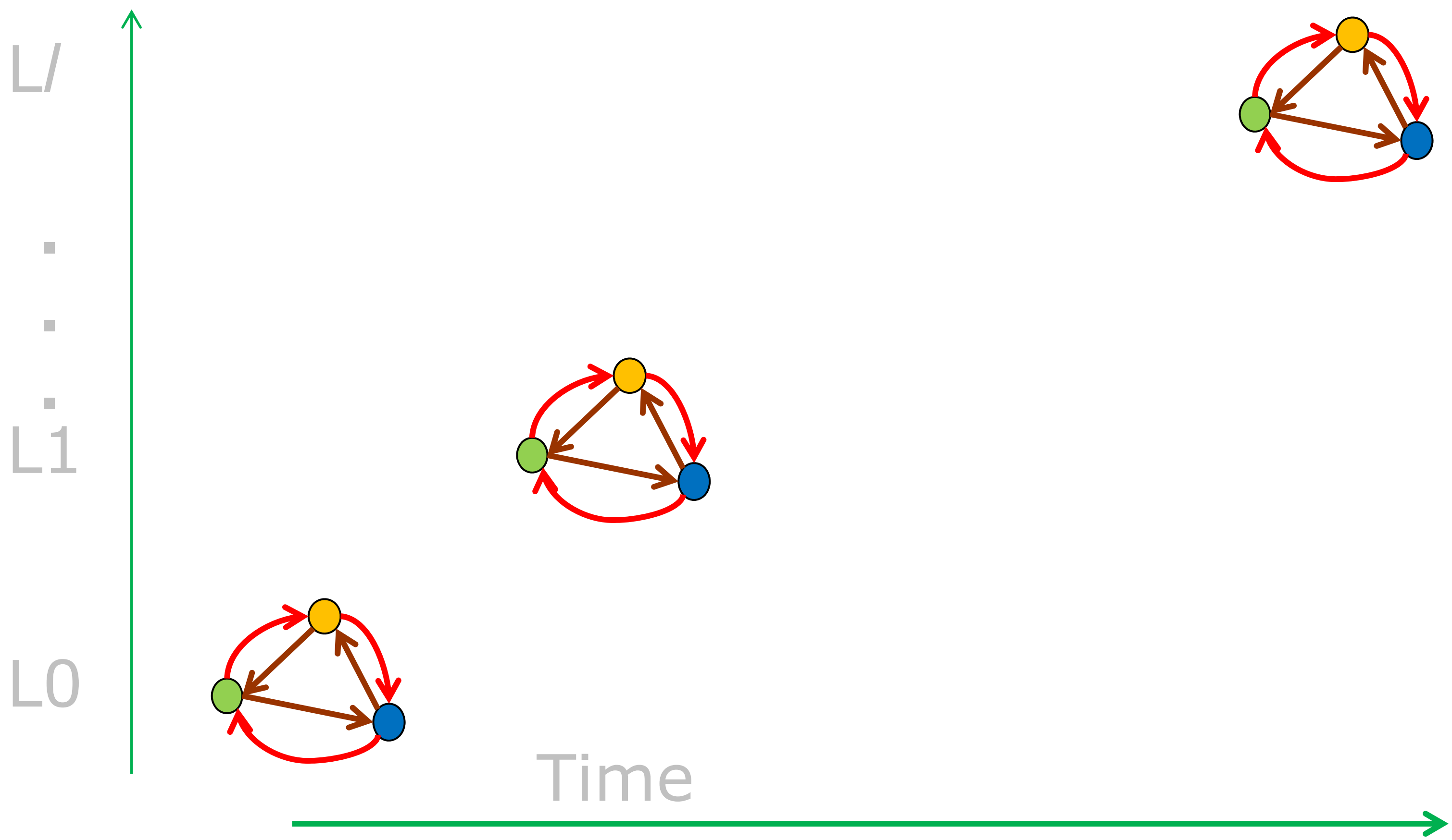
Algorithm: I-BPI

3. Starting from Level 0 up to Level l, perform 1 step of back up operator. Max $|A_{i(j)}|$ nodes



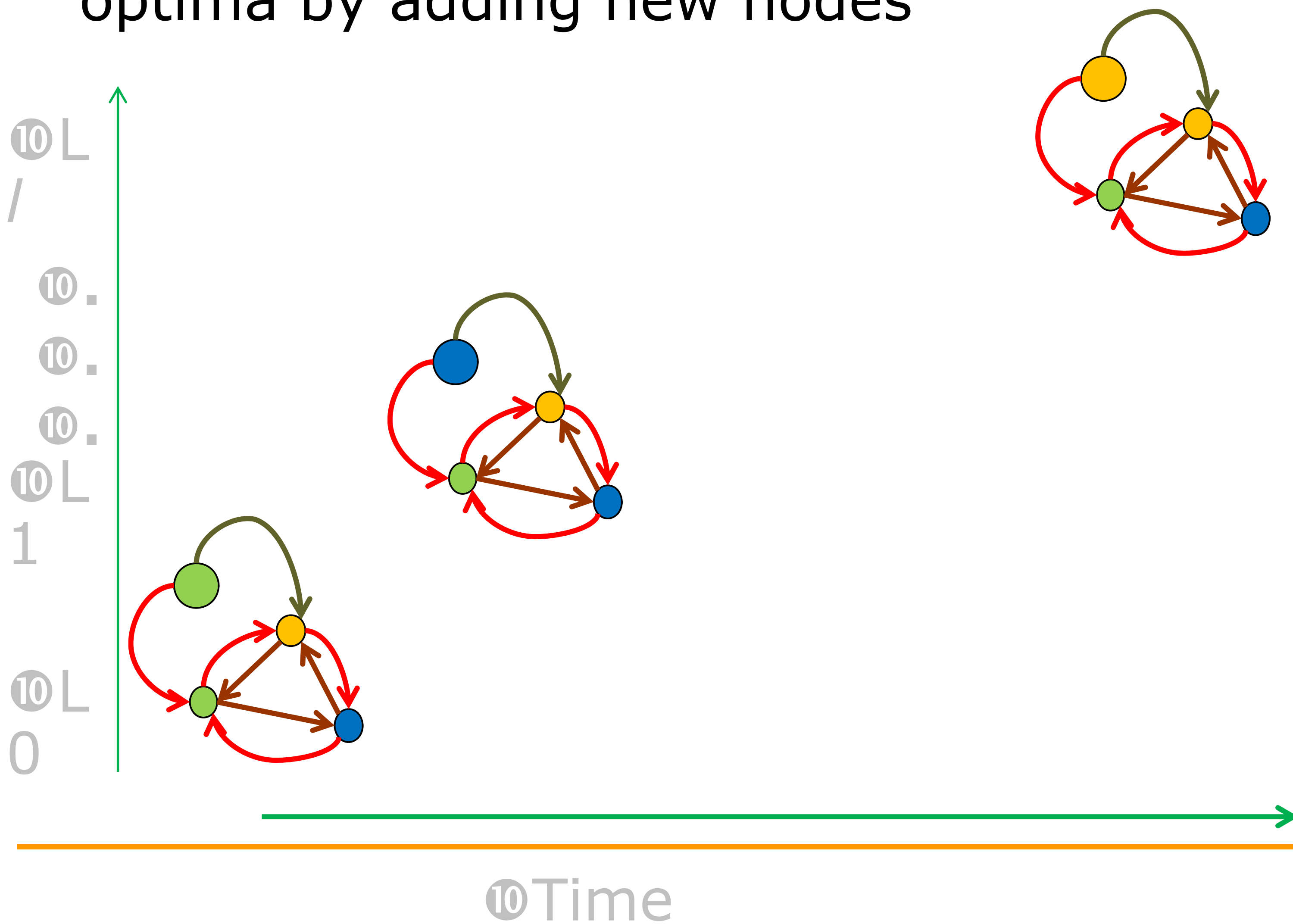
Algorithm: I-BPI

4. Starting from Level 0 up to Level *l*, reformulate IS space, perform policy evaluation followed by policy improvement at each level



Algorithm: I-BPI

5. Repeat step 4 until convergence
6. If converged, push nested controller out of local optima by adding new nodes



Evaluation

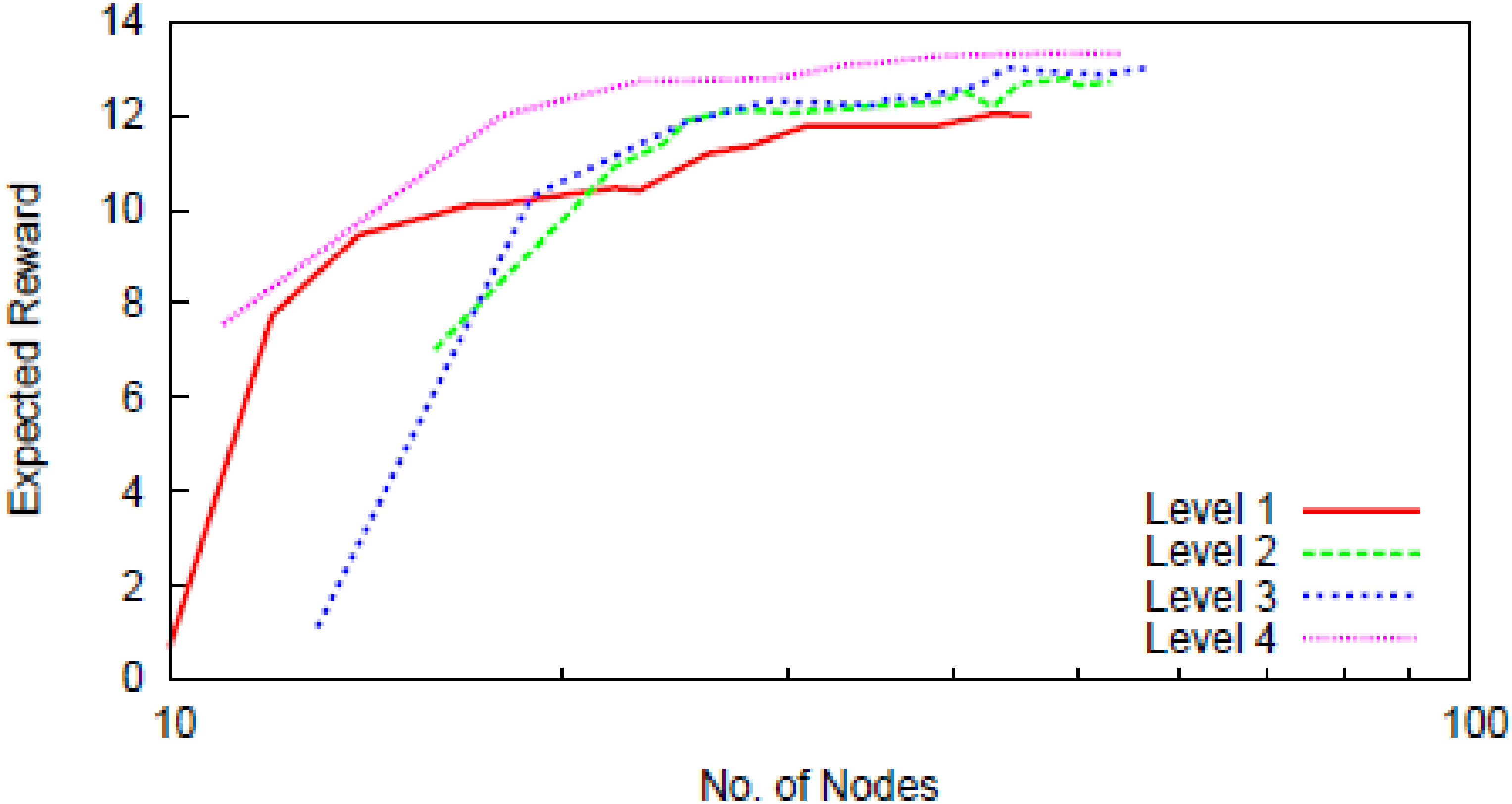
Problem	Level	Method	Time(s)	Avg. Rwd
Mult. tiger	1	I-BPI	69	11.34
		I-PBVI	2,000	5.34
	2	I-BPI	1,109	12.48
		I-PBVI	696	3.15
	3	I-BPI	3,533	13.00
	4	I-BPI	3,232	13.22
		I-BPI	15	20.22
	3	I-BPI	117	21.28
	4	I-BPI	157	21.36
AUAV*	1	I-BPI	7,979	74.08
Money Laun.*	1	I-BPI	1,354	-156.21

AUAV: 81 states, 5 actions, 4 observations

Money Laundering: 99 States, 11 actions, 9 Observations

Scales to larger problems...

