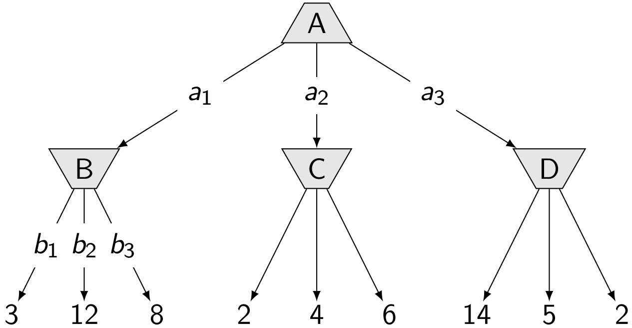
Uncertainty, Chance, and Utilities

Tomáš Svoboda and Petr Pošík

Vision for Robots and Autonomous Systems, Center for Machine Perception
Department of Cybernetics
Faculty of Electrical Engineering, Czech Technical University in Prague

March 22, 2023

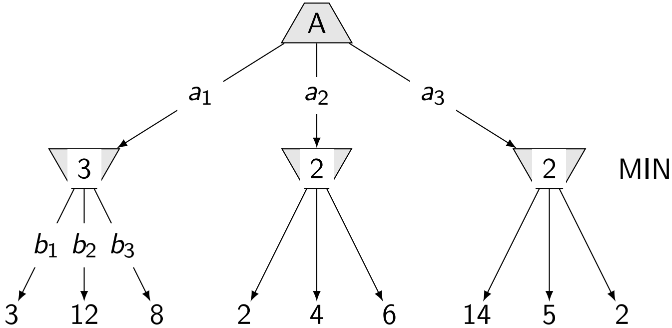
Deterministic opponent → stochastic environment



b_1, b_2, b_3 - stochastic branches, uncertain outcomes of a_1 action.

CHANCE nodes are "virtual", b_1, b_2, b_3 are not actions!

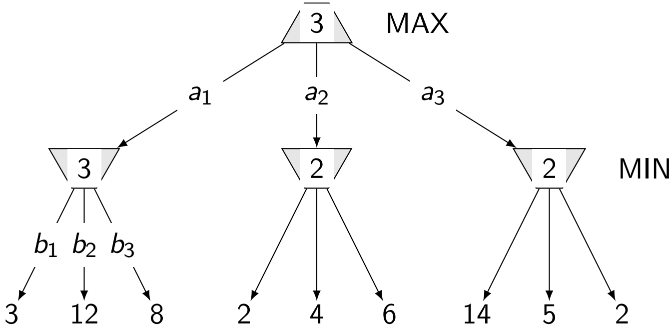
Deterministic opponent → stochastic environment



b_1, b_2, b_3 - stochastic branches, uncertain outcomes of a_1 action.

CHANCE nodes are "virtual", b_1, b_2, b_3 are not actions!

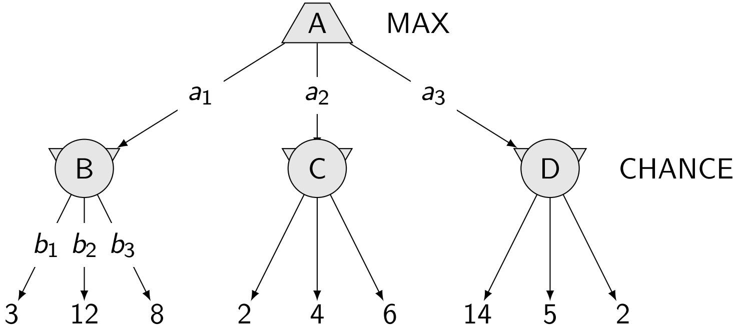
Deterministic opponent → stochastic environment



b_1, b_2, b_3 - stochastic branches, uncertain outcomes of a_1 action.

CHANCE nodes are "virtual", b_1, b_2, b_3 are not actions!

Deterministic opponent → stochastic environment



b_1, b_2, b_3 - stochastic branches, uncertain outcomes of a_1 action.

CHANCE nodes are "virtual", b_1, b_2, b_3 are not actions!

Why? Actions may fail, . . .

Video: Slipping robot. Vision for Robotics and Autonomous Systems, http://cyber.felk.cvut.cz/vras, https://youtu.be/kvEEHNyCHMs

Why? Action costs not deterministic, . . . , getting to work

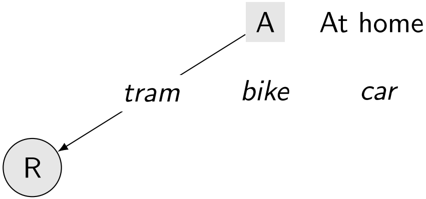
A At home

tram *bike* *car*

Random variable: Function mapping situation on rails to values $T(r_i) = t_i$:
$t_1 = T(r_1) = 3$ mins (free rails)
$t_2 = T(r_2) = 12$ mins (accident)
$t_3 = T(r_3) = 8$ mins (congestion)

MAX/MIN depends on what the t_i options and terminal numbers mean. The goal may be to get to work as fast as possible.

Why? Action costs not deterministic, . . . , getting to work



Random variable: Function mapping situation on rails to values $T(r_i) = t_i$:

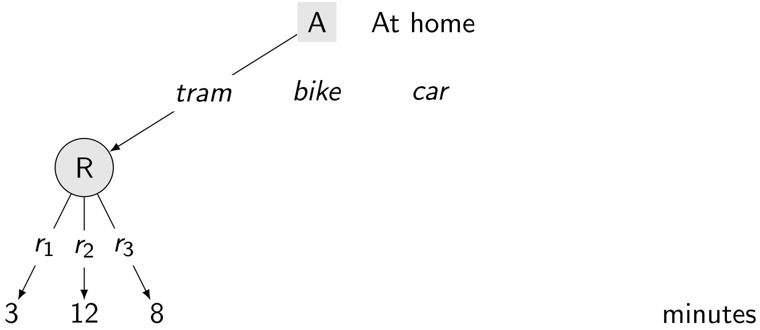
$t_1 = T(r_1) = 3$ mins (free rails)

$t_2 = T(r_2) = 12$ mins (accident)

$t_3 = T(r_3) = 8$ mins (congestion)

MAX/MIN depends on what the t_i options and terminal numbers mean. The goal may be to get to work as fast as possible.

Why? Action costs not deterministic, . . . , getting to work



Random variable: Function mapping situation on rails to values $T(r_i) = t_i$:

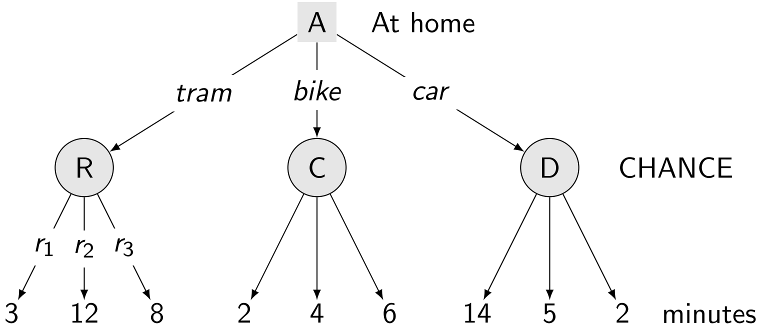
$t_1 = T(r_1) = 3$ mins (free rails)

$t_2 = T(r_2) = 12$ mins (accident)

$t_3 = T(r_3) = 8$ mins (congestion)

MAX/MIN depends on what the t_i options and terminal numbers mean. The goal may be to get to work as fast as possible.

Why? Action costs not deterministic, . . . , getting to work



Random variable: Function mapping situation on rails to values $T(r_i) = t_i$:

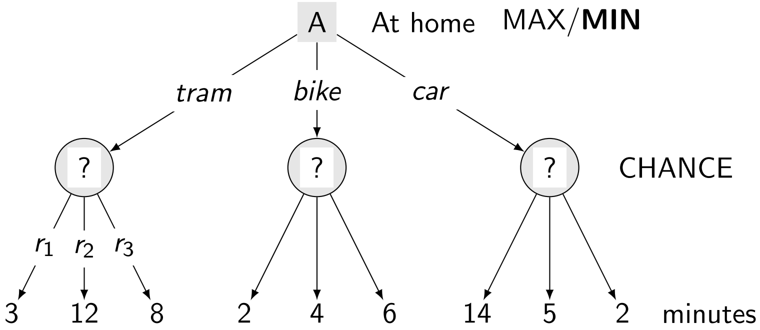
$t_1 = T(r_1) = 3$ mins (free rails)

$t_2 = T(r_2) = 12$ mins (accident)

$t_3 = T(r_3) = 8$ mins (congestion)

MAX/MIN depends on what the t_i options and terminal numbers mean. The goal may be to get to work as fast as possible.

