Single agent or multiple agents

- Many domains are characterized by multiple agents rather than a single agent.
- Game theory studies what agents should do in a multi-agent setting. A game is an abstraction of agents interacting.

Single agent or multiple agents

- Many domains are characterized by multiple agents rather than a single agent.
- Game theory studies what agents should do in a multi-agent setting. A game is an abstraction of agents interacting.
- Agents can be cooperative, competitive or somewhere in between.

Single agent or multiple agents

- Many domains are characterized by multiple agents rather than a single agent.
- Game theory studies what agents should do in a multi-agent setting. A game is an abstraction of agents interacting.
- Agents can be cooperative, competitive or somewhere in between.
- Agents that reason and act autonomoulsly can't be modeled as nature.

Multi-agent framework

- Each agent can have its own utility.

Multi-agent framework

- Each agent can have its own utility.
- Agents select actions autonomously.

Multi-agent framework

- Each agent can have its own utility.
- Agents select actions autonomously.
- Agents can have different information.

Multi-agent framework

- Each agent can have its own utility.
- Agents select actions autonomously.
- Agents can have different information.
- The outcome can depend on the actions of all of the agents.

Multi-agent framework

- Each agent can have its own utility.
- Agents select actions autonomously.
- Agents can have different information.
- The outcome can depend on the actions of all of the agents.
- Each agent's value depends on the outcome.

Normal Form of a Game

The strategic form of a game or normal-form game:

- a finite set I of agents, $\{1, \dots, n\}$.
- a set of actions A_i for each agent $i \in I$.

Normal Form of a Game

The strategic form of a game or normal-form game:

- a finite set I of agents, $\{1, \ldots, n\}$.
- a set of actions A_i for each agent $i \in I$.
  An action profile σ is a tuple $\langle a_1, \ldots, a_n \rangle$, means agent i carries out a_i.

Normal Form of a Game

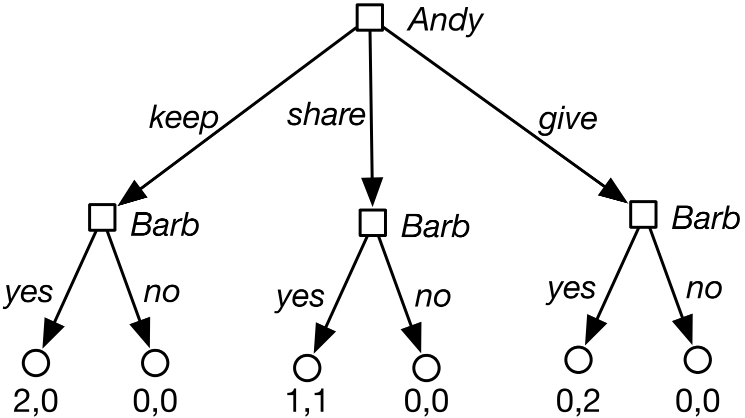
The strategic form of a game or normal-form game:

- a finite set I of agents, $\{1, \ldots, n\}$.
- a set of actions A_i for each agent $i \in I$.
  An action profile σ is a tuple $\langle a_1, \ldots, a_n \rangle$, means agent i carries out a_i.
- a utility function $utility(\sigma, i)$ for action profile σ and agent $i \in I$, gives the expected utility for agent i when all agents follow action profile σ.

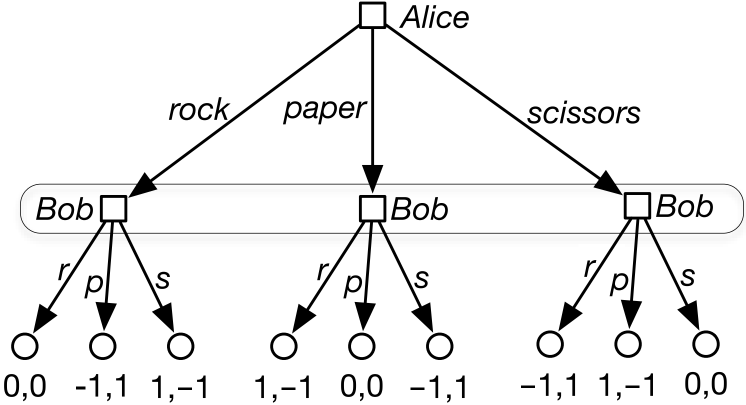
Rock-Paper-Scissors

		Bob		
		rock	*paper*	*scissors*
Alice	*rock*	$0, 0$	$-1, 1$	$1, -1$
	paper	$1, -1$	$0, 0$	$-1, 1$
	scissors	$-1, 1$	$1, -1$	0,0