Evaluation



Simulations results for multiagent tiger problem showing results obtained by simulating performance of agent controllers of various sizes for levels 1 – 4

Discussion on I-BPI

- Advantages of I-BPI
  - Is significantly quicker and scales to large problems (100s of states, tens of actions and observations)
  - Mitigates curse of history and curse of dimensionality
  - Improved solution quality
- Limitations
  - Prone to local optima
    - Escape technique may not work for certain local optima
  - Not entirely free from curses of history and dimensionality
- Future Work
  - Scale to even larger problems and more agents
  - Mealy machine implementation for controllers (Amato et al. 2011)

Summary of I-POMDPs

- **I-POMDPs:** A framework for decision making in uncertain multiagent settings
- Analogous to POMDPs but with an enriched state space
  - interactive beliefs
- Uses decision-theoretic solution concept
  - MEU
- For infinitely nested beliefs, look for fixed points
- Intractability of I-POMDPs
  - Curse of dimensionality: belief space complexity
  - Curse of history: policy space complexity

- **Exact: Equivalence classes of interactive states**
  - Lossless transformation of IS into a discrete space

- **Approximation 1: Interactive Particle Filter**
  - Randomized algorithm for approximating the nested belief update
  - Partial error bounds
- **Approximation 2: Interactive Point-based Value Iteration**
  - Algorithm for partial update of the value function
  - Linear program not needed
  - Partial and loose error bounds
- **Approximation 3: Interactive Bounded Policy Iteration**
  - Update the nested policy directly
  - Represent policies using finite-state machines
  - Local optima

- **Graphical model counterpart: Interactive Dynamic Influence Diagrams (I-DIDs)**

Applications of I-POMDPs

- Adversarial reasoning in the context of money laundering (Ng et al., 2010)
- Behavioral modeling of recursive reasoning data in Centipede Game (Doshi et al., 2010)
- Predicting opponent strategies in Lemonade Stand Game (Wunder et al., 2011)
- Learning from human teachers in the context of robotics (Woodward & Wood, 2012)

Generalizations or specializations

- Trust enabled I-POMDPs (Seymour & Peterson, 2009)
  - Models of the other agent include trust levels as well
- Parameterized I-POMDPs (Wunder et al., 2011)
  - Distribution over lower-level models is learned parameter from agent population
- Intention-aware POMDPs (Hoang & Low, 2012)
  - Specialization: Assumes that the other agent observes its state perfectly
  - Hierarchy reduces to a nested MDP
- Reinforcement learning in I-POMDPs (Ng et al., 2012)
  - Bayes-adaptive RL

Application 1: Adversarial reasoning in money laundering

- Money laundering domain
  - Red team (money launderers) hold money in accounts
    - {dirty pot, bank accounts, securities, shell companies,...}
  - Blue team (law enforcement) must sense the money
    - {no sensors, bank accounts, shell companies, casinos,...}
  - Red team's actions involve placing, layering or integrating the money, and observing the blue team's sensors
  - Blue team's actions involve placing the sensors, and observing reports and sensor information

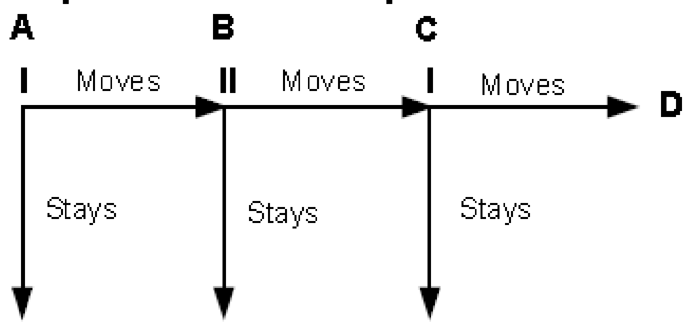
- $|S| = 99$, $|A_i| = 9$, $|A_j| = 4$, $|\Omega_i| = 11$, $|\Omega_j| = 4$

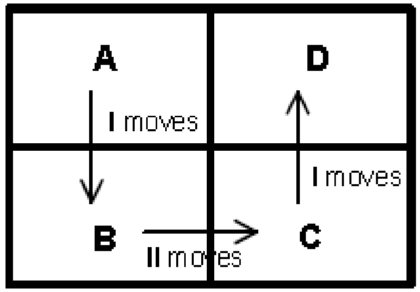
Application: Adversarial reasoning (contd.)

- Approach
  - Formulate a level 1 I-POMDP for each team
  - Combine I-PF with a sampled reachability tree for both agents to generate separate policy trees for red and blue teams with initial beliefs
- Experiments
  - Laundering game was played by simulating the two teams' policy trees across 50 trials
  - For most settings of particles and agent solution horizons, red team has the advantage!
  - Blue team wins when each team models the opponent at just horizon 1

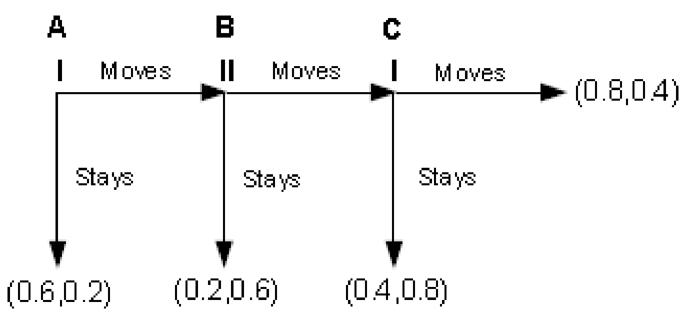
Application 2: Behavioral modeling of recursive reasoning data

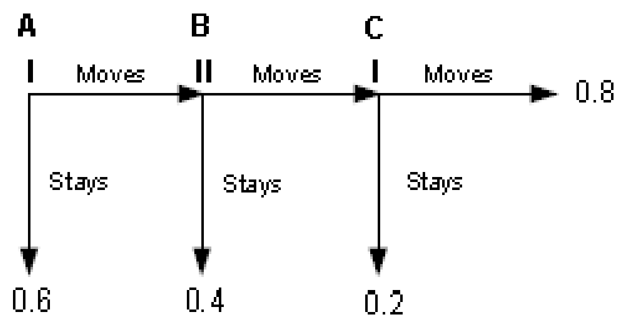
- Two large studies involving human subjects on levels of recursive reasoning
  - Two-player alternating-move game with complete and perfect information





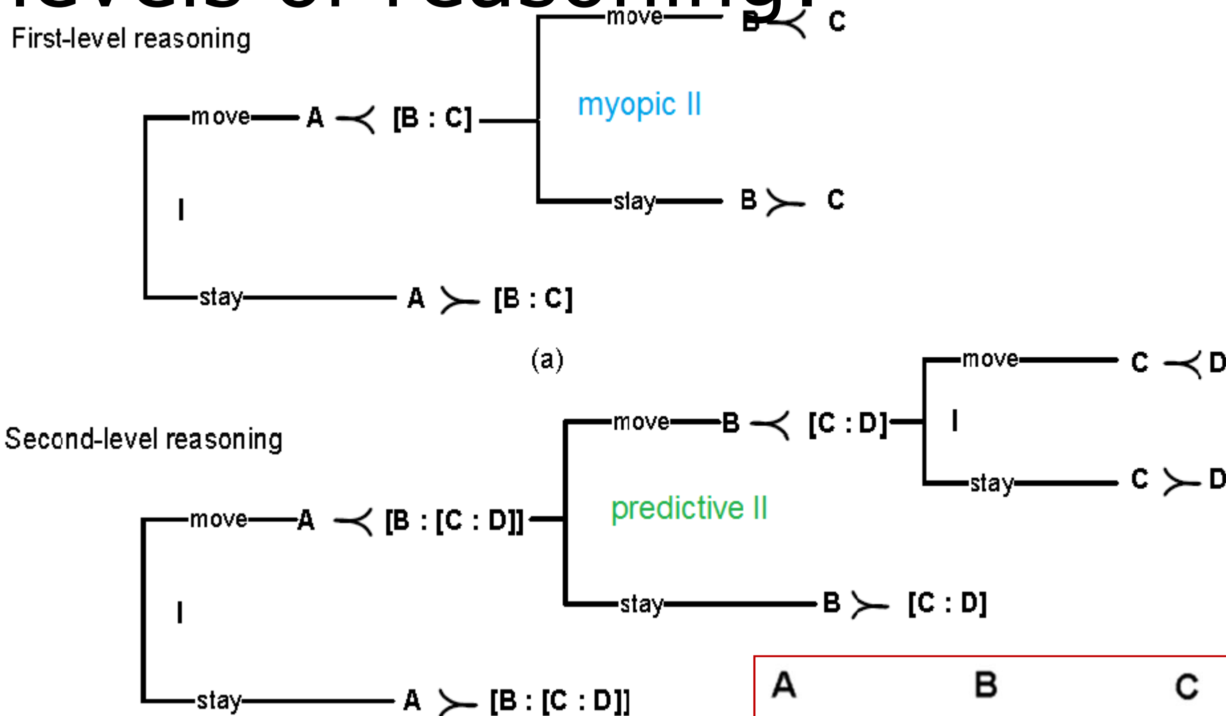
  - General sum game & fixed sum game



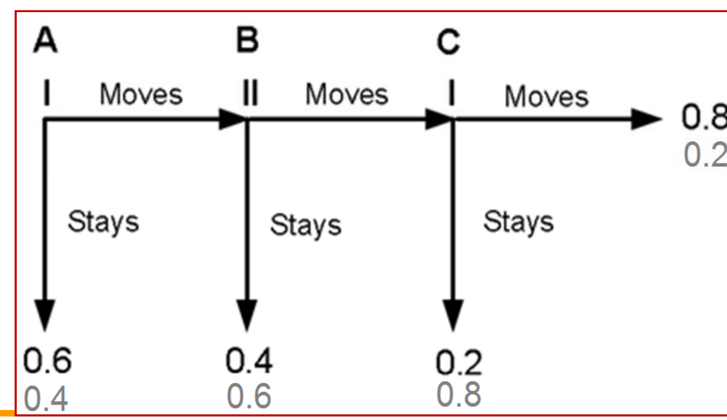


Experimental studies (contd)

- Two levels of reasoning:



- Opponent type
  - myopic
  - predictive



Computational model: Interactive POMDP

- Modeling behavioral data gathered from study
- Multiagent setting
- State space includes other agents' models
- A finitely nested I-POMDP of agent *i* with a strategy level *l* interacting with another agent *j*, is defined as

  $\langle IS_{i,l}, A, \Omega_i, T_i, O_i, R_i \rangle$

  - $IS_{i,l}$: Interactive states, defined as $IS_{i,l} = S \times M_{j,l-1}$ where
    $M_{j,l-1} = \Theta_{j,l-1} \cup SM_j$ for $l \geq 1$, and $IS_{i,0} = S$ where S is states of physical environment
    - $\theta_{j,l-1}$: intentional models of agent j, defined as $\theta_{j,l-1} = \langle b_{j,l-1}, \hat{\theta}_j \rangle$ where $b_{j,l-1}$ is j's level *l-1* belief, $\hat{\theta}_j$ is the frame
    - SM_j : subintentional models of j

Empirically informed I-POMDP

- I-POMDP$_{i,2}$:
  - Interactive States:
    - physical state space S = {A,B,C,D} (perfectly observable)
    - model set $\Theta_j = \{\theta_{j,1}, \theta_{j,0}\}$
      - $\theta_{j,1}$ is the level 1 predictive model of the opponent
      - $\theta_{j,0}$ is the level 0 myopic model of the opponent
  - Action:
    - $A_i = A_j$ = {Stay, Move} (deterministic)
  - Observation:
    - Ω_i = {Stay, Move}

Empirically informed I-POMDP (contd.)

- Descriptive decision model
  - Subjects made non-normative choice
    - Rationality errors observed
  - Quantal response model

$$q(a_i^* \in A_i) = \frac{e^{\lambda .U(b_i, a_i^*)}}{\sum_{a_i \in A_i} e^{\lambda .U(b_i, a_i)}}$$

    - $q(a_i \in A_i)$ is the probability assigned to action a_i by the model
    - $U(b_i, a_i)$ is the utility for i performing the action a_i given its belief b_i
    - λ controls how responsive is the model to value differences

Empirically informed I-POMDP (contd.)

- Descriptive judgment model
  - Subjects learned from previous game
    - learning is slow
    - subjects could be underweighting the evidence that they observe
  - Updating belief:

$$\frac{\Pr(\theta_{j,1}|o_i)}{\Pr(\theta_{j,0}|o_i)} = \frac{\Pr(\theta_{j,1})}{\Pr(\theta_{j,0})}\left\{\frac{\Pr(o_i|\theta_{j,1})}{\Pr(o_i|\theta_{j,0})}\right\}^{\gamma}$$

    - *Underweighting when $\gamma<1$*
    - Overweighting when $\gamma>1$
    - Normative updating when $\gamma=1$
  - γ controls the learning rate

Learning

- Two parameters to learn
  - γ controls learning rate
  - λ controls non-normative choice
- Gradient Descent
  - Error function: the inverse of the data likelihood

$$X = -\sum_{i=1}^{|Subj|}\sum_{g=1}^{N}\log q(a_i^*|A_i)$$

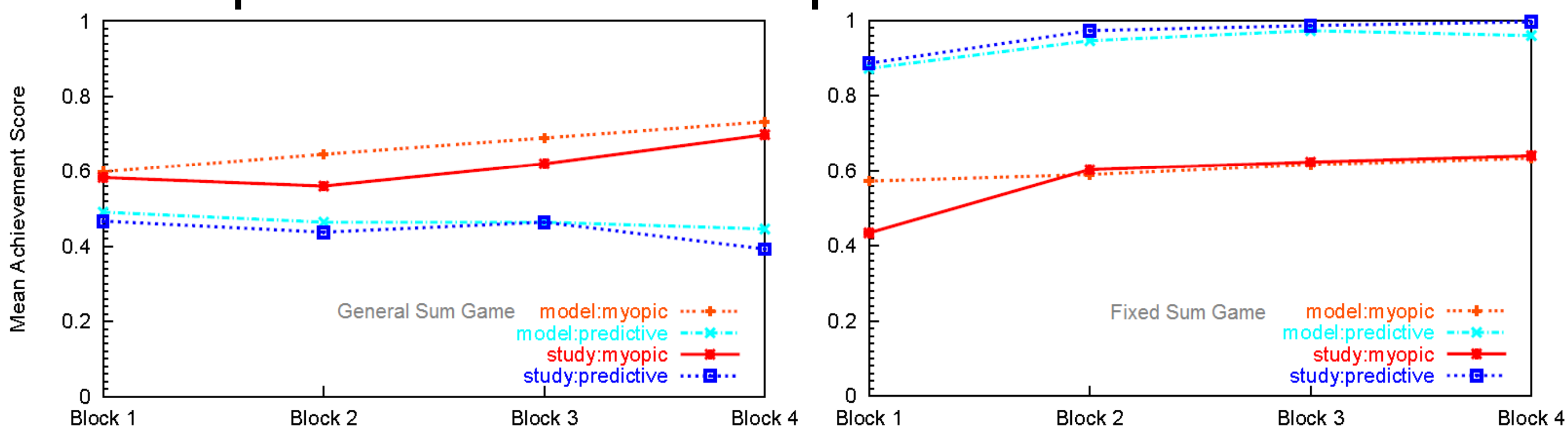
$$= -\sum_{i=1}^{|Subj|}\sum_{g=1}^{N}\log \frac{e^{\lambda\times U(b_{i,2}^{g},a_i^*)}}{\sum_{a_i\in A_i} e^{\lambda\times U(b_{i,2}^{g},a_i)}}$$

  - a_i^* is the action from A_i selected by subject *i* in the *g*th game

Results

- We utilized the learned values to parameterize the underweighting and quantal response models within the I-POMDP

- Comparison of model predictions with actual



Application 3: Learning from a human teacher

- Domain
  - Agent (robot) learning interactively from a non-technical human teacher
    - Learning by demonstration
    - Learning by reinforcements
  - Interaction consists of signals generated by the agent and teacher
    - Examples of signals: words, gestures, facial expressions, eye gaze, rewards, ...

Application to learning (contd.)

- Approach
  - Model the learning problem as a I-POMDP
  - All signals from the teacher and environment are modeled as agent’s observations
  - Teacher is modeled in the agent’s IS
    - Teacher’s belief about the state of the world, about agent’s variables and beliefs are maintained
  - Action selection accounts for the predicted future actions of the teacher

- Benefits of the approach
  - Principled formulation of the problem
  - Complex interactions possible due to nested modeling

Application to learning (contd.)