Why? Action costs not deterministic, . . . , getting to work



Random variable: Function mapping situation on rails to values $T(r_i) = t_i$:

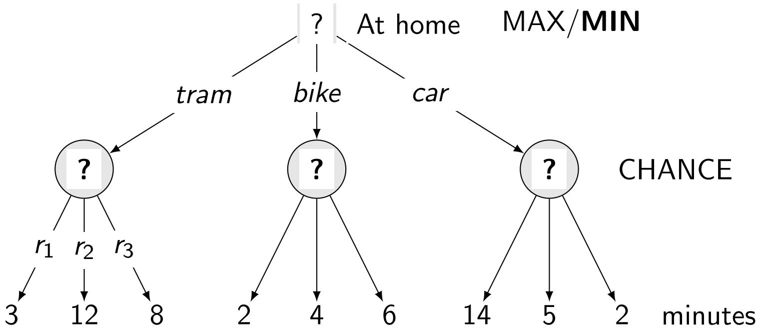
$t_1 = T(r_1) = 3$ mins (free rails)

$t_2 = T(r_2) = 12$ mins (accident)

$t_3 = T(r_3) = 8$ mins (congestion)

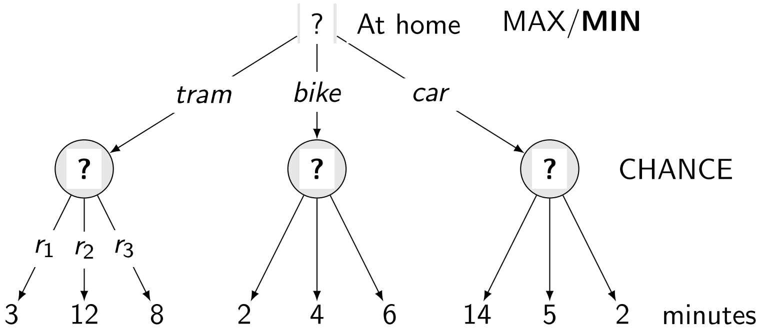
MAX/MIN depends on what the t_i options and terminal numbers mean. The goal may be to get to work as fast as possible.

Chance nodes values



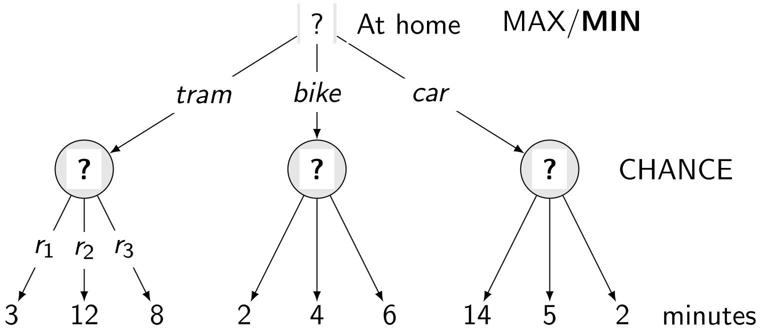
- Average case, not the worst case.
- Calculate expected utilities ...
- i.e. take weighted average (expectation) of successors

Chance nodes values



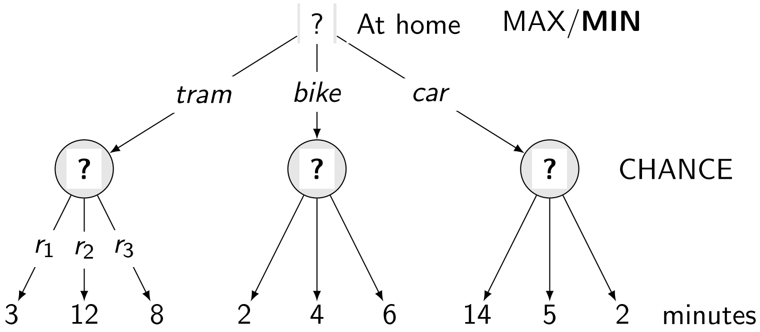
- Average case, not the worst case.
- Calculate expected utilities . . .
- i.e. take weighted average (expectation) of successors

Chance nodes values



- Average case, not the worst case.
- Calculate expected utilities . . .
- i.e. take weighted average (expectation) of successors

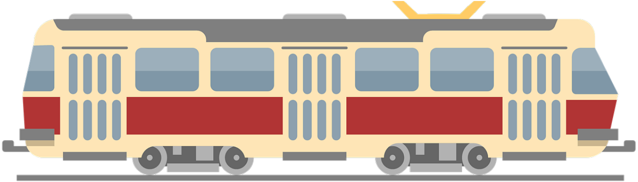
Chance nodes values



- Average case, not the worst case.
- Calculate expected utilities . . .
- i.e. take weighted average (expectation) of successors

Random variables, probability distribution, ...

- Random variable - a function that maps experiment outcomes to values
- Probability distribution - assignment of probabilities (weights) to the values



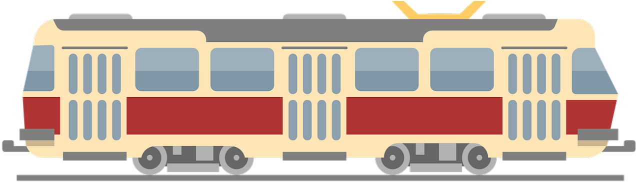
- Random variable: $T(s)$ - maps situation on rails to values
- Values of $T(s)$: $T(s) \in \{3, 12, 8\}$, corresponding to outcomes s (free rails, accident, congestion)
- Probability distribution: $P(T = 3) = 0.3$, $P(T = 12) = 0.1$, $P(T = 8) = 0.6$

A few reminders from laws of probability. Probabilities:

- always non-negative,
- sum over all possible outcomes is equal to 1.

Random variables, probability distribution, ...

- Random variable - a function that maps experiment outcomes to values
- Probability distribution - assignment of probabilities (weights) to the values



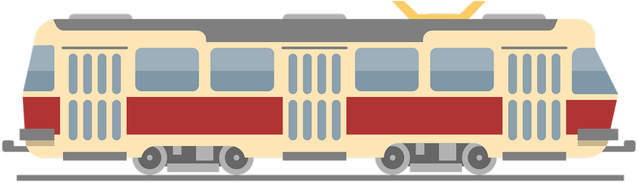
- Random variable: $T(s)$ - maps situation on rails to values
- Values of $T(s)$: $T(s) \in \{3, 12, 8\}$, corresponding to outcomes s (free rails, accident, congestion)
- Probability distribution: $P(T = 3) = 0.3$, $P(T = 12) = 0.1$, $P(T = 8) = 0.6$

A few reminders from laws of probability. Probabilities:

- always non-negative,
- sum over all possible outcomes is equal to 1.

Random variables, probability distribution, ...

- Random variable - a function that maps experiment outcomes to values
- Probability distribution - assignment of probabilities (weights) to the values



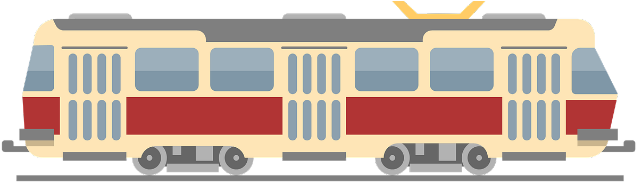
- Random variable: $T(s)$ - maps situation on rails to values
- Values of $T(s)$: $T(s) \in \{3, 12, 8\}$, corresponding to outcomes s (free rails, accident, congestion)
- Probability distribution: $P(T = 3) = 0.3$, $P(T = 12) = 0.1$, $P(T = 8) = 0.6$

A few reminders from laws of probability. Probabilities:

- always non-negative,
- sum over all possible outcomes is equal to 1.

Random variables, probability distribution, ...

- Random variable - a function that maps experiment outcomes to values
- Probability distribution - assignment of probabilities (weights) to the values



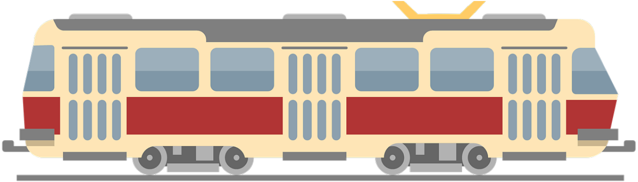
- Random variable: $T(s)$ - maps situation on rails to values
- Values of $T(s)$: $T(s) \in \{3, 12, 8\}$, corresponding to outcomes s (free rails, accident, congestion)
- Probability distribution: $P(T = 3) = 0.3$, $P(T = 12) = 0.1$, $P(T = 8) = 0.6$

A few reminders from laws of probability. Probabilities:

- always non-negative,
- sum over all possible outcomes is equal to 1.