Extensive Form of a Game

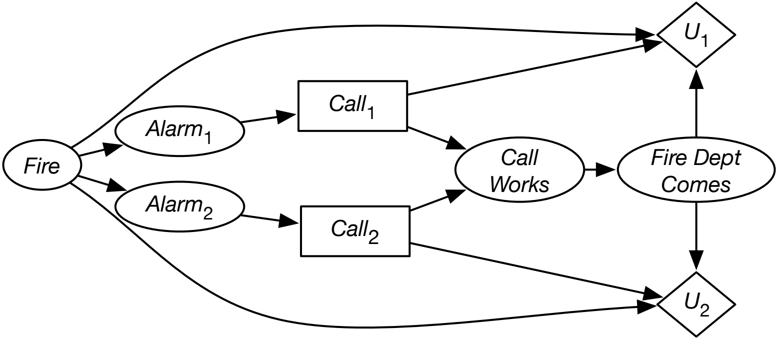


Extensive Form of an imperfect-information Game



Bob cannot distinguish the nodes in an information set.

Multiagent Decision Networks

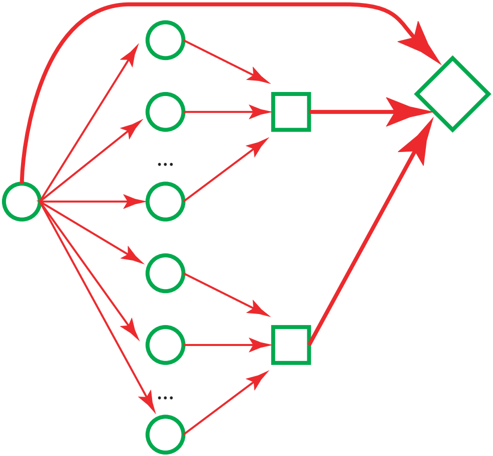


Value node for each agent.

Each decision node is owned by an agent.

The parents of each decision node specify what that agent will observe when making the decision

Multiple Agents, shared value



Complexity of Multi-agent decision theory

- It can be exponentially harder to find optimal multi-agent policy even with a shared values.
- Why?

Complexity of Multi-agent decision theory

- It can be exponentially harder to find optimal multi-agent policy even with a shared values.
- Why? Because dynamic programming doesn't work:
  - If a decision node has n binary parents, dynamic programming lets us solve 2^n decision problems.
  - This is much better than policies (where d is the number of decision alternatives).
- Multiple agents with shared values is equivalent to having a single forgetful agent.

Complexity of Multi-agent decision theory

- It can be exponentially harder to find optimal multi-agent policy even with a shared values.
- Why? Because dynamic programming doesn't work:
  - If a decision node has n binary parents, dynamic programming lets us solve 2^n decision problems.
  - This is much better than d^{2^n} policies (where d is the number of decision alternatives).
- Multiple agents with shared values is equivalent to having a single forgetful agent.

Fully Observable + Multiple Agents

- If agents act sequentially and can observe the state before acting: Perfect Information Games.

Fully Observable + Multiple Agents

- If agents act sequentially and can observe the state before acting: Perfect Information Games.
- Can do dynamic programming or search:
  Each agent maximizes for itself.

Fully Observable + Multiple Agents

- If agents act sequentially and can observe the state before acting: Perfect Information Games.
- Can do dynamic programming or search:
  Each agent maximizes for itself.
- Multi-agent MDPs: value function for each agent.
  each agent maximizes its own value function.

Fully Observable + Multiple Agents

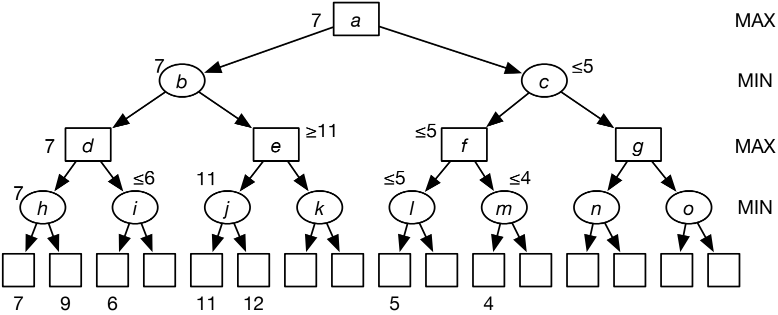
- If agents act sequentially and can observe the state before acting: Perfect Information Games.
- Can do dynamic programming or search:
  Each agent maximizes for itself.
- Multi-agent MDPs: value function for each agent.
  each agent maximizes its own value function.
- Multi-agent reinforcement learning: each agent has its own Q function.