- Benefits (contd.)
  - Acting to reduce inconsistency in its modeling of the teacher's modeling
    - Interrupt the teacher to request a change in teaching subject
    - Ask a clarification whether the previous action of the teacher was about a different topic
    - Issue a correction to the teacher about the topic of the question that the agent had asked

Brief digression: Cooperation

- Multiple agents share a common reward function (team of agents)
- Common initial belief over physical states
- Popular framework for cooperative decision making
  - Decentralized POMDPs

Decentralized POMDP

Definition of a DEC-POMDP for 2 agents

$$\langle S, A, T, \Omega, O, R, OC \rangle$$

A is the set of joint actions of both agents

T is the transition function given joint actions. The transitions may be independent of other's actions

Ω is the set of joint observations

O is the joint observation function. Observations may be local and independent

R is the reward function which is identical for each agent

Decentralized POMDP

Objective of a DEC-POMDP is to compute a joint policy which optimizes the collective reward for all agents. A joint policy is a tuple of local policies $\pi = \langle \pi_i, \pi_j \rangle$

Solving a DEC-POMDP is a NEXP-Complete problem

Each local policy, π_i , is a mapping from the agent's local history of observations to its actions which optimize the agent's reward

Specializations

Markov team decision problem (MTDP)

- Observations may be local and independent:
  $$O(\langle o_1, o_2 \rangle \mid a_1, a_2, s) = O_1(o_1 \mid a_1, s).O_2(o_2 \mid a_2, s)$$
- Rewards of each agent are identical (team)

Networked distributed POMDP (ND-POMDP)

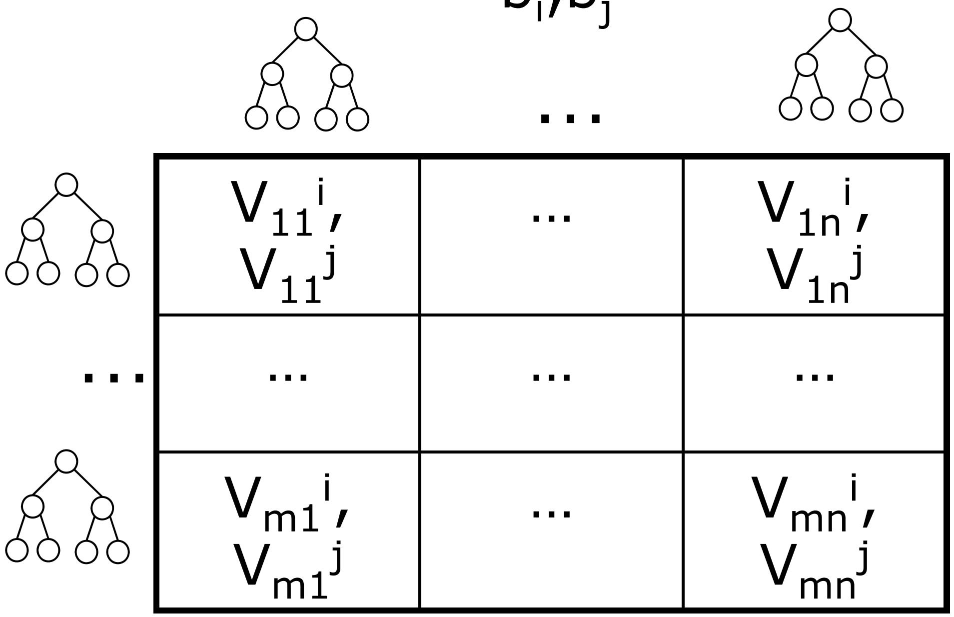
- Both transitions and observations are local and independent
  $$T(\langle s_1', s_2' \rangle \mid a_1, a_2, \langle s_1, s_2 \rangle) = T_1(s_1' \mid a_1, s_1).T_2(s_2' \mid a_2, s_2)$$
- Interaction between agents is through the rewards. Agent’s rewards are influenced by some of the other agents (neighborhood)

Solving DEC-POMDP

Naive approach

Simply convert to a normal form game and use iterated elimination of dominant policies or choose Nash equilibrium

b_i, b_j



	…	
V_{11}^i, V_{11}^j	…	V_{1n}^i, V_{1n}^j
…	…	…
V_{m1}^i, V_{m1}^j	…	V_{mn}^i, V_{mn}^j

Solving DEC-POMDP

Naive approach

Simply convert to a normal form game and use iterated elimination of dominant policies or choose Nash equilibrium

Not a good idea!!
Number of policies doubly exponential
in the number of horizons

DP in DEC-POMDP

Generalize DP operator of POMDP to DEC-POMDP

- Prune policy sets of both agents simultaneously using iterated elimination of dominated policies

- Remove a policy if it is not optimal at any multiagent belief. A multiagent belief of an agent is a distribution over the space of physical states and all policies of other agent