Random variables, probability distribution, . . .

- Random variable - a function that maps experiment outcomes to values
- Probability distribution - assignment of probabilities (weights) to the values



- Random variable: $T(s)$ - maps situation on rails to values
- Values of $T(s)$: $T(s) \in \{3, 12, 8\}$, corresponding to outcomes s (free rails, accident, congestion)
- Probability distribution: $P(T = 3) = 0.3$, $P(T = 12) = 0.1$, $P(T = 8) = 0.6$

A few reminders from laws of probability, Probabilities:

- always non-negative,
- sum over all possible outcomes is equal to 1.

Expectations, ...

How long does it take to go to work by tram?

- Depends on the random variable T with possible values t_1, t_2, t_3 (corresponding to situation on rails).
- What is the expectation of the time?

  Using values t_1, t_2, t_3 of random variable T:

$$E(T) = P(t_1)t_1 + P(t_2)t_2 + P(t_3)t_3$$

  Or, using random outcomes r_1, r_2, r_3:

$$E(T) = P(r_1)T(r_1) + P(r_2)T(r_2) + P(r_3)T(r_3)$$

Expected value of a discrete r.v.: Weighted average

Expectations, . . .

How long does it take to go to work by tram?

- Depends on the random variable T with possible values t_1, t_2, t_3 (corresponding to situation on rails).
- What is the expectation of the time?
  Using values t_1, t_2, t_3 of random variable T:

$$E(T) = P(t_1)t_1 + P(t_2)t_2 + P(t_3)t_3$$

Or, using random outcomes r_1, r_2, r_3:

$$E(T) = P(r_1)T(r_1) + P(r_2)T(r_2) + P(r_3)T(r_3)$$

Expected value of a discrete r.v.: Weighted average

Expectations, . . .

How long does it take to go to work by tram?

- Depends on the random variable T with possible values t_1, t_2, t_3 (corresponding to situation on rails).
- What is the expectation of the time?
  Using values t_1, t_2, t_3 of random variable T:

$$E(T) = P(t_1)t_1 + P(t_2)t_2 + P(t_3)t_3$$

  Or, using random outcomes r_1, r_2, r_3:

$$E(T) = P(r_1)T(r_1) + P(r_2)T(r_2) + P(r_3)T(r_3)$$

Expected value of a discrete r.v.: Weighted average

Expectimax

function EXPECTIMAX(state) **return** a value
  if IS-TERMINAL(state): **return** UTILITY(state)
  if state (next agent) is **MAX**: **return** MAX-VALUE(state)
  if state (next agent) is **CHANCE**: **return** EXP-VALUE(state)
end function

---

function MAX-VALUE(state) **return** value v
  $v \leftarrow -\infty$
  for a **in** ACTIONS(state) **do**
    $v \leftarrow$ **max**(v, EXPECTIMAX(RESULT(state,a)))
  end for
end function

---

function EXP-VALUE(state) **return** value v
  $v \leftarrow 0$
  for all $r \in$ random outcomes **do**
    $v \leftarrow v + P(r)$ EXPECTIMAX(RESULT(state,r))
  end for
end function

Expectimax

```
function EXPECTIMAX(state) return a value
    if IS-TERMINAL(state): return UTILITY(state)
    if state (next agent) is MAX: return MAX-VALUE(state)
    if state (next agent) is CHANCE: return EXP-VALUE(state)
end function
```

```
function MAX-VALUE(state) return value v
    v ← −∞
    for a in ACTIONS(state) do
        v ← max(v, EXPECTIMAX(RESULT(state,a)))
    end for
end function
```

```
function EXP-VALUE(state) return value v
    v ← 0
    for all r ∈ random outcomes do
        v ← v + P(r) EXPECTIMAX(RESULT(state,r))
    end for
end function
```

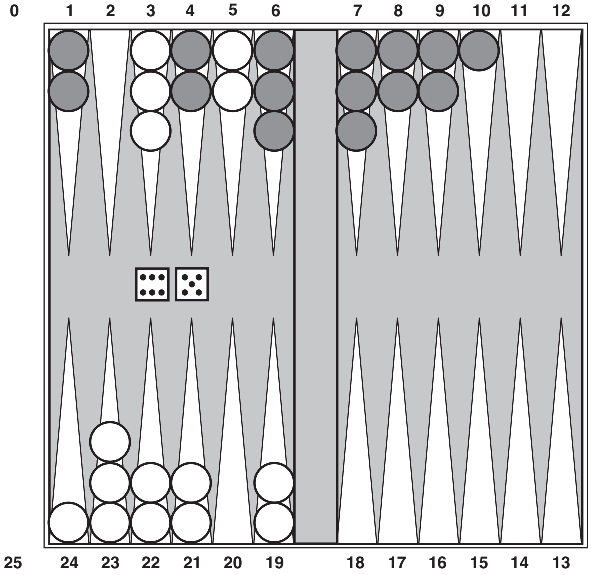
How about the Reversi game?

- Is there any space for randomness?
- Is the opponent really greedy and clever enough?
- Hope for chance when there is adversarial world – Dangerous optimism .
- Assuming worst case even if it is not likely – Dangerous pessimism .

How about the Reversi game?

- Is there any space for randomness?
- Is the opponent really greedy and clever enough?
- Hope for chance when there is adversarial world – Dangerous optimism .
- Assuming worst case even if it is not likely – Dangerous pessimism .

Games with chance **and** strategy



Random variable: Throwing two dice

Do we care which die comes first?

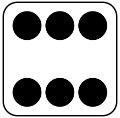
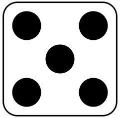
What is the probability of , ?[1]

- A 1/24
- B 1/36
- C 1/18
- D 1/6

[1]Source of dice images: https://flyclipart.com/dice-clipart-tool-rolling-dice-clipart-248574

Random variable: Throwing two dice

Do we care which die comes first?

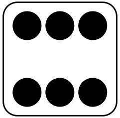
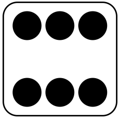
What is the probability of ,  ?[1]

- A 1/24
- B 1/36
- C 1/18
- D 1/6

[1] Source of dice images: https://flyclipart.com/dice-clipart-tool-rolling-dice-clipart-248574

Random variable: Throwing two dice

Do we care which die comes first?

What is the probability of ,  ?[1]

A 1/24

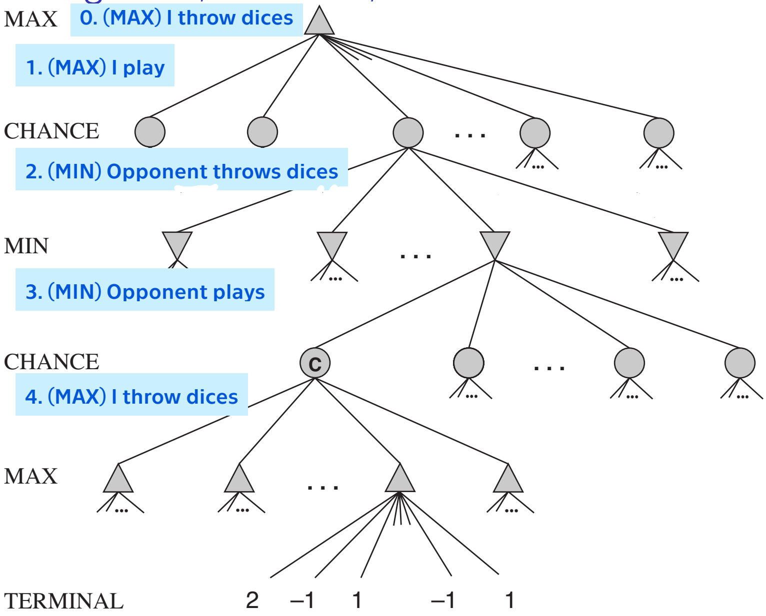
B 1/36

C 1/18

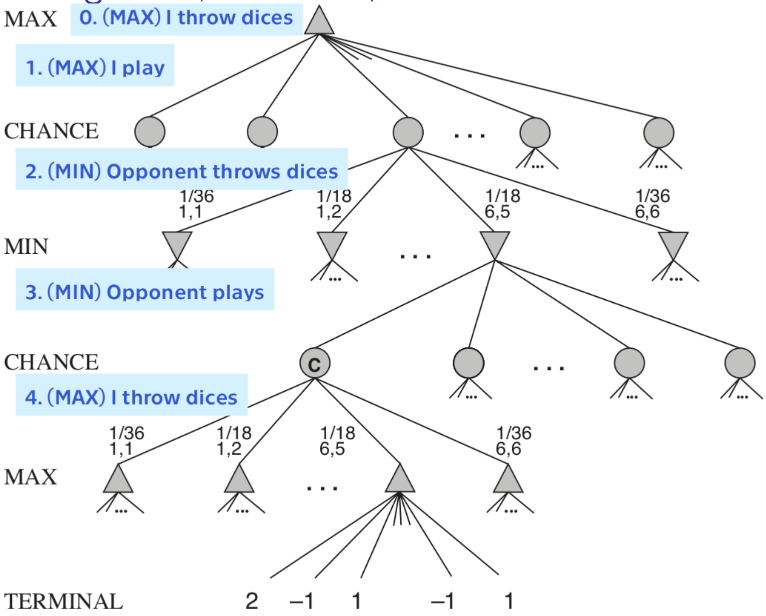
D 1/6

[1]Source of dice images: https://flyclipart.com/dice-clipart-tool-rolling-dice-clipart-248574