Fully Observable + Multiple Agents

- If agents act sequentially and can observe the state before acting: Perfect Information Games.
- Can do dynamic programming or search:
  Each agent maximizes for itself.
- Multi-agent MDPs: value function for each agent.
  each agent maximizes its own value function.
- Multi-agent reinforcement learning: each agent has its own Q function.
- Two person, competitive (zero sum) $\Longrightarrow$ minimax.

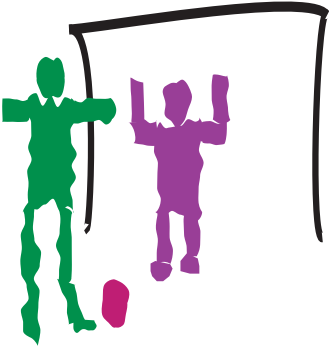
```
procedure Minimax(n)
                                ▷ n is a node. Returns value of n, path
    if n is a leaf node then
        return evaluate(n), None
    else if n is a MAX node then
        max_score $= -\infty$;  max_path=None
        for each child c of n do
            score, path := Minimax(c)
            if score $>$ max_score then
                max_score := score ;  best_path := n : path
        return max_score, best_path
    else
        min_score $= \infty$;  max_path=None
        for each child c of n do
            score, path := Minimax(c)
            if score $<$ min_score then
                min_score := score ;  best_path := c : path
        return min_score, best_path
```

Pruning Dominated Strategies



square MAX nodes are controlled by an agent that wants to maximize the score, round MIN nodes are controlled by an adversary who wants to minimize the score.

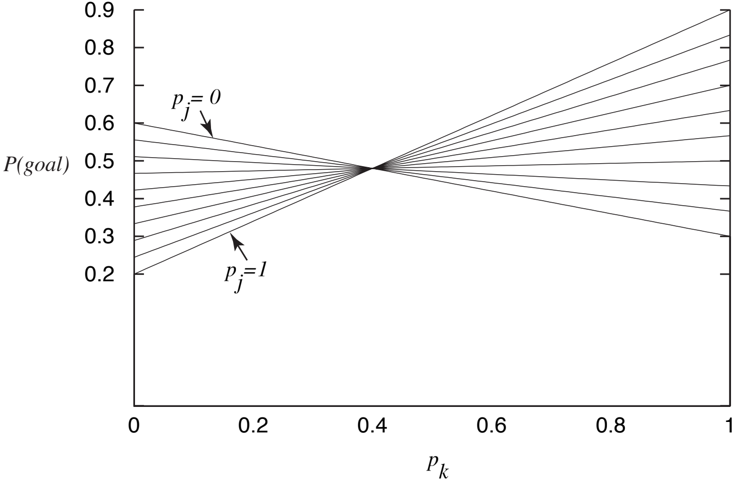
Partial Observability and Competition



		goalkeeper left	 right
kicker	left	0.6	0.2
	right	0.3	0.9

Probability of a goal.

Stochastic Policies



Strategy Profiles

- Assume a general *n*-player game,
- A strategy for an agent is a probability distribution over the actions for this agent.

Strategy Profiles

- Assume a general *n*-player game,
- A strategy for an agent is a probability distribution over the actions for this agent.
- A strategy profile is an assignment of a strategy to each agent.

Strategy Profiles

- Assume a general *n*-player game,
- A strategy for an agent is a probability distribution over the actions for this agent.
- A strategy profile is an assignment of a strategy to each agent.
- A strategy profile σ has a utility for each agent.
  Let $utility(\sigma, i)$ be the utility of strategy profile σ for agent i.
- If σ is a strategy profile:
  σ_i is the strategy of agent i in σ,
  σ_{-i} is the set of strategies of the other agents.
  Thus σ is $\sigma_i\sigma_{-i}$

Nash Equilibria

- σ_i is a best response to σ_{-i} if for all other strategies σ'_i for agent i,

$$utility(\sigma_i \sigma_{-i}, i) \geq utility(\sigma'_i \sigma_{-i}, i).$$

Nash Equilibria

- σ_i is a best response to σ_{-i} if for all other strategies σ'_i for agent i,

$$utility(\sigma_i \sigma_{-i}, i) \geq utility(\sigma'_i \sigma_{-i}, i).$$

- A strategy profile σ is a Nash equilibrium if for each agent i, strategy σ_i is a best response to σ_{-i}. That is, a Nash equilibrium is a strategy profile such that no agent can do better by unilaterally deviating from that profile.