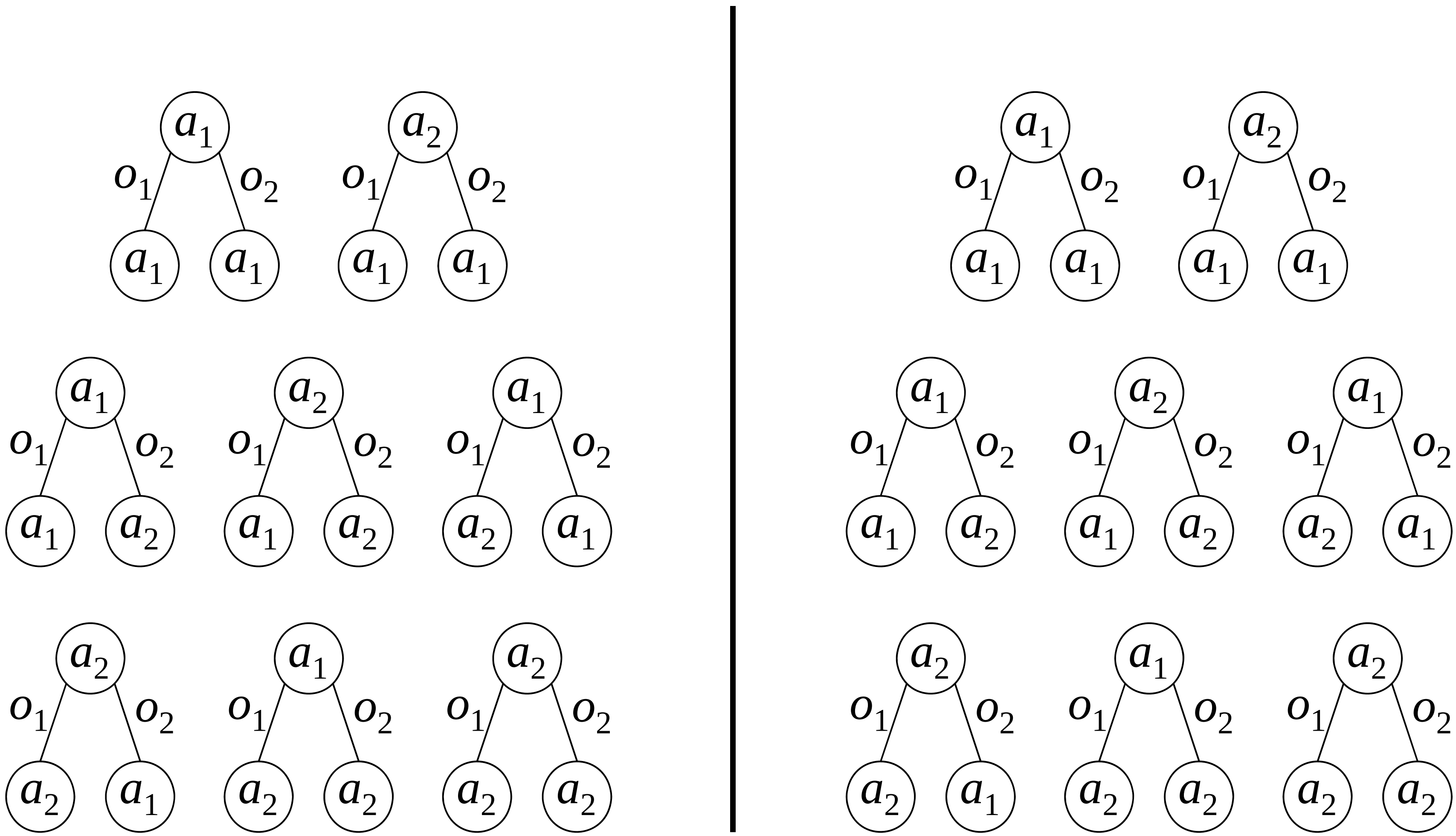
DP in DEC-POMDP

Start with horizon 1 policies

agent *i*	agent *j*
a_1 a_2	a_1 a_2

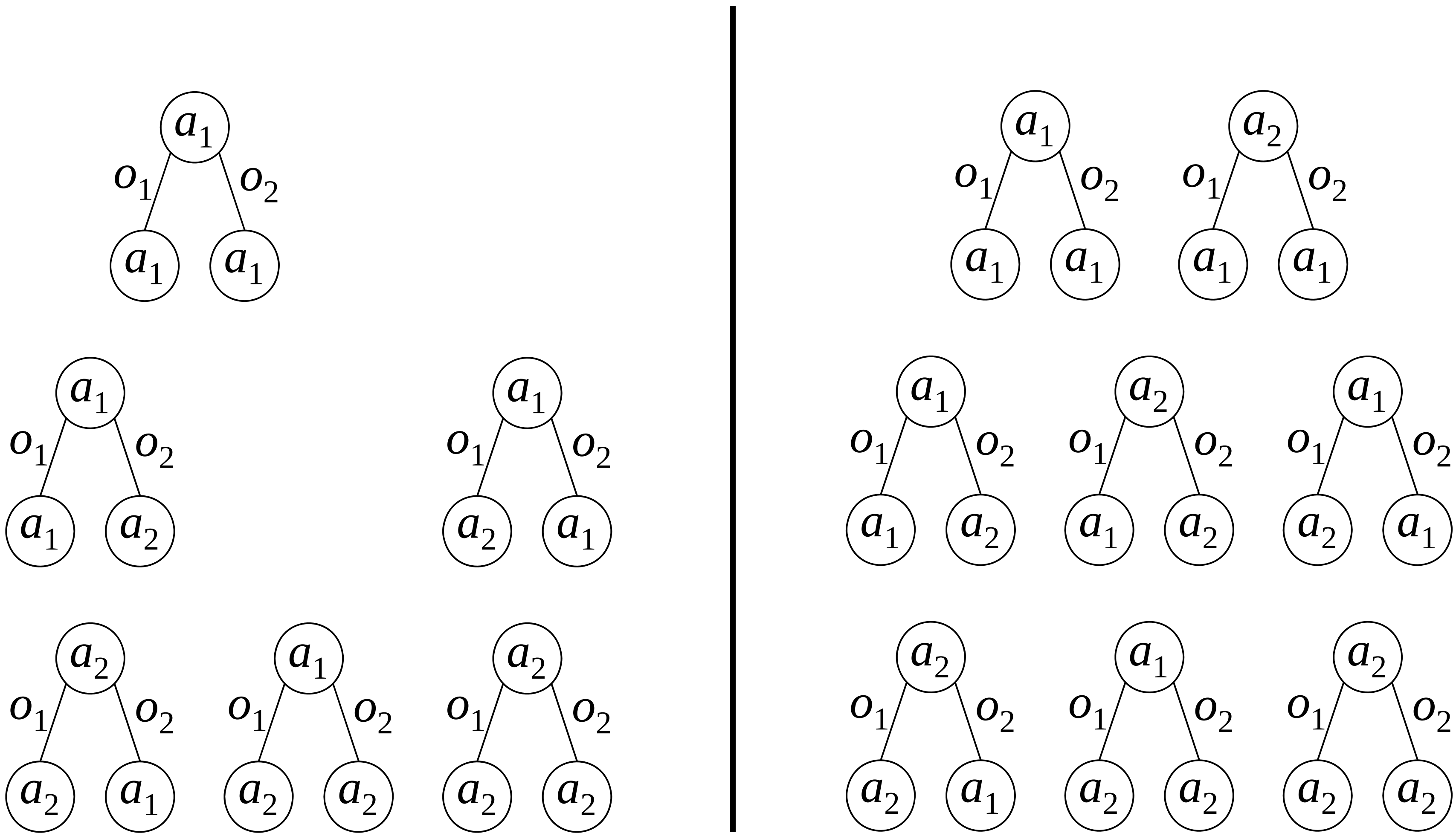
DP in DEC-POMDP

Perform an exhaustive backup



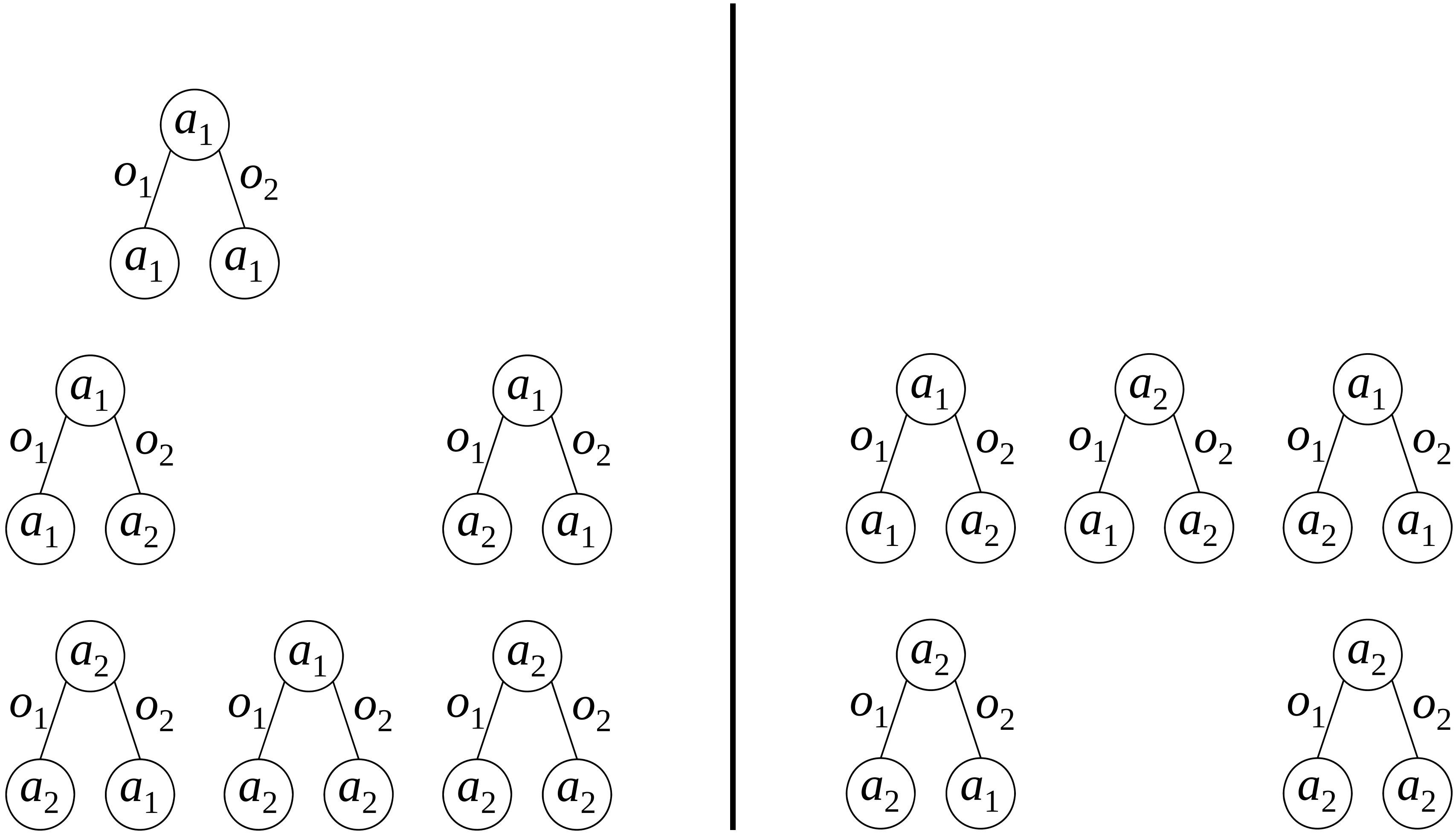
DP in DEC-POMDP

Eliminate dominated policies – iteration 1



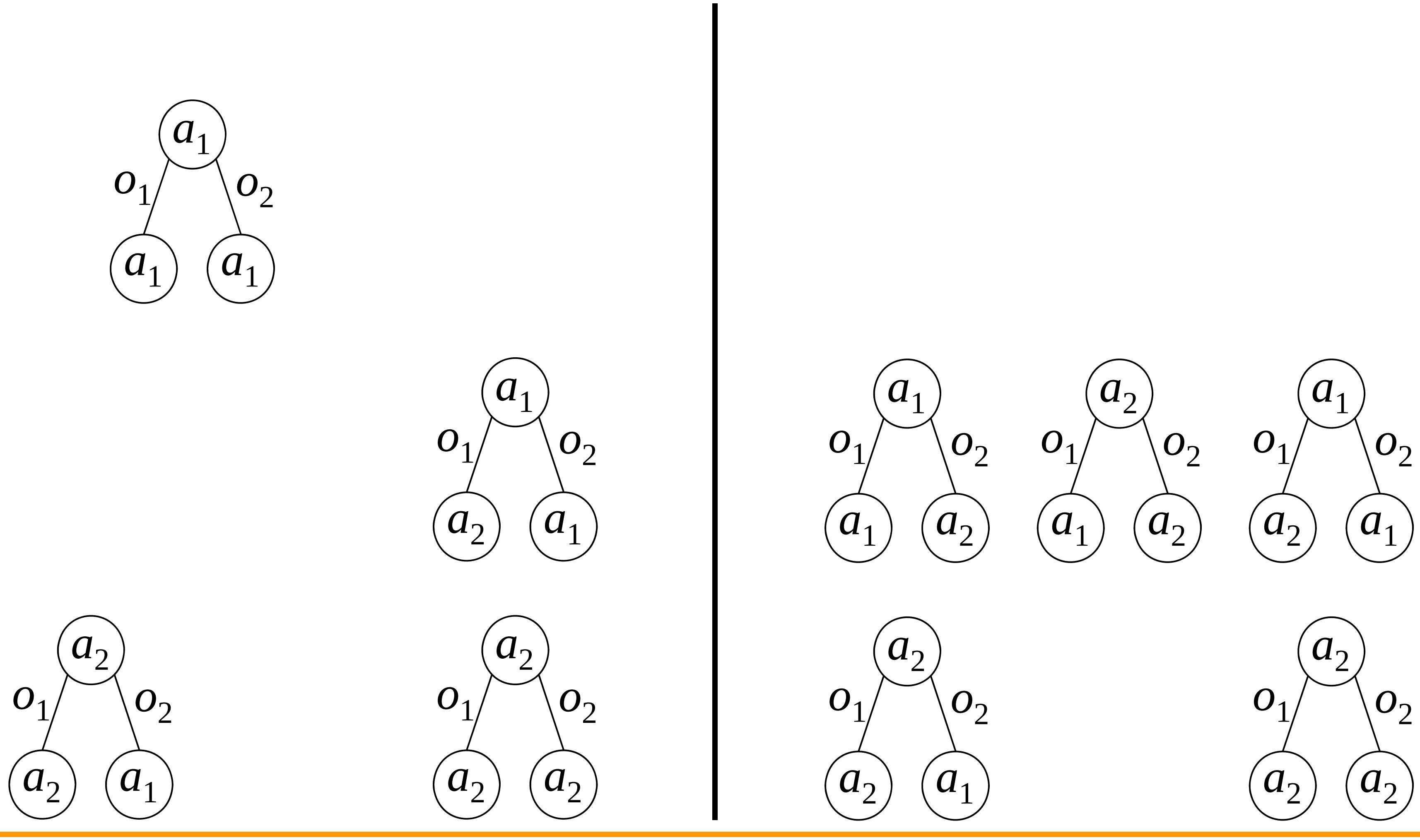
DP in DEC-POMDP

Eliminate dominated policies – iteration 2



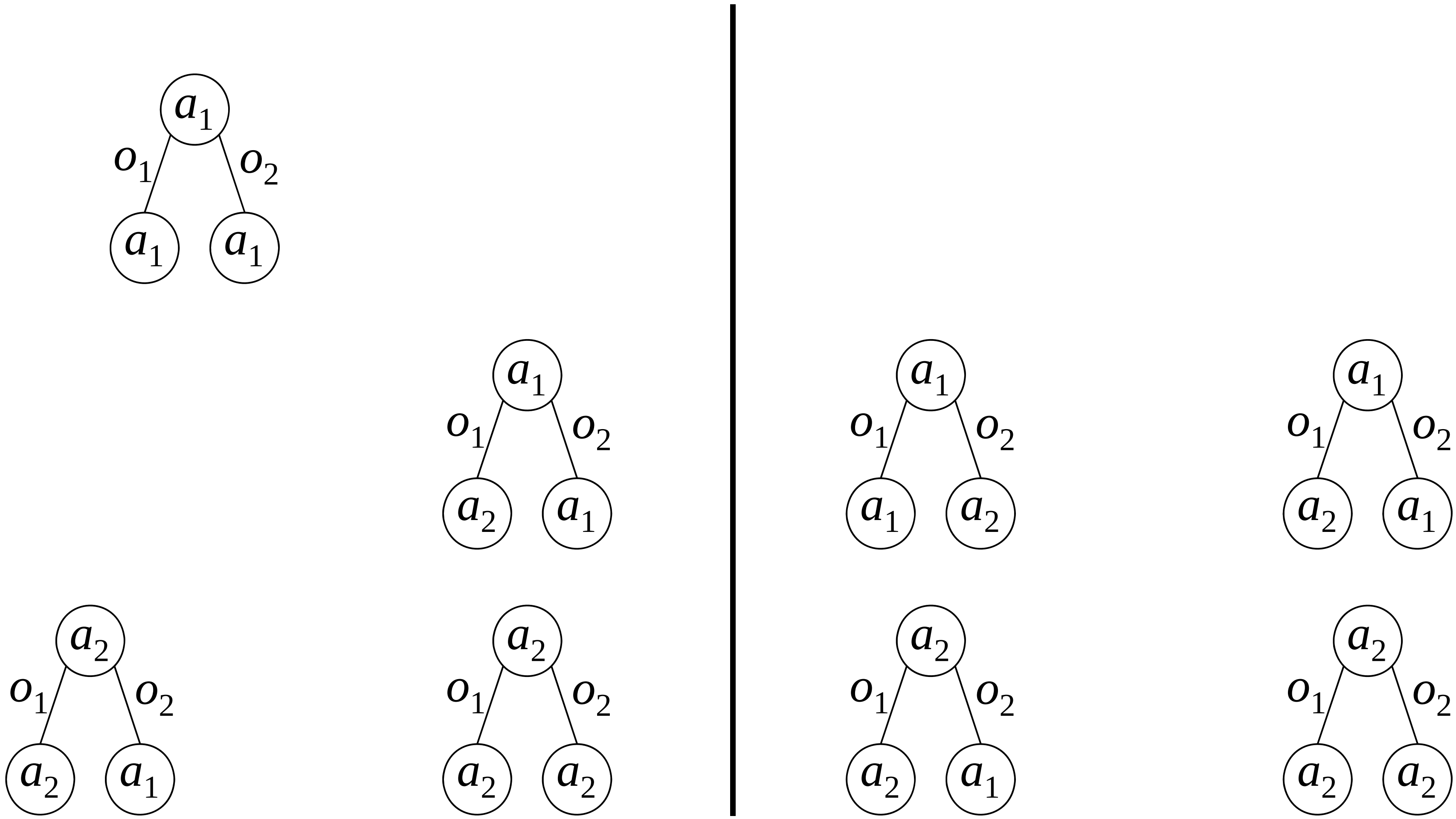
DP in DEC-POMDP

Eliminate dominated policies – iteration 3



DP in DEC-POMDP

Eliminate dominated policies – iteration 4



DP in DEC-POMDP

Interesting property:

DP may be used to find the optimal joint policy in DEC-POMDP

This is because in the cooperative case, removing (weakly) dominated policies preserves at least one optimal joint policy. If more than one policy remains, simply select the joint policy that is largest in value

Improving DP in DEC-POMDP

An exponential number of policy trees are generated during each backup stage for each agent. Many of these policies are dominated

Can we reduce the number of trees that are generated?

Point based DP in DEC-POMDP

Point based DP

- Select a set of multiagent belief points
- Prune and backup policies that are optimal at these points
- Expand the set of multiagent belief points

We prune the policy set but at the expense of optimality. Larger number of belief points $\rightarrow$ lesser loss in optimality

Point based DP in DEC-POMDP

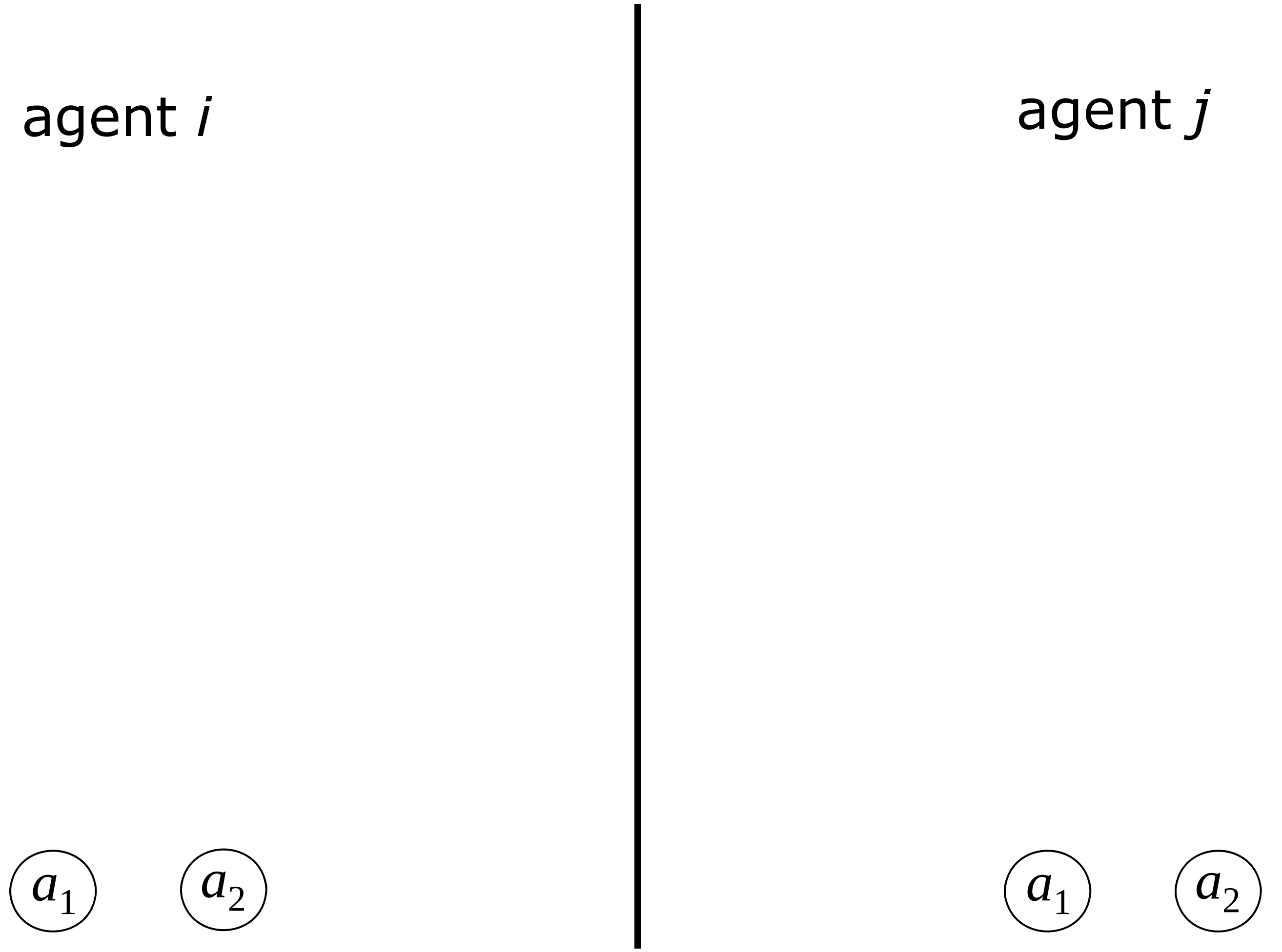
Some techniques for expanding belief points