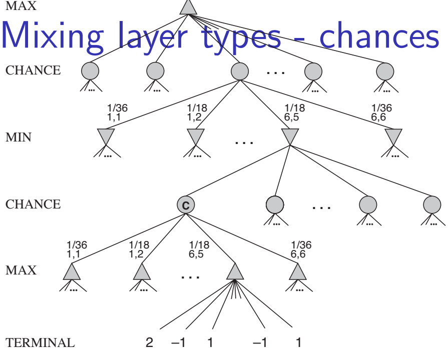
Mixing MAX, CHANCE, and MIN nodes

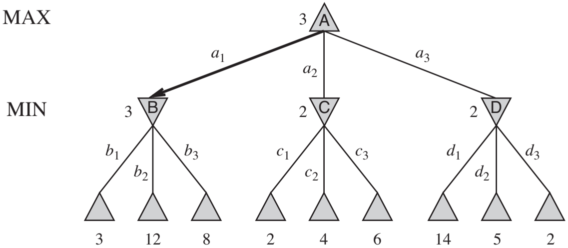


Mixing MAX, CHANCE, and MIN nodes



Mixing layer types - chances inserted

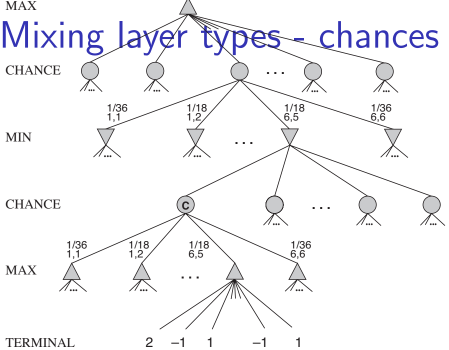


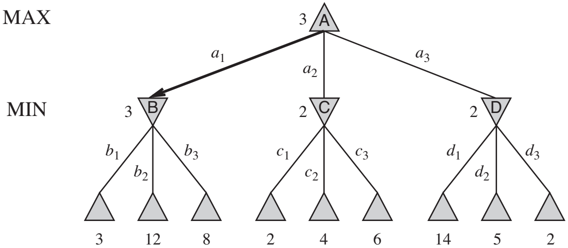


Extra random agent that moves after each MAX and MIN agent

$$\text{EXPECTIMINIMAX}(s) = \\ = \begin{cases} \text{UTILITY}(s, \text{MAX}) & \text{if } \text{IS-TERMINAL}(s) \\ \max_a \text{EXPECTIMINIMAX}(\text{RESULT}(s,a)) & \text{if } \text{TO-PLAY}(s) = \text{MAX} \\ \min_a \text{EXPECTIMINIMAX}(\text{RESULT}(s,a)) & \text{if } \text{TO-PLAY}(s) = \text{MIN} \\ \sum_r P(r)\text{EXPECTIMINIMAX}(\text{RESULT}(s,r)) & \text{if } \text{TO-PLAY}(s) = \text{CHANCE} \end{cases}$$

Mixing layer types - chances inserted



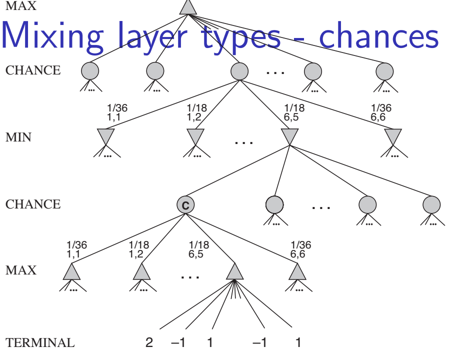


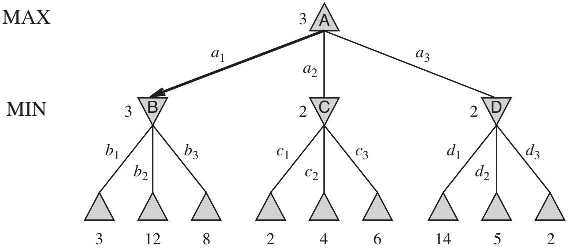
Extra random agent that moves after each MAX and MIN agent

$$\text{EXPECTIMINIMAX}(s) =$$

$$= \begin{cases} \text{UTILITY}(s, \text{MAX}) & \text{if } \text{IS-TERMINAL}(s) \\ \max_a \text{EXPECTIMINIMAX}(\text{RESULT}(s, a)) & \text{if } \text{TO-PLAY}(s) = \text{MAX} \\ \min_a \text{EXPECTIMINIMAX}(\text{RESULT}(s, a)) & \text{if } \text{TO-PLAY}(s) = \text{MIN} \\ \sum_r P(r)\text{EXPECTIMINIMAX}(\text{RESULT}(s, r)) & \text{if } \text{TO-PLAY}(s) = \text{CHANCE} \end{cases}$$

Mixing layer types - chances inserted



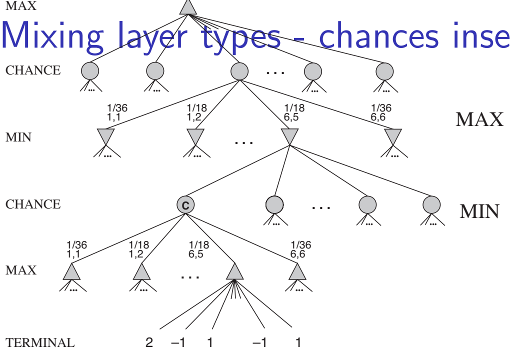


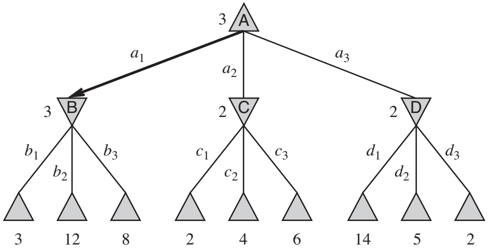
Extra random agent that moves after each MAX and MIN agent

$$\textsc{expectiminimax}(s) =$$

$$= \begin{cases} \textsc{utility}(s, \textsc{max}) & \text{if} \quad \textsc{is-terminal}(s) \\ \max_a \textsc{expectiminimax}(\textsc{result}(s,a)) & \text{if} \quad \textsc{to-play}(s) = \textsc{max} \\ \min_a \textsc{expectiminimax}(\textsc{result}(s,a)) & \text{if} \quad \textsc{to-play}(s) = \textsc{min} \\ \sum_r P(r)\textsc{expectiminimax}(\textsc{result}(s,r)) & \text{if} \quad \textsc{to-play}(s) = \textsc{chance} \end{cases}$$

Mixing layer types - chances inserted



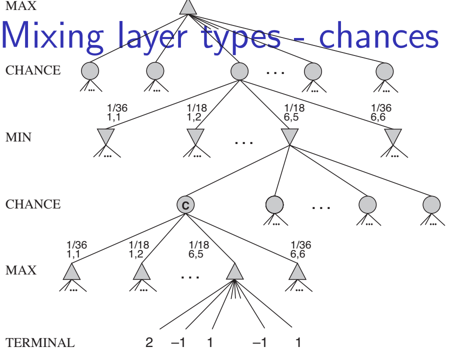


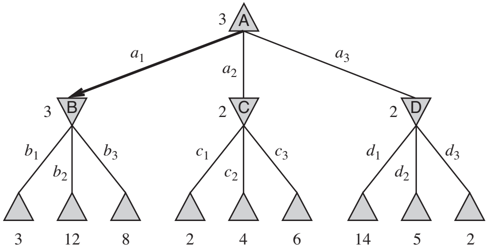
Extra random agent that moves after each MAX and MIN agent

$$\text{EXPECTIMINIMAX}(s) =$$

$$= \begin{cases} \text{UTILITY}(s, \text{MAX}) & \text{if } \text{IS-TERMINAL}(s) \\ \max_a \text{EXPECTIMINIMAX}(\text{RESULT}(s, a)) & \text{if } \text{TO-PLAY}(s) = \text{MAX} \\ \min_a \text{EXPECTIMINIMAX}(\text{RESULT}(s, a)) & \text{if } \text{TO-PLAY}(s) = \text{MIN} \\ \sum_r P(r)\text{EXPECTIMINIMAX}(\text{RESULT}(s, r)) & \text{if } \text{TO-PLAY}(s) = \text{CHANCE} \end{cases}$$