Nash Equilibria

- σ_i is a best response to σ_{-i} if for all other strategies σ'_i for agent i,

$$utility(\sigma_i\sigma_{-i}, i) \geq utility(\sigma'_i\sigma_{-i}, i).$$

- A strategy profile σ is a Nash equilibrium if for each agent i, strategy σ_i is a best response to σ_{-i}. That is, a Nash equilibrium is a strategy profile such that no agent can do better by unilaterally deviating from that profile.
- Theorem [Nash, 1950] Every finite game has at least one Nash equilibrium.

Multiple Equilibria

Hawk-Dove Game:

		Agent 2	
		dove	hawk
Agent 1	dove	R/2,R/2	0,R
	hawk	R,0	-D,-D

D and R are both positive with $D >> R$.

Coordination

Just because you know the Nash equilibria doesn't mean you know what to do:

		Agent 2	
		shopping	football
Agent 1	shopping	2,1	0,0
	football	0,0	1,2

Prisoner's Dilemma

Two strangers are in a game show. They each have the choice:

- Take $100 for yourself
- Give $1000 to the other player

This can be depicted as the playoff matrix:

		Player 2	
		take	give
Player 1	take	100,100	1100,0
	give	0,1100	1000,1000

Tragedy of the Commons

Example:

- There are 100 agents.
- There is an common environment that is shared amongst all agents. Each agent has 1/100 of the shared environment.
- Each agent can choose to do an action that has a payoff of +10 but has a -100 payoff on the environment
  or do nothing with a zero payoff

Tragedy of the Commons

Example:

- There are 100 agents.
- There is an common environment that is shared amongst all agents. Each agent has 1/100 of the shared environment.
- Each agent can choose to do an action that has a payoff of +10 but has a -100 payoff on the environment
  or do nothing with a zero payoff
- For each agent, doing the action has a payoff of

Tragedy of the Commons

Example:

- There are 100 agents.
- There is an common environment that is shared amongst all agents. Each agent has $1/100$ of the shared environment.
- Each agent can choose to do an action that has a payoff of $+10$ but has a -100 payoff on the environment
  or do nothing with a zero payoff
- For each agent, doing the action has a payoff of
  $10 - 100/100 = 9$
- If every agent does the action the total payoff is

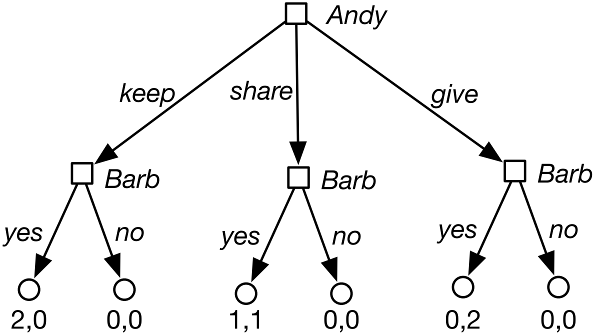
Tragedy of the Commons

Example:

- There are 100 agents.
- There is an common environment that is shared amongst all agents. Each agent has 1/100 of the shared environment.
- Each agent can choose to do an action that has a payoff of +10 but has a -100 payoff on the environment
  or do nothing with a zero payoff
- For each agent, doing the action has a payoff of
  $10 - 100/100 = 9$
- If every agent does the action the total payoff is
  $1000 - 10000 = -9000$

Extensive Form of a Game

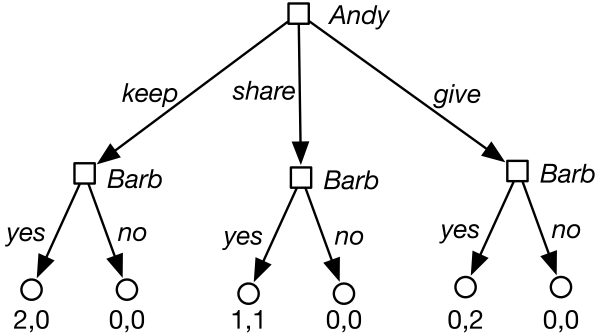
What are the Nash equilibria of:



A strategy for Barb is a choice of what to do in each situation.
Action profile eg 1: Andy: keep, Barb: no if keep, otherwise yes.
Action profile eg 2: Andy: share, Barb: yes always

Extensive Form of a Game

What are the Nash equilibria of:



A strategy for Barb is a choice of what to do in each situation.
Action profile eg 1: Andy: keep, Barb: no if keep, otherwise yes.
Action profile eg 2: Andy: share, Barb: yes always
What if the 2,0 payoff was 1.9,0.1?