- Random generation – Generate more belief points randomly
- Stochastic trajectory – Sample other's action, next states, observations and update belief

Belief expansion represents overhead that should be managed cautiously. Trade off optimality for efficiency

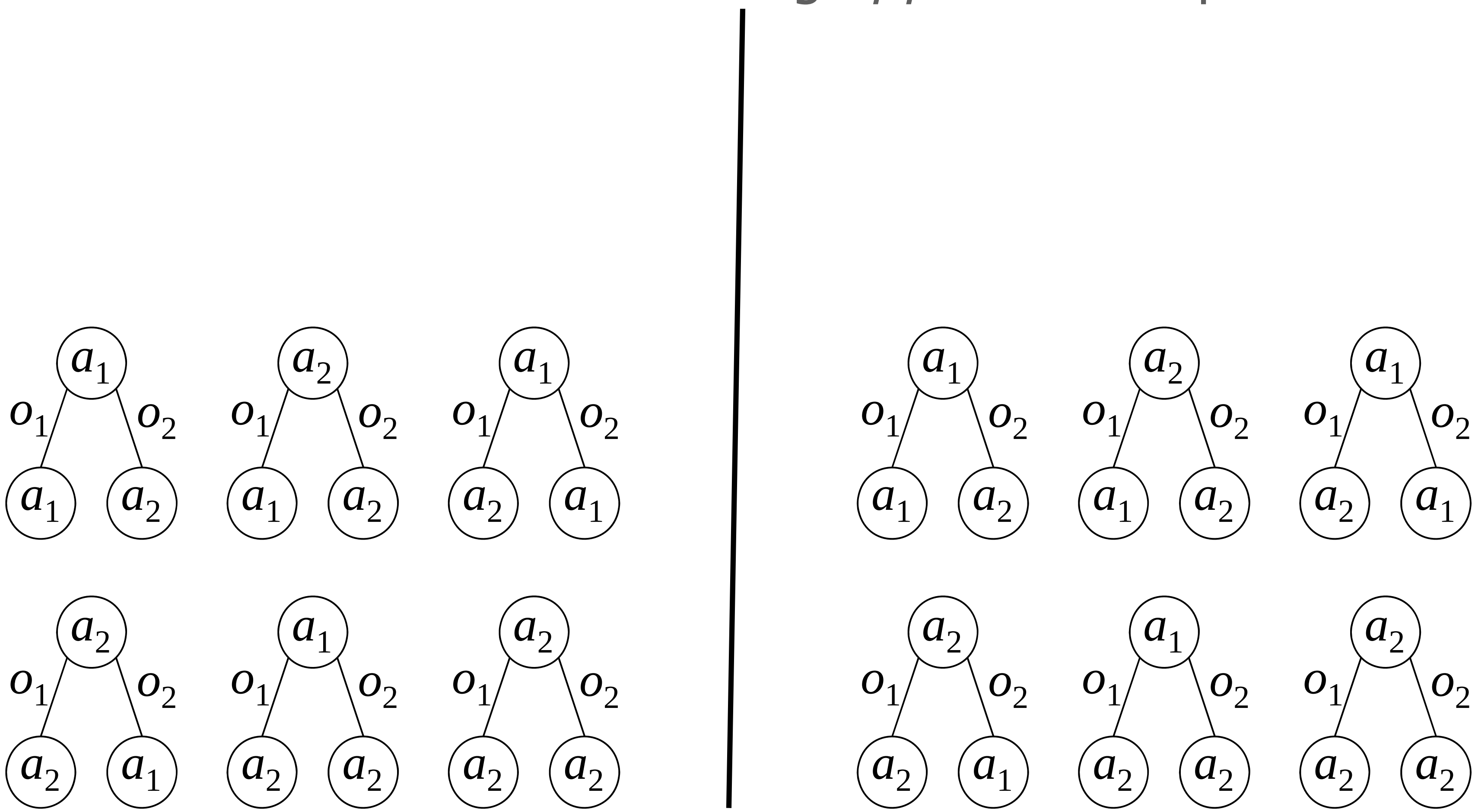
Memory bounded DP (MBDP)

Start with horizon 1 policies



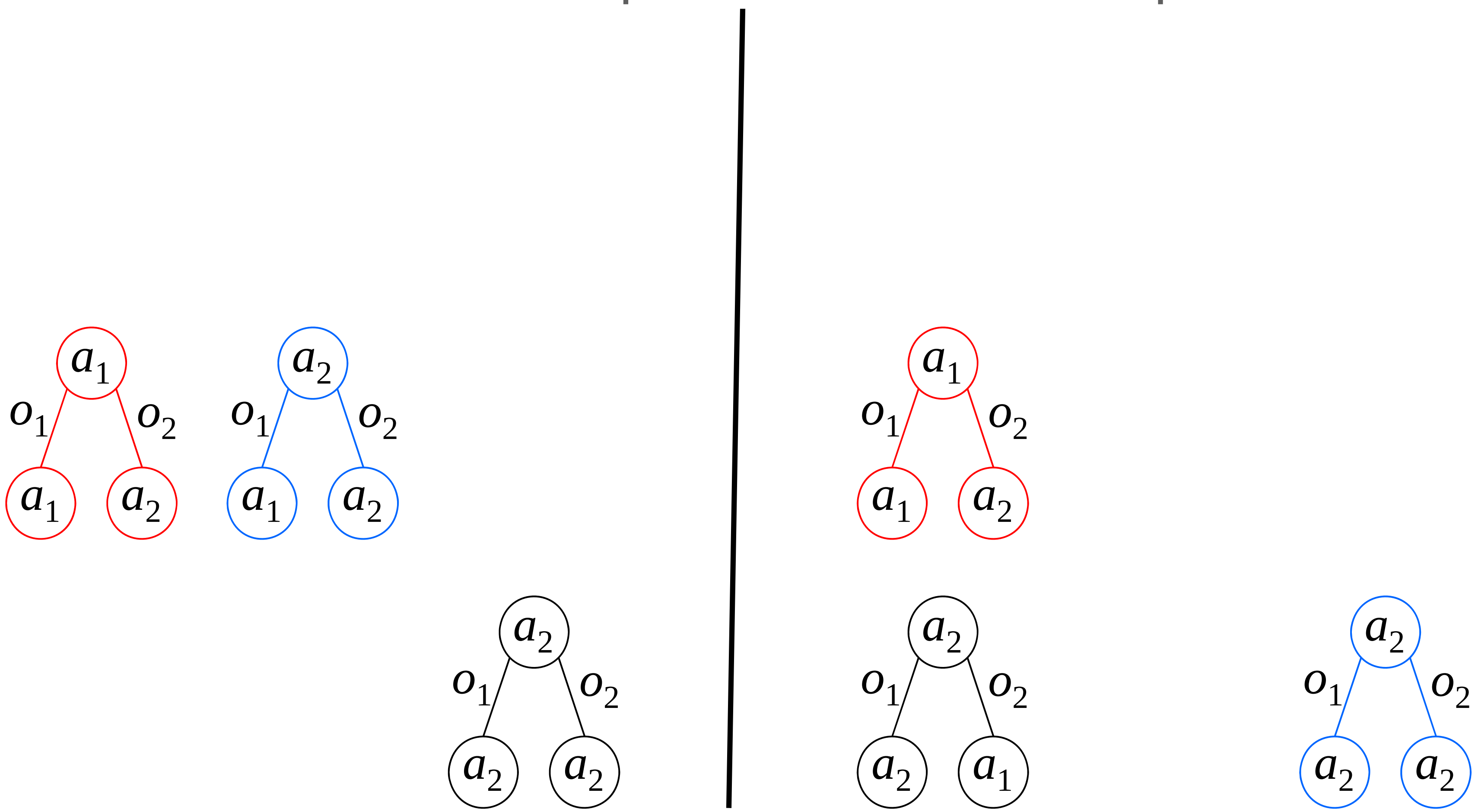
MBDP

Perform an exhaustive backup and select *maxTrees*
Generate belief states using *approximate* policies



MBDP

Select pairs with largest value at the belief states
Use these pairs for next backup



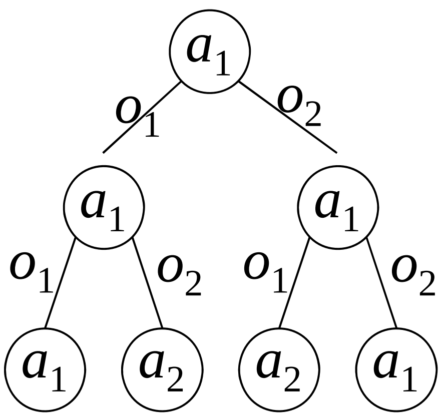
Joint Equilibrium Search for Policies

Search for joint policy such that the individual policies of agents are in equilibrium

- Policy computation is centralized but execution of policies is distributed
- Centralized planning addresses the problem of multiple equilibria

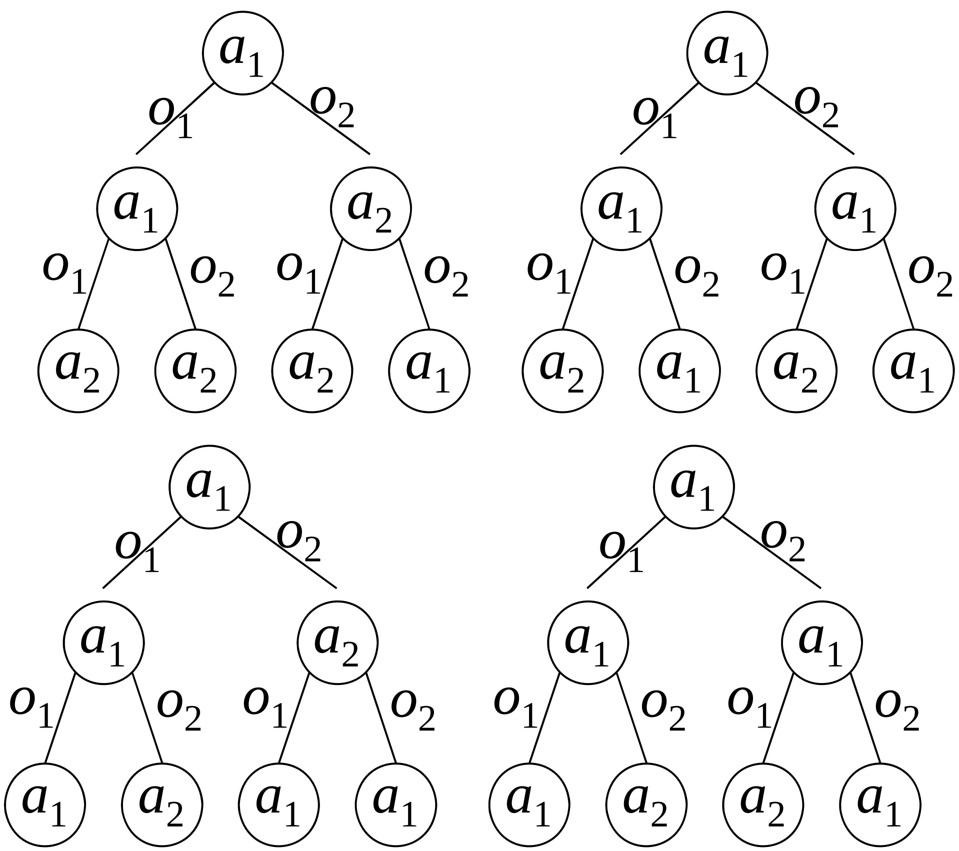
JESP

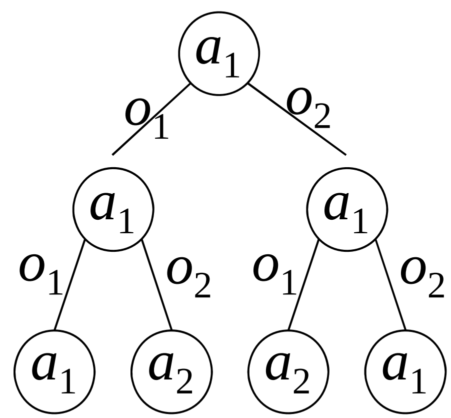
Fix other agent's policy



JESP

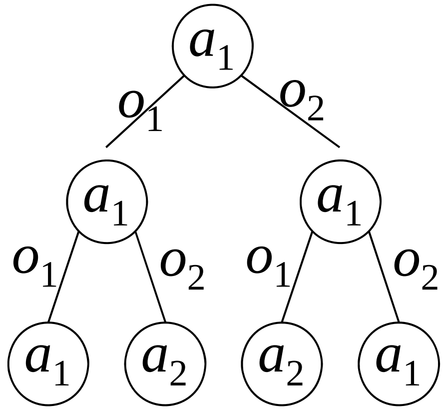
Generate all possible policies for agent *i*