Mixing layer types - chances inserted



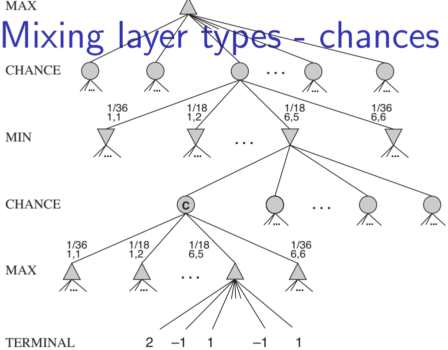


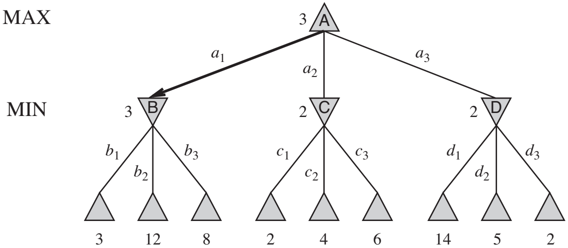
Extra random agent that moves after each MAX and MIN agent

$$\text{EXPECTIMINIMAX}(s) =$$

$$= \begin{cases} \text{UTILITY}(s, \text{MAX}) & \text{if } \text{IS-TERMINAL}(s) \\ \max_a \text{EXPECTIMINIMAX}(\text{RESULT}(s, a)) & \text{if } \text{TO-PLAY}(s) = \text{MAX} \\ \min_a \text{EXPECTIMINIMAX}(\text{RESULT}(s, a)) & \text{if } \text{TO-PLAY}(s) = \text{MIN} \\ \sum_r P(r) \text{EXPECTIMINIMAX}(\text{RESULT}(s, r)) & \text{if } \text{TO-PLAY}(s) = \text{CHANCE} \end{cases}$$

Mixing layer types - chances inserted



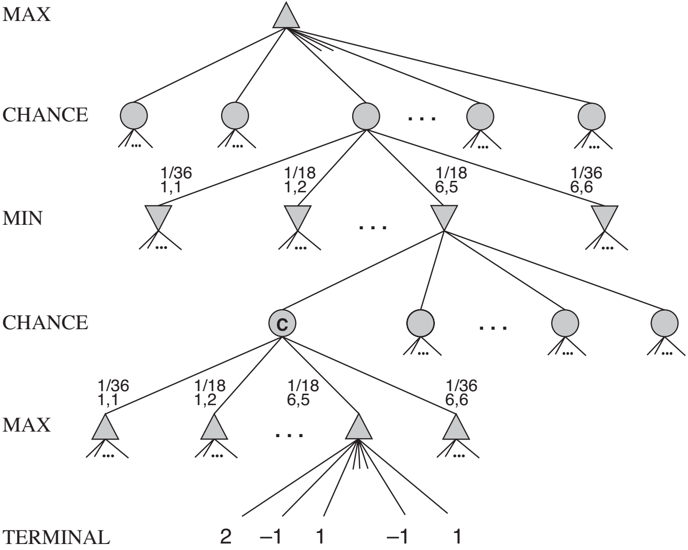


Extra random agent that moves after each MAX and MIN agent

$$\text{EXPECTIMINIMAX}(s) =$$

$$= \begin{cases} \text{UTILITY}(s, \text{MAX}) & \text{if } \text{IS-TERMINAL}(s) \\ \max_a \text{EXPECTIMINIMAX}(\text{RESULT}(s, a)) & \text{if } \text{TO-PLAY}(s) = \text{MAX} \\ \min_a \text{EXPECTIMINIMAX}(\text{RESULT}(s, a)) & \text{if } \text{TO-PLAY}(s) = \text{MIN} \\ \sum_r P(r)\text{EXPECTIMINIMAX}(\text{RESULT}(s, r)) & \text{if } \text{TO-PLAY}(s) = \text{CHANCE} \end{cases}$$

Mixing chance into min/max tree. How big is the tree going to be?



- b branching factor
- m maximum depth
- n number of distinct rolls

What is the time complexity of EXPECTIMINIMAX?