Computing Nash Equilibria

To compute a Nash equilibria for a game in strategic form:

- Eliminate dominated strategies
- Determine which actions will have non-zero probabilities. This is the support set.
- Determine the probability for the actions in the support set

Eliminating Dominated Strategies

Agent 2

		d_2	e_2	f_2
	a_1	3,5	5,1	1,2
Agent 1	b_1	1,1	2,9	6,4
	c_1	2,6	4,7	0,8

Computing probabilities in randomized strategies

Given a support set:

- Why would an agent will randomize between actions $a_1 \ldots a_k$?

Computing probabilities in randomized strategies

Given a support set:

- Why would an agent will randomize between actions $a_1 \ldots a_k$? Actions $a_1 \ldots a_k$ have the same value for that agent given the strategies for the other agents.

Computing probabilities in randomized strategies

Given a support set:

- Why would an agent will randomize between actions $a_1 \dots a_k$? Actions $a_1 \dots a_k$ have the same value for that agent given the strategies for the other agents.
- This forms a set of simultaneous equations where variables are probabilities of the actions

Computing probabilities in randomized strategies

Given a support set:

- Why would an agent will randomize between actions $a_1 \ldots a_k$? Actions $a_1 \ldots a_k$ have the same value for that agent given the strategies for the other agents.
- This forms a set of simultaneous equations where variables are probabilities of the actions
- If there is a solution with all the probabilities in range (0,1) this is a Nash equilibrium.

Computing probabilities in randomized strategies

Given a support set:

- Why would an agent will randomize between actions $a_1 \ldots a_k$? Actions $a_1 \ldots a_k$ have the same value for that agent given the strategies for the other agents.
- This forms a set of simultaneous equations where variables are probabilities of the actions
- If there is a solution with all the probabilities in range (0,1) this is a Nash equilibrium.

Search over support sets to find a Nash equilibrium

Example: computing Nash equilibrium

		goalkeeper	
		left	right
kicker	left	0.6	0.2
	right	0.3	0.9

Probability of a goal.

When would goalkeeper randomize?

Example: computing Nash equilibrium

		goalkeeper left	goalkeeper right
kicker	left	0.6	0.2
	right	0.3	0.9

Probability of a goal.

When would goalkeeper randomize?

$$P(goal \mid jump\ left) \quad = \quad p(goal \mid jump\ right)$$

Example: computing Nash equilibrium

		goalkeeper left	 right
kicker	left	0.6	0.2
	right	0.3	0.9

Probability of a goal.

When would goalkeeper randomize?

$$\begin{aligned} P(goal \mid jump\ left) &= p(goal \mid jump\ right) \\ kr * 0.3 + (1 - kr) * 0.6 &= kr * 0.9 + (1 - kr) * 0.2 \end{aligned}$$

Example: computing Nash equilibrium

		goalkeeper	
		left	right
kicker	left	0.6	0.2
	right	0.3	0.9

Probability of a goal.

When would goalkeeper randomize?

$$\begin{aligned}
P(goal \mid jump\ left) &= p(goal \mid jump\ right) \\
kr * 0.3 + (1 - kr) * 0.6 &= kr * 0.9 + (1 - kr) * 0.2 \\
0.6 - 0.2 &= (0.6 - 0.3 + 0.9 - 0.2) * kr \\
kr &= 0.4
\end{aligned}$$

Learning to Coordinate (multiple agents, single state)

- Each agent maintains $P[A]$ a probability distribution over actions.
- Each agent maintains $Q[A]$ an estimate of value of doing A given policy of other agents.
- Repeat:
  - select action a using distribution P,
  - do a and observe payoff
  - update Q:

Learning to Coordinate (multiple agents, single state)

- Each agent maintains $P[A]$ a probability distribution over actions.
- Each agent maintains $Q[A]$ an estimate of value of doing A given policy of other agents.
- Repeat:
  - select action a using distribution P,
  - do a and observe payoff
  - update Q: $Q[a] \leftarrow Q[a] + \alpha(payoff - Q[a])$
  - incremented probability of best action by δ.
  - decremented probability of other actions