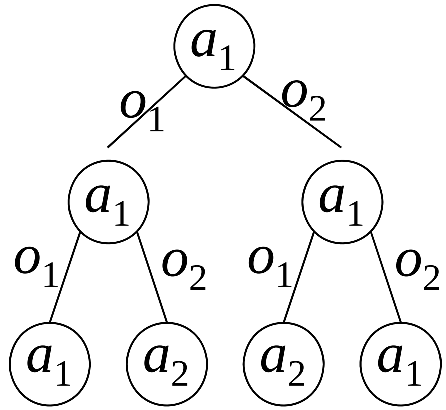




JESP

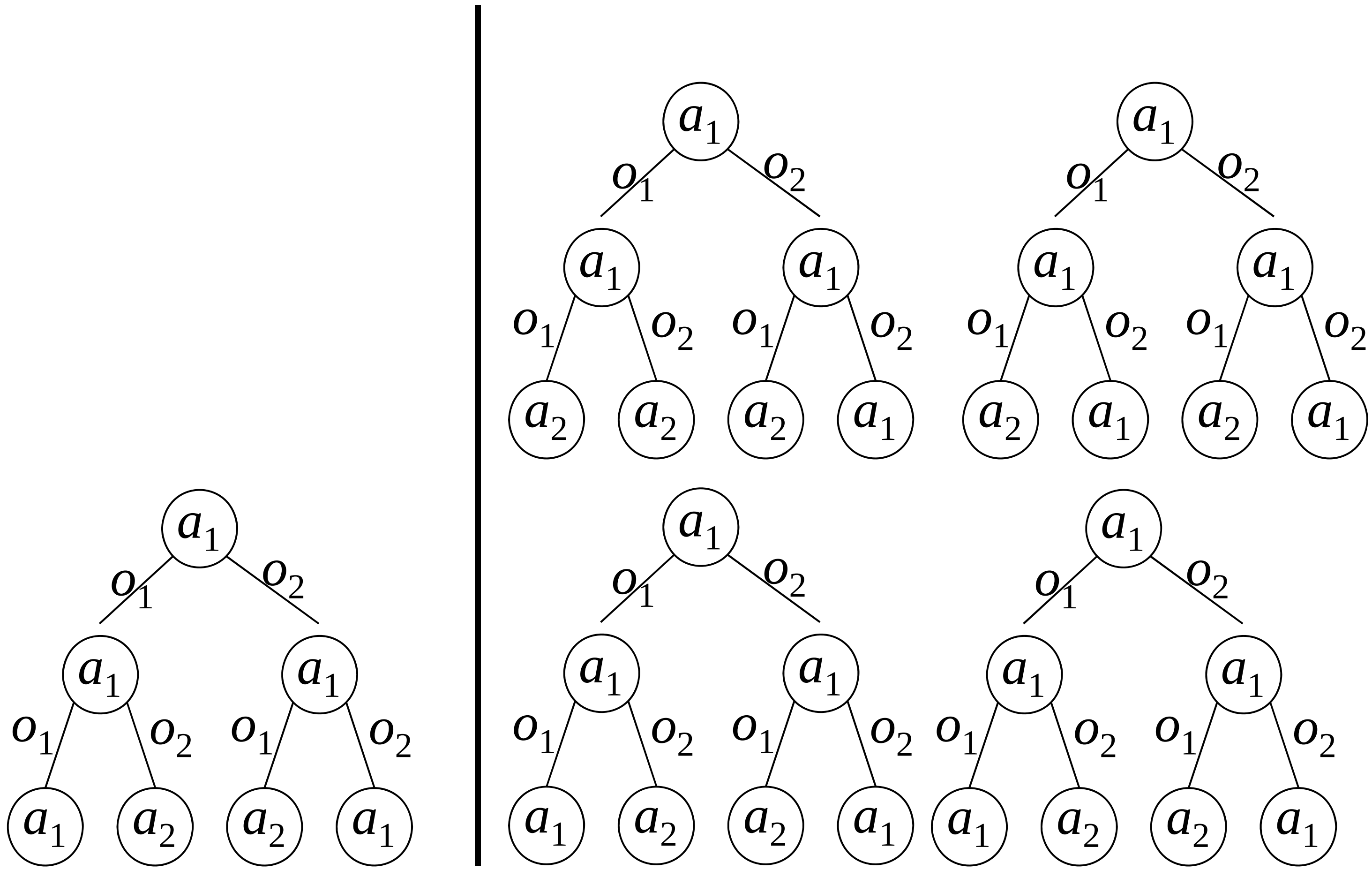
Select best response policy for *i*





JESP

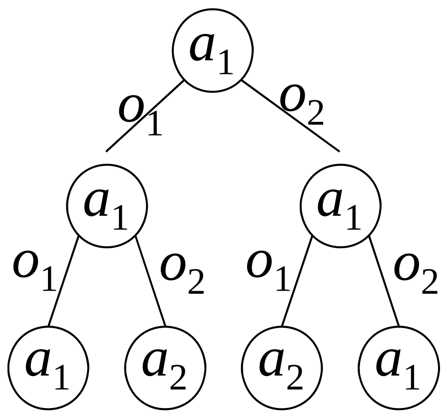
List all policies for agent *j*

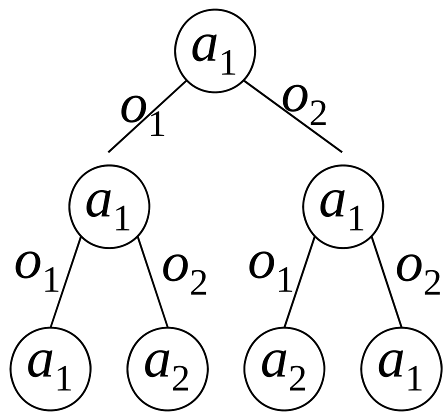


JESP

Select best response policy for *j* and iterate

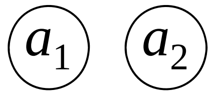
Policies are in equilibrium and represent a local optimum

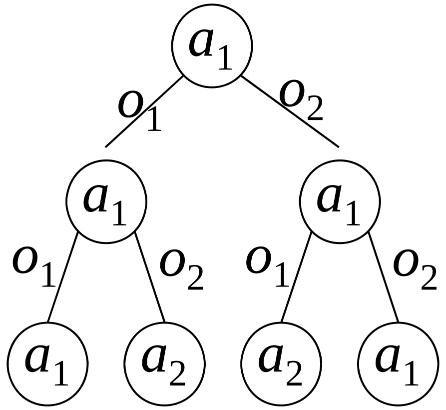




DP-JESP

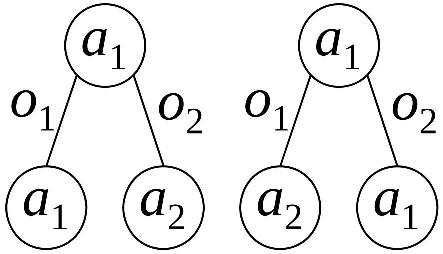
Instead of listing all policies,
build a best response policy using DP

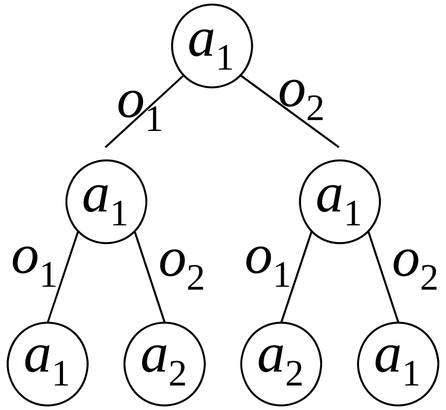




DP-JESP

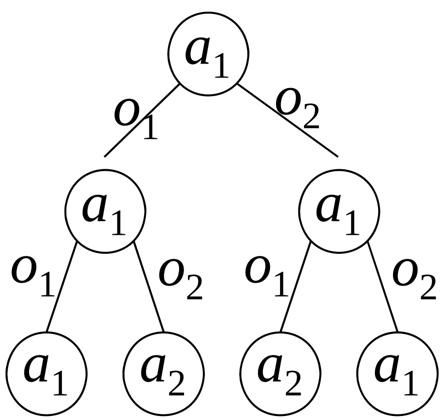
Generate a set of reachable belief points and perform DP over them