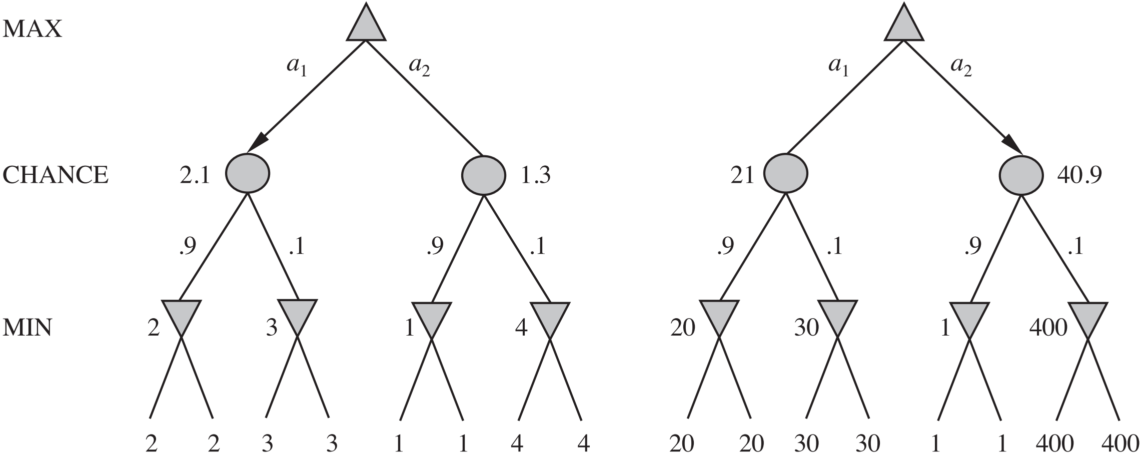
A $O(b^{mn})$

B $O(b^m n)$

C $O(b^m n^b)$

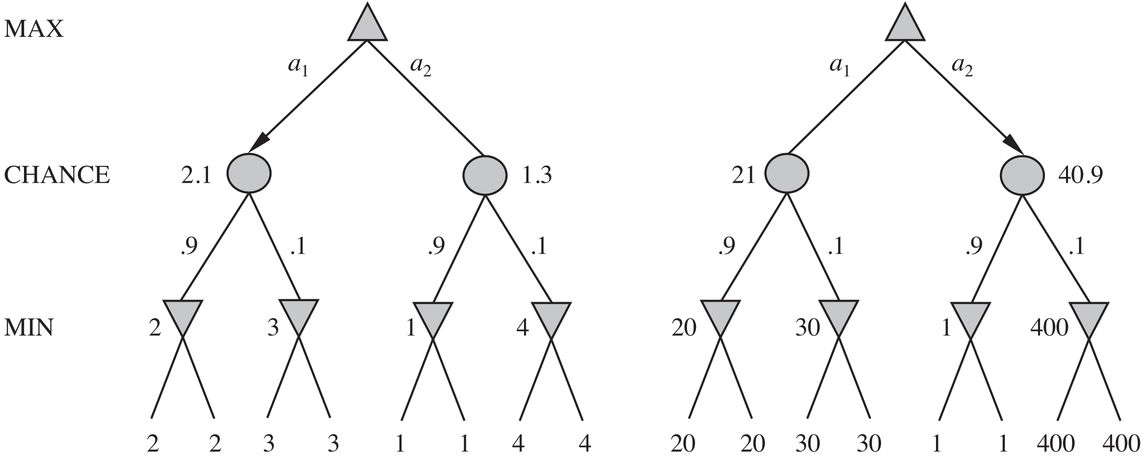
D $O(b^m n^m)$

Evaluation function



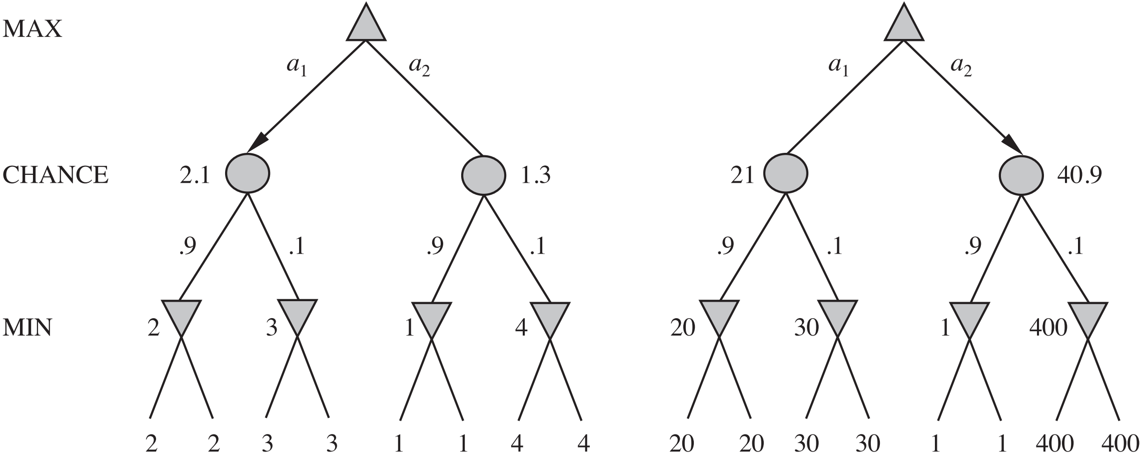
- Left: a_1 is the best. Right: a_2 is the best. Ordering of the (terminal) leaves is the same.
- Scale matters! Not only ordering.
- Can we prune the tree? (α, β like?)

Evaluation function



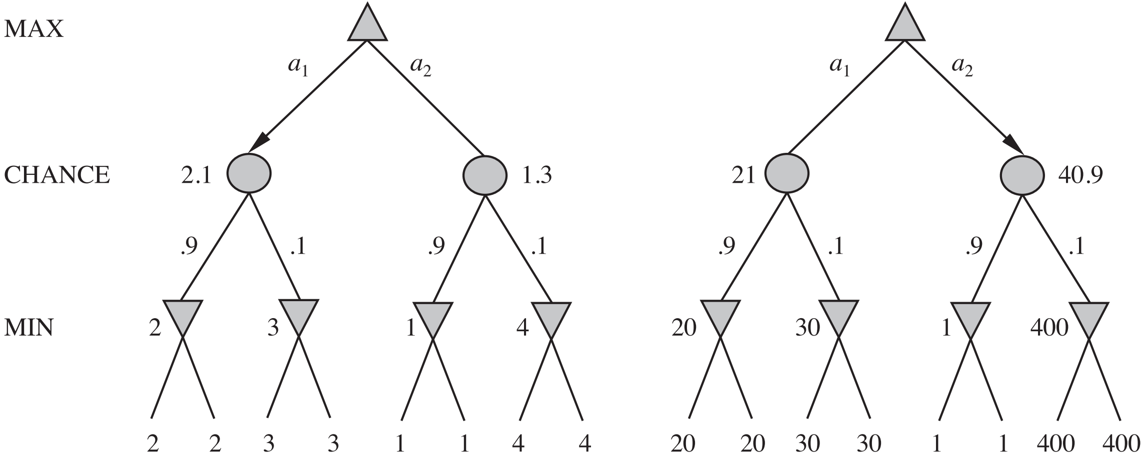
- Left: a_1 is the best. Right: a_2 is the best. Ordering of the (terminal) leaves is the same.
- Scale matters! Not only ordering.
- Can we prune the tree? (α, β like?)

Evaluation function



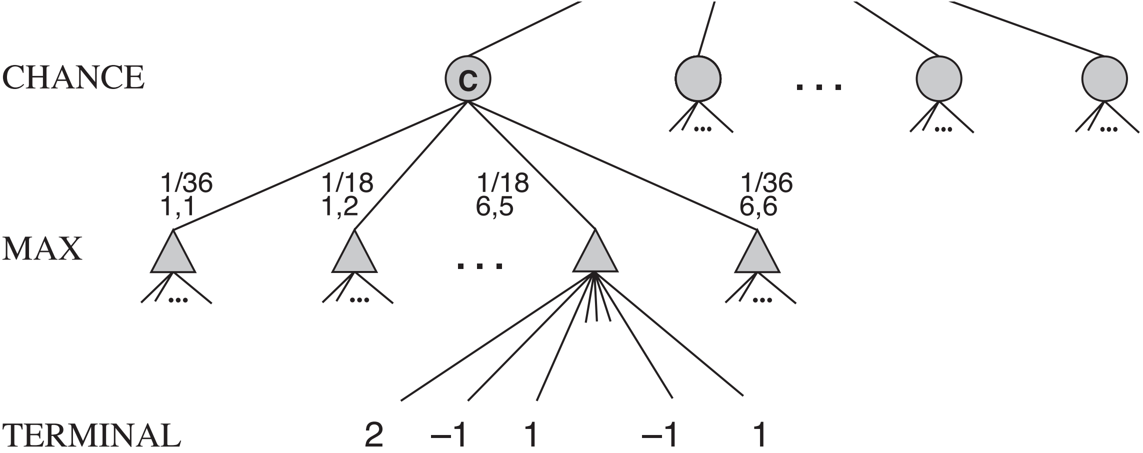
- Left: a_1 is the best. Right: a_2 is the best. Ordering of the (terminal) leaves is the same.
- Scale matters! Not only ordering.
- Can we prune the tree? (α, β like?)

Evaluation function



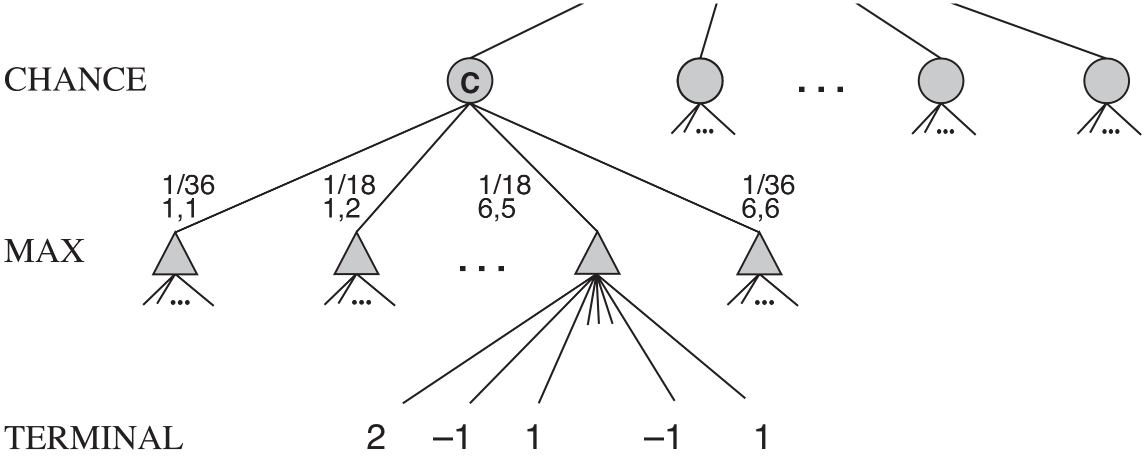
- Left: a_1 is the best. Right: a_2 is the best. Ordering of the (terminal) leaves is the same.
- Scale matters! Not only ordering.
- Can we prune the tree? (α, β like?)

Pruning expectiminimax tree



- Bounds on terminal utilities needed. Terminal values from −2 to 2.
- Monte Carlo simulation for evaluation of a position (state).