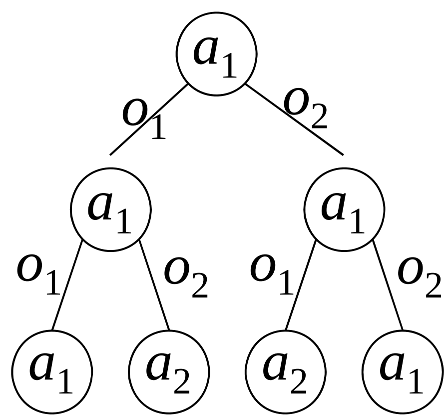




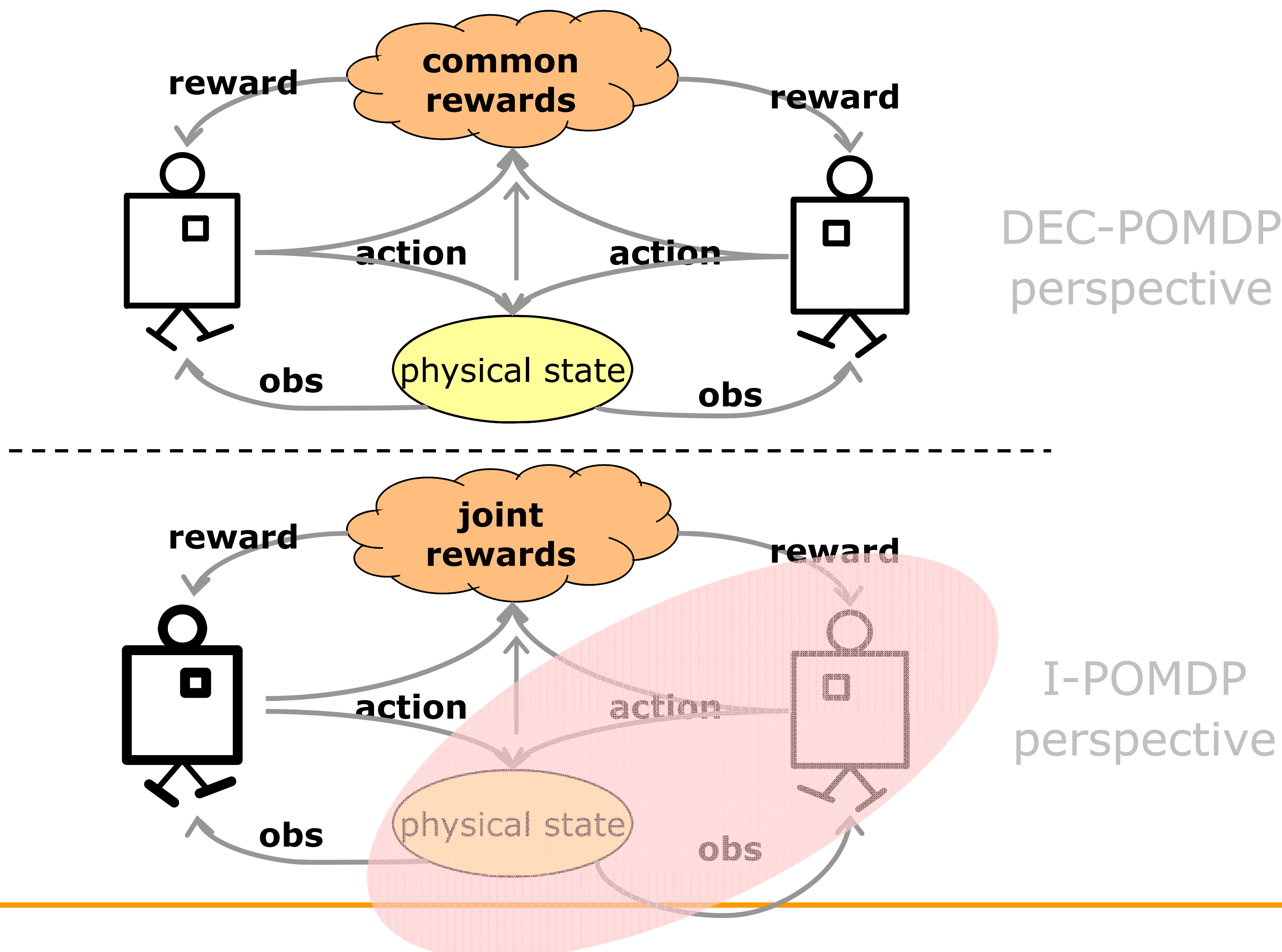
DP-JESP

Generate a set of reachable belief points and perform DP over them

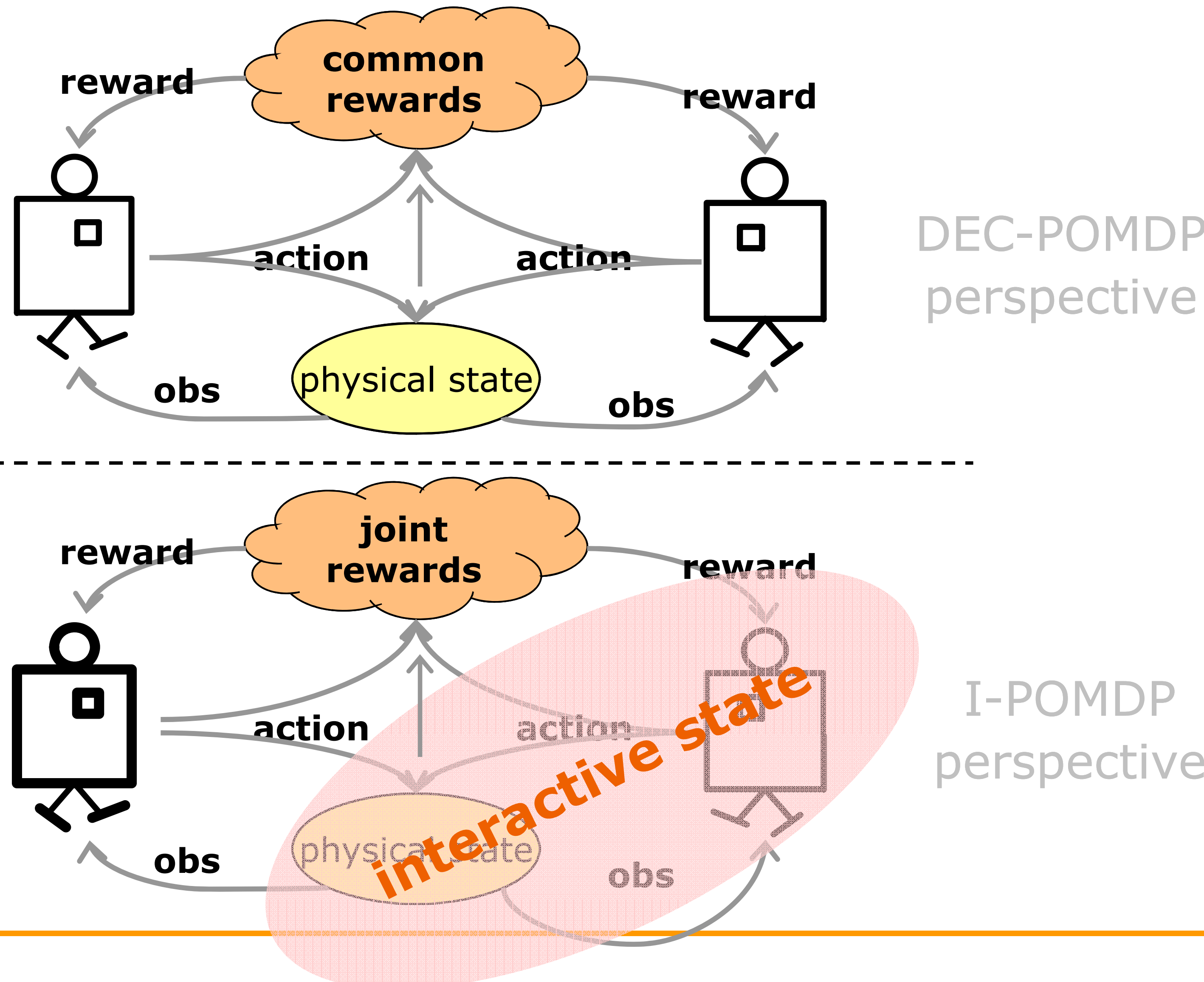




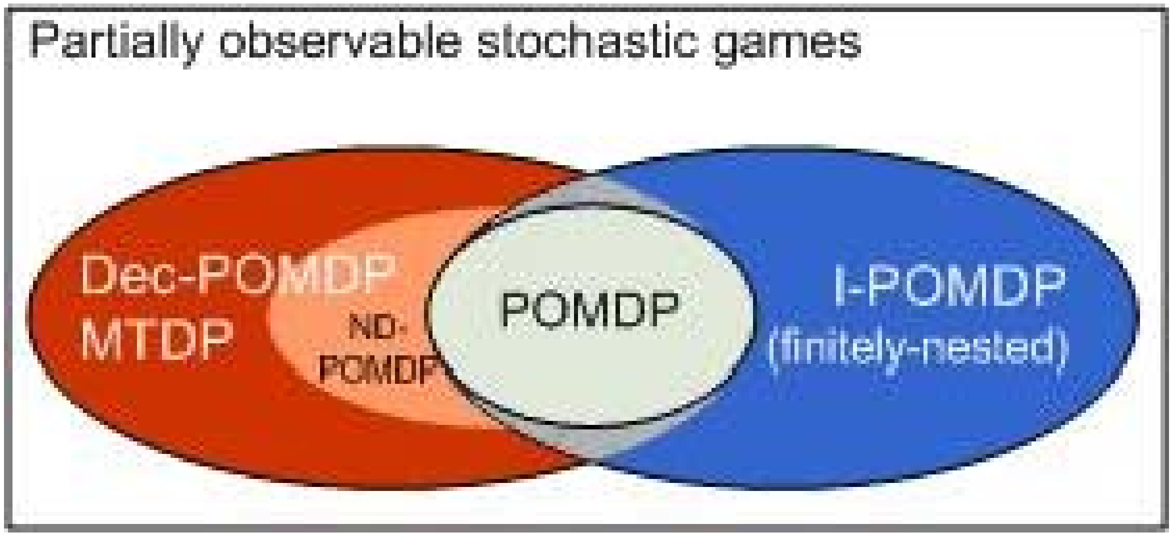
DEC-POMDP and I-POMDP



DEC-POMDP and I-POMDP



Relationship between models



Gray area in the intersecting region of Dec- and I-POMDP indicates the current uncertainty about whether team behavior as produced by a Dec-POMDP could be obtained from a finitely nested I-POMDP for certain problems, as well

This region may not be empty and is a topic of ongoing investigations

Roadmap

- Driving applications
  - search and rescue
- Multiagent decision making
  - description, requirements, complexity
- Game theory
  - classroom game
  - repeated strategic and Bayesian games
  - fictitious play and regret matching
- Stochastic games
  - DEC-MDP and its specializations
- Partially observable stochastic games
  - I-POMDP framework
  - I-POMDP solution techniques
  - I-POMDP applications
  - Dec-POMDP
- Uncertainty Utilization
  - TTD-MDP, Multiagent EMT

Human-Agent Collaboration

- Possible to create a training tool for human emergency response teams.
  - E.g. firefighter managers have been trained using RoboCup Rescue.
- Emergency protocols allow a stochastic model of humans interacting with a simulated environment.
  - Can it be used to devise a flexible training environment?
  - How can we diversify the experience to provide a sufficient span of scenarios?
  - Can a certain degree of surprise be ensured?

Interactive simulations

- Interaction is a sequence of complex events which are
  - extended in time
  - have a component hidden from the human player
- Surprise can be achieved by
  - Exposition of information contrary to the known
    - Find that the building is not abandoned
  - Sequencing of events that require polar response
    - False report of a fire in the North followed by a report that it is in the South

Interactive simulations

- Interaction is a sequence of complex events which are
  - extended in time
  - have a component hidden from the human player
- Surprise can be achieved by
  - Exposition of information contrary to the known
    - Find that the building is not abandoned
  - Sequencing of events that require polar response
    - False report of a fire in the North followed by a report that it is in the South
- How do we produce these sequences?

Interactive simulations

- Interaction is a sequence of complex events which are
  - extended in time
  - have a component hidden from the human player
- Surprise can be achieved by
  - Exposition of information contrary to the known
    - Find that the building is not abandoned
  - Sequencing of events that require polar response
    - False report of a fire in the North followed by a report that it is in the South
- How do we produce different sequences?

Interactive simulations

- Interaction is a sequence of complex events which are
  - extended in time
  - have a component hidden from the human player
- Surprise can be achieved by
  - Exposition of information contrary to the known
    - Find that the building is not abandoned
  - Sequencing of events that require polar response
    - False report of a fire in the North followed by a report that it is in the South
- How do we produce different sequences?
  - Interactive simulations $\equiv$ dynamic narratives

Trajectory Distribution: Intuition

- Markovian environment representation $< S, A, T >$
  - States are plot points experienced by a player
  - Actions are effects external to the player
  - State transitions are plot connections

Trajectory Distribution: Intuition

- Markovian environment representation $< S, A, T >$
  - States are plot points experienced by a player
    - A firefighter discovers a new fire hazard
    - Police finds a new witness
  - Actions are effects external to the player
  - State transitions are plot connections

Trajectory Distribution: Intuition

- Markovian environment representation $< S, A, T >$
  - States are plot points experienced by a player
  - Actions are effects external to the player
  - State transitions are plot connections

Trajectory Distribution: Intuition

- Markovian environment representation $< S, A, T >$
  - States are plot points experienced by a player
  - Actions are effects external to the player
    - A witness approaches the firefighter
    - A bank robbery occurs
  - State transitions are plot connections

Trajectory Distribution: Intuition

- Markovian environment representation $< S, A, T >$
  - States are plot points experienced by a player
  - Actions are effects external to the player
  - State transitions are plot connections

Trajectory Distribution: Intuition

- Markovian environment representation $< S, A, T >$
  - States are plot points experienced by a player
  - Actions are effects external to the player
  - State transitions are plot connections
    - Subject to the player's behaviour (stochasticity)
    - Subject to the narrator's decisions (actions)

Trajectory Distribution: Intuition

- Markovian environment representation $< S, A, T >$
  - States are plot points experienced by a player
  - Actions are effects external to the player
  - State transitions are plot connections

Trajectory Distribution: Intuition

- Markovian environment representation $< S, A, T >$
  - States are plot points experienced by a player
  - Actions are effects external to the player
  - State transitions are plot connections
- A story is a trajectory over plot points (states)
- Trajectory distribution means that a different story is told every time

Example – Fire Chief game

- A Fire Chief manages 3 firefighter teams
- Consider three stories:
  - Story 1
  - “Yesterday a firefighter Team A has been withdrawn from the Toy Factory fire and sent to the Docks. As your correspondent has later discovered, the Docks housed dangerous materials, which led to the infamous explosion and the subsequent perish of Team A.”

Example – Fire Chief game

- A Fire Chief manages 3 firefighter teams
- Consider three stories:
  - Story 2
  - “Earlier today, following an anonymous tip, the Fire Chief sent both Team A and Team B to the Docks, leaving only Team C to handle the fire in our beloved Toy Factory. However, this controversial decision proved to be prudent, since it has prevented the explosion of dangerous chemicals in the Docks.”

Example – Fire Chief game

- A Fire Chief manages 3 firefighter teams
- Consider three stories:
  - Story 3
  - “Our ancient Toy Factory sustained yesterday irrecoverable damage due to the fire that spread from its storage rooms. All three of our firefighter teams where at the time deployed at the Docks, where a minor chemicals leak was handled by one of them. As a result, by the time they arrived at the Toy Factory the place was engulfed in flames.”

Example – State Space

- Consider the ratios of firefighter teams present to the necessary number of teams: $(x : x^*, y : y^*)$

Example – State Space

- Consider the ratios of firefighter teams present to the necessary number of teams: $(x : x^*, y : y^*)$
- A story is then a trajectory through this state space

Example – State Space

- Consider the ratios of firefighter teams present to the necessary number of teams: $(x : x^*, y : y^*)$
- A story is then a trajectory through this state space
  - Story 1
  - $(3 : 1, 0 : 1)$ – All teams are at the Toy Factory
  - $(2 : 0, 1 : 2)$ – Team A is recalled to the Docks
  - $(2 : 0, 0 : 2)$ – Explosion kills Team A

Example – State Space

- Consider the ratios of firefighter teams present to the necessary number of teams: $(x : x^*, y : y^*)$
- A story is then a trajectory through this state space
  - Story 2
  - $(3 : 2, 0 : 1)$ – All teams are at the Toy Factory
  - $(1 : 1, 2 : 2)$ – Team A and B are set to the Docks
  - $(1 : 0, 2 : 0)$ – Explosion is prevented at the docks

Example – State Space

- Consider the ratios of firefighter teams present to the necessary number of teams: $(x : x^*, y : y^*)$
- A story is then a trajectory through this state space
  - Story 3
  - $(0 : 1, 3 : 1)$ – All teams are at the Docks
  - $(0 : 3, 3 : 0)$ – Docks are safe, Toy Factory ablaze
  - $(3 : 0, 0 : 0)$ – Too late: Toy Factory burned down

Example – Actions and Transitions

- States are the ratios of firefighter teams present to the necessary number of teams: $(x : x^*, y : y^*)$
  - A story is a trajectory through this state space
- Actions are hints and information given to the player
  - Anonymous call about chemicals at the Docks
  - TV coverage of the Toy Factory fire
  - An explosion at the Docks

Example – Actions and Transitions

- States are the ratios of firefighter teams present to the necessary number of teams: $(x : x^*, y : y^*)$
  - A story is a trajectory through this state space
- Actions are hints and information given to the player
  - Anonymous call about chemicals at the Docks
  - TV coverage of the Toy Factory fire
  - An explosion at the Docks
- How do we choose actions to produce Story 1?
  - How do we choose actions so that Story 3 is more likely?

Target Trajectory Distribution MDP

- Given an Markovian environment: $< S, T, A >$ where
  - S is the set of states of the world,
  - A is the set of actions,
  - $T : S \times A \rightarrow \Delta(S)$ is the transition function with $T(s'|s, a)$ being the probability of the world changing from state s to state s' if the action a was applied.
- Can we prefer a specific long term sequence?
  - Can the preference be soft, i.e. a distribution?

TTD-MDP (cont)

- Let $\tau \subset S^+$ be a set of finite sequences of states.
  - We will assume that τ is formed by paths in a tree.
- Let $\mathcal{P}(\cdot)$ be a distribution over τ.
  - $\mathcal{P}$ represents our preferences over various, long-term system developments
- A TTD-MDP is defined by a tuple $<<S, T, A>, \tau, \mathcal{P}>$
  - Notice that a transition function $\mathcal{T} : \tau \times A \rightarrow \Delta(\tau)$ is naturally induced by T.

TTD-MDP: Questions

- Given a TTD-MDP, $<<S, T, A>, \tau, \mathcal{P}>$
- What is the policy $\pi : \tau \rightarrow A$ that induces $\mathcal{P}$?
  - Is it always possible to produce $\mathcal{P}$?
    - No, transition function T may prevent that.

TTD-MDP: Questions

- Given a TTD-MDP, $<<S, T, A>, \tau, \mathcal{P}>$
- What is the policy $\pi : \tau \longrightarrow A$ that induces $\mathcal{P}$?
  - Is it always possible to produce $\mathcal{P}$?
    - No, transition function T may prevent that.
  - How do we measure performance?
    - Information Theory provides a divergence measure between two distributions: Kullback-Leibler divergence

TTD-MDP: Questions

- Given a TTD-MDP, $<< S, T, A >, \tau, \mathcal{P} >$
- What is the policy $\pi : \tau \rightarrow A$ that induces $\mathcal{P}$?
  - Is it always possible to produce $\mathcal{P}$?
    - No, transition function T may prevent that.
  - How do we measure performance?
    - Information Theory provides a divergence measure between two distributions: Kullback-Leibler divergence
  - Can the policy be computed on-line?
    - Yes, the structure of τ combined with appropriate performance measure allow that.

TTD-MDP: Further Questions

- Assumes complete observability
  - Active plot point is always known to the narrator
  - Will not hold if the narrator is part of the simulation
- Trajectories are finite
  - What if it's a never-ending story?
  - Can a TTD-like principle be defined for infinite trajectories?
- Single agent
  - What if the simulation includes multiple "narrators"?
  - Can a similar TTD principle be applied for multi-agent simulations?

Example – Story

- Two police precincts are fighting organised crime
  - They are unable to catch the leader
    - There are signs of him being in the precinct, but not the exact location
  - They know that increased patrols make him uncomfortable
  - If the leader moves from precinct to precinct, his crime activity is disrupted
- Ideally the police would like to modulate patrols so as to keep the crime leader in constant agitation

Partially observable environment

- A partially observable Markovian environment $< S, s_0, A, T, \Omega, O >$
  - S state space of the world, s_0 is the initial state
  - A is a set of actions available to the agent
  - $T : S \times A \times S \to [0, 1]$ is the transition function
  - Ω is the set of all possible observations
  - $O : \Omega \times S \times A \times S \to [0, 1]$ is the observability function.
    - $O(o|s', a, s)$ is the probability that the agent will observe o if it performed a and the world shifted from s to s'.

Markovian, but not (PO)MDP

- Given a Markovian environment $< S, A, T, O, \Omega >$
- To describe a task within the environment:
  - Expression of preferences
    - Need to encode infinite system development
    - Include multiple developments
    - Include randomisation
  - Reference system dynamics $\tau : S \times S \rightarrow [0, 1]$
    - Stochastic rule
    - Chains infinite sequences
    - Native to the environment model
  - Performance measure
    - Kullback-Leibler divergence
    - Need to (quickly) converge to the reference

Example

- Two police precincts are fighting organised crime
  - They are unable to catch the leader
    - There are signs of him being in the precinct, but not the exact location
  - They know that increased patrols make him uncomfortable
  - If the leader moves from precinct to precinct, his crime activity is disrupted
- Ideally the police would like to modulate patrols so as to keep the crime leader in constant agitation

Example (cont)

- Environment $< S, \bigotimes A_i, T, \bigotimes \Omega_i, \{O_i\} >$
  - $S = \{pr_1, pr_2\}$ is the set of precincts
  - $A_i = \{higher, lower\}$ is increasing or decreasing patrols
  - $\Omega_i = S$ is an indicator of leader's presence in the precinct
  - T reflects leader's tendency to move
  - O_i reflects the police capability to gather information
- Reference dynamics is then $\tau(s', s) = \begin{cases} 1 & s \neq s' \\ 0 & otherwise \end{cases}$

Questions

- The environment is only partially observable
  - How can we even know what kind of state sequence is being reproduced?
  - Can we know what kind of system dynamics represents that sequence?
- Given a reference signal represented by system dynamics τ
  - How can we construct the policy that produces the reference?