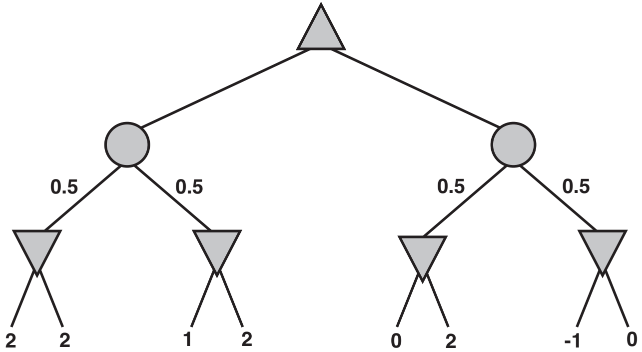
Pruning expectiminimax tree

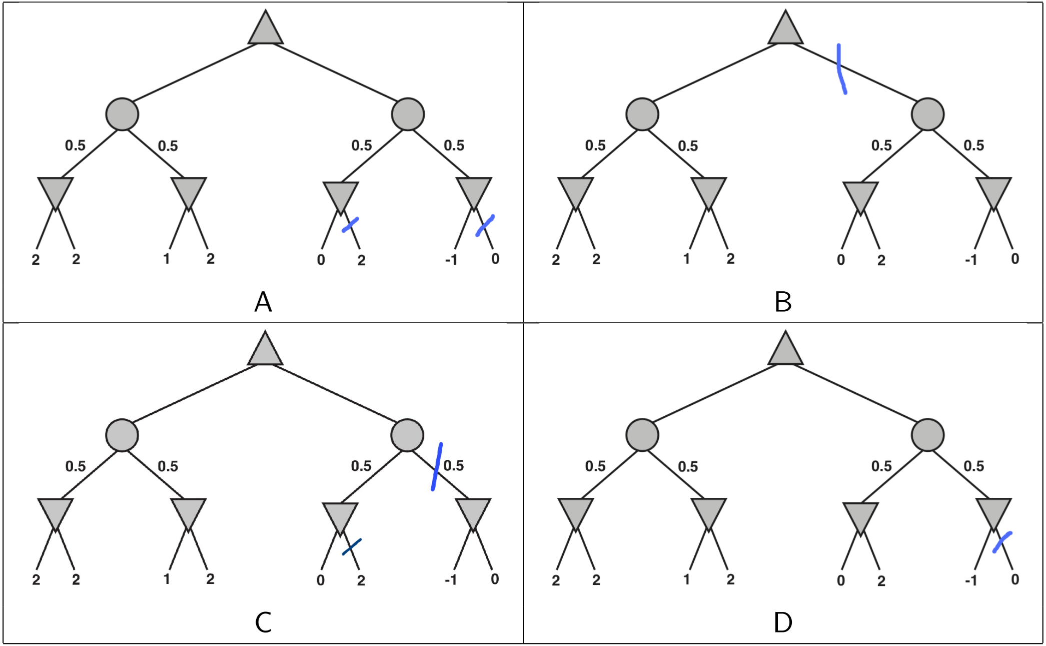


- Bounds on terminal utilities needed. Terminal values from -2 to 2.
- Monte Carlo simulation for evaluation of a position (state).

Where to prune the Expectimax tree

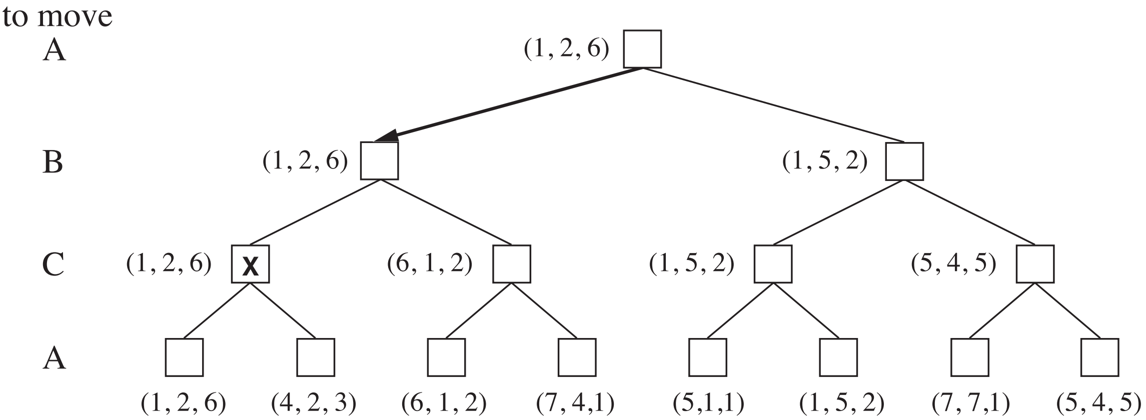
- Assume terminal nodes bounded to -2 to 2, inclusive
- Going from left to right.
- Which branches can be pruned out?





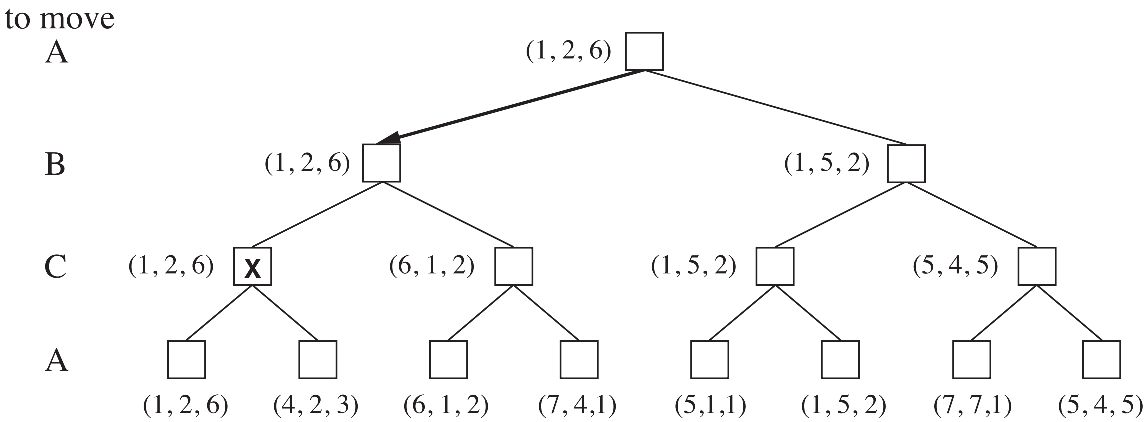
Assume terminal nodes bounded to -2 to 2, inclusive. Going from left to right.

Multi-player games



- Utility tuples
- Each player maximizes its own
- Coalitions, cooperations, competitions may be dynamic

Multi-player games



- Utility tuples
- Each player maximizes its own
- Coalitions, cooperations, competitions may be dynamic

Uncertainty recap (enough games, back to the robots/agents)

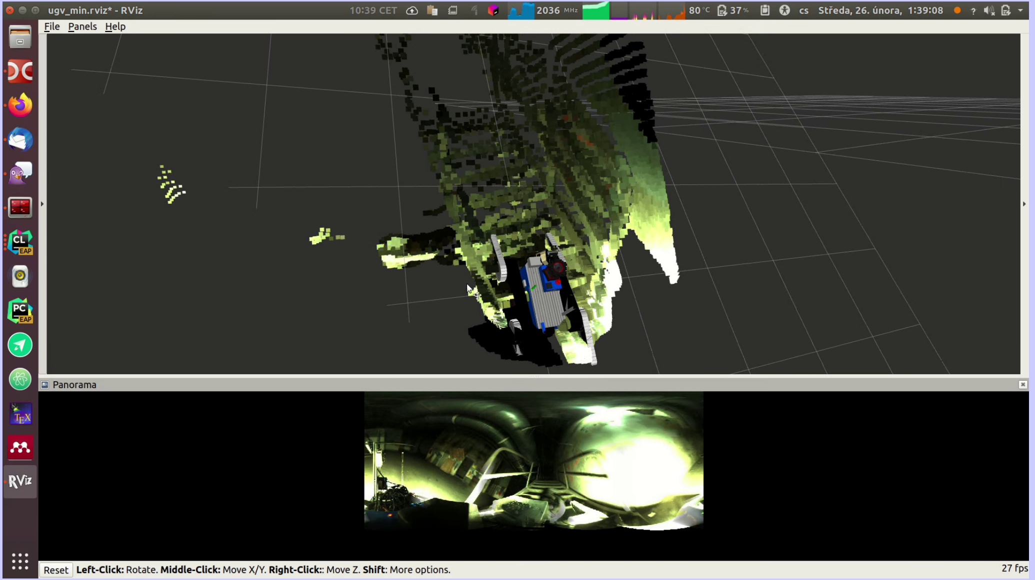
Uncertainty recap (enough games, back to the robots/agents)

- Uncertain outcome of an action.

Uncertainty recap (enough games, back to the robots/agents)

- Uncertain outcome of an action.
- Robot/Agent may not know the current state!

Uncertain outcome of an action



Video: Climbing stairs failure, From: http://robotics.fel.cvut.cz/cras/darpa-subt/

Uncertain, partially observable environment

Amatrice, Italy, 2016.

Uncertain, partially observable environment

- Current state s may be unknown, observations $\mathbf{e}$

$\mathbf{e}$

Amatrice, Italy, 2016.

Uncertain, partially observable environment

- Current state s may be unknown, observations $\mathbf{e}$
- Take action a

$a, \mathbf{e}$

Amatrice, Italy, 2016.