Recording the world

- How do we know where we are?
  - We can summarise all our knowledge in a single distribution $p_t : S \longrightarrow [0,1]$
  - $p_t(s)$ expresses the degree (probability) to which we believe that the state at time t is s.
- How do we compute it?
  - $p_0(s) = 1$ iff $s = s_0$
  - Given that an agent performed action a and received observation o:

$$p_{t+1}(s) \propto O(o|s,a) \sum_{s'} \mathcal{T}(s|a,s')p_t(s')$$

Explaining the world: EMT

- How do we know how the world moves?
  - We can estimate the instantiated dynamics:
    $$\tau : S \times S \rightarrow [0, 1]$$
    - For τ has to hold $p_{t+1} = p_t * \tau$
  - There are many such τ
    - Make a conservative update:
    $$\tau_{t+1} = \arg \min_{\tau : p_{t+1} = p_t * \tau} d(\tau, \tau_t)$$
  - If $d(\cdot, \cdot)$ is Kullback-Leibler divergence the update is termed Extended Markov Tracking (EMT)
    - EMT's update is shorthanded $H[p_{t+1} \leftarrow p_t, \tau_t]$

EMT Control

- It is possible to utilise EMT to construct an on-line policy to reproduce a reference dynamics τ^*
- Control loop is composed by
  - Belief update
  - EMT estimation of system development
  - Let $T_a = T(\cdot|a,\cdot)$. Action choice

$$a^* = \arg\min_a D_{KL}(\ H[p_t * T_a \leftarrow p_t, \tau_t] \ \|\ \ \tau^*)$$

  - Application of a^*.
- But can it be used in a multi-agent setting?

Stigmergy

- Stigmergy is a mechanism of spontaneous, indirect coordination
  - Trace left in the environment by an action stimulates the performance of a subsequent action, by the same or a different agent.
- Assume that two agents choose actions a_1,a_2 and the joint operation (a_1, a_2) is applied on a common system state.
  - In a stigmergic environment observations will provide information on the state dynamics and enable action coordination

Multi-agent EMT

- Given an environment: $< S, \bigotimes A_i, T, \bigotimes O_i, \{\Omega_i\} >$, and a reference dynamics τ^*
- Let each agent run independent EMT based control on the complete actions space $\bigotimes A_i$ as follows:
  - Update beliefs p_t according to T and O_i
  - Compute EMT estimate of system development
  - Compute optimal *joint action*

$$a^* = (a_1^*, ..., a_N^*) = \arg\min_a D_{KL}(\ H[p_t{*}T_a \leftarrow p_t, \tau_t] \,\|\ \tau^*)$$

  - Apply a_i^*

Stigmergy – example

- Each police precinct will
  - Estimate the apparent crime leader behaviour
  - Predict the effect of a coordinated patrols.
  - Apply the local portion of the joint action

Stigmergy – example

- Each police precinct will
  - Estimate the apparent crime leader behaviour
    - Using crime leader model and EMT
  - Predict the effect of a coordinated patrols.
  - Apply the local portion of the joint action

Stigmergy – example

- Each police precinct will
  - Estimate the apparent crime leader behaviour
  - Predict the effect of a coordinated patrols.
  - Apply the local portion of the joint action

Stigmergy – example

- Each police precinct will
  - Estimate the apparent crime leader behaviour
  - Predict the effect of a coordinated patrols.
    - These joint actions are not necessarily the same
  - Apply the local portion of the joint action

Stigmergy – example

- Each police precinct will
  - Estimate the apparent crime leader behaviour
  - Predict the effect of a coordinated patrols.
  - Apply the local portion of the joint action

Stigmergy – example

- Each police precinct will
  - Estimate the apparent crime leader behaviour
  - Predict the effect of a coordinated patrols.
  - Apply the local portion of the joint action
    - Combined into a joint action different from all player choices

Stigmergy – example

- Each police precinct will
  - Estimate the apparent crime leader behaviour
  - Predict the effect of a coordinated patrols.
  - Apply the local portion of the joint action
- Crime leader responds to the combined joint action leading to stigmergy

Stigmergy – example

- Each police precinct will
  - Estimate the apparent crime leader behaviour
  - Predict the effect of a coordinated patrols.
  - Apply the local portion of the joint action
- Crime leader responds to the combined joint action leading to stigmergy
  - Observations provide a correlation signal
  - Dynamics estimates are correlated
  - Locally computed joint actions will not differ

Stigmergy – example

- Each police precinct will
  - Estimate the apparent crime leader behaviour
  - Predict the effect of a coordinated patrols.
  - Apply the local portion of the joint action
- Crime leader responds to the combined joint action leading to stigmergy
  - Observations provide a correlation signal
  - Dynamics estimates are correlated
  - Locally computed joint actions will not differ
    - too much too frequently
    - in their effect on the dynamics estimate

Stochasticity is Bad

- System is continually changing

Stochasticity is Bad

- System is continually changing
  - No single state trajectory is certain

Stochasticity is Bad

- System is continually changing
  - No single state trajectory is certain
- In partially observable systems

Stochasticity is Bad

- System is continually changing
  - No single state trajectory is certain
- In partially observable systems
  - Can not track a single state trajectory

Stochasticity is Bad

- System is continually changing
  - No single state trajectory is certain
- In partially observable systems
  - Can not track a single state trajectory
  - Concept of system dynamics is needed

Stochasticity is Bad

- System is continually changing
  - No single state trajectory is certain
- In partially observable systems
  - Can not track a single state trajectory
  - Concept of system dynamics is needed
  - Only apparent dynamics can be used

Stochasticity is Good

- System is continually changing
  - No single state trajectory is certain
- In partially observable systems
  - Can not track a single state trajectory
  - Concept of system dynamics is needed
  - Only apparent dynamics can be used

Roadmap

- Driving applications
  - search and rescue
- Multiagent decision making
  - description, requirements, complexity
- Game theory
  - classroom game
  - repeated strategic and Bayesian games
  - fictitious play and regret matching
- Stochastic games
  - DEC-MDP and its specializations
- Partially observable stochastic games
  - I-POMDP framework
  - I-POMDP solution techniques
  - I-POMDP applications
  - Dec-POMDP
- Uncertainty Utilization
  - TTD-MDP, Multiagent EMT

Bibliography

- Game theory
  1. Fundenberg, D. and Tirole, J., *Game theory*. MIT Press (textbook)
  2. Owen, *G., Game theory*. 3rd Edition, Academic Press (textbook)
  3. Binmore, K., *Essays on foundations of game theory.* Pittman, (edited book)
  4. Harsanyi, J. C. (1967). Games with incomplete information played by 'Bayesian' players. *Management. Science*, 14(3), 159-182 (reference on Bayesian games)
  5. Fudenberg, D., & Levine, D. (1997). *Theory of Learning in Games*. MIT Press (book for fictitious play)
  6. Aumann, R. J. (1999). Interactive epistemology i: Knowledge. *International Journal of Game Theory*, *28*, 263-300
  7. Brandenburger, A., & Dekel, E. (1993). Hierarchies of beliefs and common knowledge. *Journal of Economic Theory*, *59*, 189-198 (ref. on hierarchical belief systems)

- Regret matching
  1. Hart, S. & Mas-Colell A. (2000). A simple adaptive procedure leading to correlated equilibrium. *Econometrica*, 68(5): 1127-1150
  2. Hart, S. (2005). Adaptive heuristics. *Econometrica*, 73(5): 1401-1430

Bibliography

- Interactive POMDP

1. Gmytrasiewicz, P. J., & Doshi, P. (2005). A framework for sequential planning in multi-agent settings. *Journal of Artificial Intelligence Research*, *24:49-79* (ref. on I-POMDP)
2. Doshi, P., & Gmytrasiewicz, P.J. (2006). On the Difficulty of Achieving Equilibrium in Interactive POMDPs. *Twenty First National Conference on Artificial Intelligence (AAAI)* (ref. on convergence properties of I-POMDP)
3. Doshi, P. (2007). Improved State Estimation in Multiagent Settings with Continuous or Large Discrete State Spaces. *Twenty Second Conference on Artificial Intelligence (AAAI)* (ref. on state estimation for continuous state spaces)
4. Doshi, P., & Gmytrasiewicz, P. J. (2009). Monte Carlo Sampling Methods for Approximating Interactive POMDPs. *Journal of Artificial Intelligence Research*, *34:297-337* (ref. on PF in I-POMDP)
5. Perez, D., & Doshi, P. (2008). Generalized Point Based Value Iteration for Interactive POMDPs. In Twenty-third Conference on Artificial Intelligence (AAAI) (ref. on PBVI in I-POMDP)
6. Sonu, E. and Doshi, P. (2012). Generalized and Bounded Policy Iteration for Interactive POMDPs. *Eleventh International Autonomous Agents and Multiagent Systems Conference (AAMAS)* (ref. on BPI in I-POMDP)
7. Hoang, T. & Low, K. (2012). Intention-Aware Planning under Uncertainty for Interactingwith Self-Interested, Boundedly Rational Agents. *Eleventh International Autonomous Agents and Multiagent Systems Conference (AAMAS)* (ref. on specialization, IA-POMDPs)
8. Ng, B., Boakye, K., Meyers, C., & Wang, A. (2012). Bayes-Adaptive Interactive POMDPs. *Twenty-Sixth AAAI Conference on Artificial Intelligence* (ref. on RL in I-POMDP)

Bibliography

- Applications of Interactive POMDP

1. Seymour, R.S., & Peterson, G.L. (2009). Responding to Sneaky Agents in Multi-agent Domains. *Twenty-Second International Florida Artificial Intelligence Research Society Conference (FLAIRS)*
2. Ng, B., Meyers, C., Boakye, K., & Nitao, J. (2010). Towards applying interactive POMDPs to real-world adversary modeling. *Innovative Applications in Artificial Intelligence (IAAI)*
3. Wunder, M., Kaisers, M., Yaros, J.R., Littman, M. (2011). Using Iterated Reasoning to Predict Opponent Strategies. *Tenth International Conference on Autonomous Agents and Multiagent Systems (AAMAS)*
4. Woodward, M.P., & Wood, R.J. (2012). Learning from Humans as an I-POMDP. *CoRR, http://arxiv.org/abs/1204.0274*
5. Doshi, P., Qu, X., Goodie, A., & Young, D. (2012). Modeling Human Recursive Reasoning using Empirically-Informed Interactive POMDPs. *IEEE Transactions on Systems, Man and Cybernetics (SMC), Part A,* Vol. 42(6):1529-1542

Bibliography

- DEC-MDP
  1. Bernstein, D. S., Givan, R., Immerman, N., & Zilberstein, S. (2002). The complexity of decentralized control of Markov decision processes. *Mathematics of Operations Research*, *27*(4), 819-840 (ref. on complexity of DEC-MDPs)
  2. Becker, R., Zilberstein, S., Lesser, V., & Goldman, C. V. (2003). Transition-independent decentralized markov decision processes. In *Autonomous Agents and Multiagent Systems Conference* (ref. on TI-DEC-MDPs)

- Uncertainty Utilization
  1. Rabinovich, Z. & Rosenschein, J. S. (2005). Multiagent Coordination by Extended Markov Tracking. In *Autonomous Agents and Multiagent Systems Conference* (ref. on EMT)
  2. David L. Roberts, Mark J. Nelson, Charles L. Isbell, Jr., Michael Mateas, and Michael L. Littman. (2006). Targeting Specific Distributions of Trajectories in MDPs. In *Twenty-first Conference on Artificial Intelligence (ref. on TTD-MDPs)*
  3. Sooraj Bhat, David L. Roberts, Mark J. Nelson, Charles L. Isbell, and Michael Mateas. (2007). A Globally Optimal Online Algorithm for TTD-MDPs. In *Autonomous Agents and Multiagent Systems Conference* (ref. on TTD-MDPs)

Bibliography

- DEC-POMDP and specializations

1. Bernstein, D. S., Givan, R., Immerman, N., & Zilberstein, S. (2002). The complexity of decentralized control of Markov decision processes. *Mathematics of Operations Research*, *27*(4), 819-840 (ref. on complexity)
2. Hansen, E., Bernstein, D., & Zilberstein, S. (2004). Dynamic Programming for Partially Observable Stochastic Games. In *Nineteenth National Conference on Artificial Intelligence* (ref. on DP in DEC-POMDPs)
3. Szer, D., & Charpillet, F. (2006). Point-based Dynamic Programming for DEC-POMDPs. In *Twenty-First National Conference on Artificial Intelligence* (ref. on point based DP in DEC-POMDPs)
4. Seuken, S., and Zilberstein, S. (2007) Memory-bounded Dynamic Programming for Decentralized POMDPs. In *International Joint Conference on Artificial Intelligence* (ref. on MBDP in DEC-POMDPs)
5. Nair, R., Tambe, M., Yokoo, M., Pynadath, D., & Marsella, S. (2003). Taming Decentralized Pomdps :Towards Efficient Policy Computation for Multiagent Settings. In *International Joint Conference on Artificial Intelligence* (ref. on MTDP and JESP)
6. Nair R., Varakantham, P., Tambe, M., & Yokoo, M. (2005). Networked Distributed POMDPs: A Synthesis of Distributed Constraint Optimization and POMDPs. In *Twentieth National Conference on Artificial Intelligence* (ref. on ND-POMDPs)

Bibliography

- Survey of multiagent decision making
  1. Seuken, S., & Zilberstein, S. (2008). Formal Models and Algorithms for Decentralized Decision Making Under Uncertainty. *Journal of Autonomous Agents and Multiagent Systems 17(2):190–250.*
  2. Doshi, P. (2012). “Decision Making in Complex Multiagent Contexts: A Tale of Two Frameworks”, *AI Magazine*, Vol. 33(4):82-95

- Evaluation
  1. Sonu, E., & Doshi, P. (2012). GaTAC: A Scalable and Realistic Testbed for Multiagent Decision Making (Demonstration). *Eleventh International Autonomous Agents and Multiagent Systems Conference (AAMAS)* (ref. on an evaluation framework for I- and Dec-POMDPs)