Uncertain, partially observable environment

- Current state s may be unknown, observations $\mathbf{e}$
- Take action a
- Uncertain outcome $\textsc{Result}(a)$

$$\textsc{Result}(a) \qquad a, \mathbf{e}$$

Amatrice, Italy, 2016.

Uncertain, partially observable environment

- Current state s may be unknown, observations $\mathbf{e}$
- Take action a
- Uncertain outcome $\textsc{Result}(a)$
- Probability of outcome s' given $\mathbf{e}$ is

$$P(\textsc{Result}(a) = s' | a, \mathbf{e})$$

Amatrice, Italy, 2016.

Uncertain, partially observable environment

- Current state s may be unknown, observations $\mathbf{e}$
- Take action a
- Uncertain outcome $\text{RESULT}(a)$
- Probability of outcome s' given $\mathbf{e}$ is

$$P(\text{RESULT}(a) = s' | a, \mathbf{e})$$

- Utility function $U(s)$ corresponds to agent preferences.

Amatrice, Italy, 2016.

Uncertain, partially observable environment

- Current state s may be unknown, observations $\mathbf{e}$
- Take action a
- Uncertain outcome $\textsc{Result}(a)$
- Probability of outcome s' given $\mathbf{e}$ is

$$P(\textsc{Result}(a) = s' | a, \mathbf{e})$$

- Utility function $U(s)$ corresponds to agent preferences.
- Expected utility of an action a given $\mathbf{e}$:

$$EU(a|\mathbf{e}) = \sum_{s'} P(\textsc{Result}(a) = s' | a, \mathbf{e}) U(s')$$

Amatrice, Italy, 2016.

Rational agent

Agent's expected utility of an action a given $\mathbf{e}$:

$$EU(a|\mathbf{e}) = \sum_{s'} P(\text{RESULT}(a) = s'|a, \mathbf{e}) U(s')$$

What should a rational agent do?

Is it then all solved? Do we know all what we need?

- $P(\text{RESULT}(a) = s'|a, \mathbf{e})$
- $U(s')$

Rational agent

Agent's expected utility of an action a given $\mathbf{e}$:

$$EU(a|\mathbf{e}) = \sum_{s'} P(\text{RESULT}(a) = s'|a, \mathbf{e})U(s')$$

What should a rational agent do?
Is it then all solved? Do we know all what we need?

- $P(\text{RESULT}(a) = s'|a, \mathbf{e})$
- $U(s')$

Rational agent

Agent's expected utility of an action a given $\mathbf{e}$:

$$EU(a|\mathbf{e}) = \sum_{s'} P(\textsc{Result}(a) = s'|a, \mathbf{e})U(s')$$

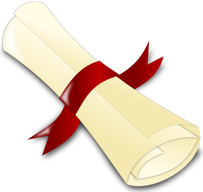
What should a rational agent do?
Is it then all solved? Do we know all what we need?

- $P(\textsc{Result}(a) = s'|a, \mathbf{e})$
- $U(s')$

Utilities







- Where do utilities come from?
- Does averaging make sense?
- Do they exist?
- What if our preferences can't be described by utilities?

Agent/Robot Preferences

- Prizes A, B
- Lottery: uncertain prizes $L = [p, A; (1-p), B]$

Preference, indifference, ...

- Robot prefers A over B: $A \succ B$
- Robot has no preferences: $A \sim B$
- in between: $A \succsim B$

Agent/Robot Preferences

- Prizes A, B
- Lottery: uncertain prizes $L = [p, A; (1-p), B]$

Preference, indifference, ...

- Robot prefers A over B: $A \succ B$
- Robot has no preferences: $A \sim B$
- in between: $A \succsim B$

Rational preferences

- Transitivity: $(A \succ B) \wedge (B \succ C) \Rightarrow (A \succ C)$
- Completeness: $(A \succ B) \vee (B \succ A) \vee (A \sim B)$
- Continuity: $(A \succ B \succ C) \Rightarrow \exists p \, [p, A; 1-p, C] \sim B$
- Substituability: $A \sim B \Rightarrow [p, A; 1-p, C] \sim [p, B; 1-pC]$. The same for $\succ$ and $\sim$.
- Monotonocity: $A \succ B \Rightarrow (p > q) \Leftrightarrow [p, A; 1-p, B] \succ [q, A; 1-q, B]$. Agent must prefer a lottery with higher chance to win.
- Decomposability, compressing compound lotteries into one: $[p, A; 1-p, [q, B; 1-q, C]] \sim [p, A; (1-p)q, B; (1-p)(1-q), C]$

Axioms of utility theory.

Motivation: if agent/robot violates an axiom $\Rightarrow$ irrational agent/robot.

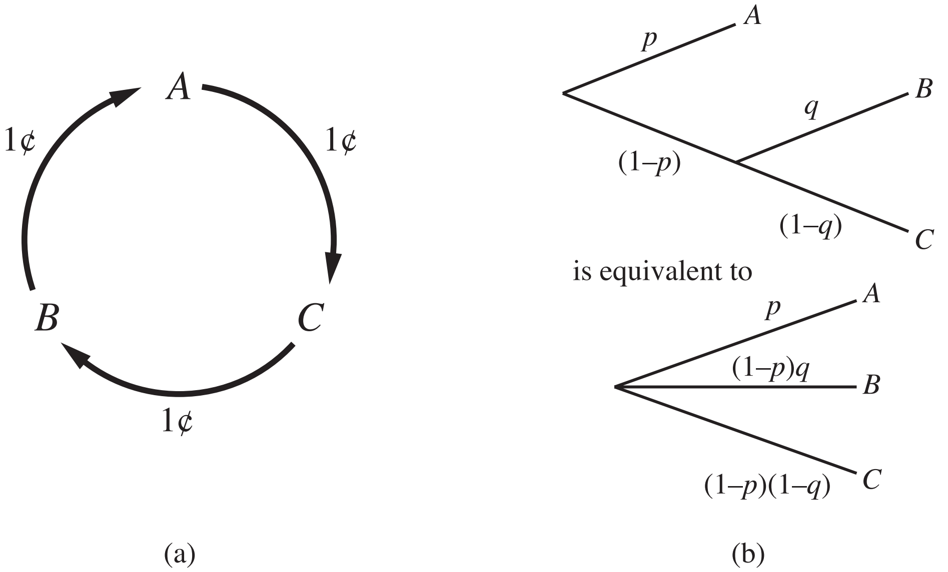
Rational preferences

- Transitivity: $(A \succ B) \wedge (B \succ C) \Rightarrow (A \succ C)$
- Completeness: $(A \succ B) \vee (B \succ A) \vee (A \sim B)$
- Continuity: $(A \succ B \succ C) \Rightarrow \exists p \, [p, A; 1-p, C] \sim B$
- Substituability: $A \sim B \Rightarrow [p, A; 1-p, C] \sim [p, B; 1-pC]$. The same for $\succ$ and $\sim$.
- Monotonocity: $A \succ B \Rightarrow (p > q) \Leftrightarrow [p, A; 1-p, B] \succ [q, A; 1-q, B]$. Agent must prefer a lottery with higher chance to win.
- Decomposability, compressing compound lotteries into one: $[p, A; 1-p, [q, B; 1-q, C]] \sim [p, A; (1-p)q, B; (1-p)(1-q), C]$

Axioms of utility theory.
Motivation: if agent/robot violates an axiom $\Rightarrow$ irrational agent/robot.

Transitivity and decomposability

Goods A, B, C and (nontransitive) preferences of an (irrational) agent $A \succ B \succ C \succ A$.



Maximum expected utility principle

Given the rational preferences (constraints), there exists a real valued function u such that:

$$u(A) > u(B) \quad \Leftrightarrow \quad A \succ B$$
$$u(A) = u(B) \quad \Leftrightarrow \quad A \sim B$$

Expected utility of a Lotery L (outcomes s_i with probabilities p_i):

$$L([p_1, S_1; \cdots ; p_n, S_n]) = \sum_i p_i u(S_i)$$

Proof in [5].
Is a utility u function unique?

Maximum expected utility principle

Given the rational preferences (constraints), there exists a real valued function u such that:

$$
\begin{aligned}
u(A) > u(B) &\quad \Leftrightarrow \quad A \succ B \\
u(A) = u(B) &\quad \Leftrightarrow \quad A \sim B
\end{aligned}
$$

Expected utility of a Lotery L (outcomes s_i with probabilities p_i):

$$L([p_1, S_1; \cdots ; p_n, S_n]) = \sum_i p_i u(S_i)$$

Proof in [5].

Is a utility u function unique?

Maximum expected utility principle

Given the rational preferences (constraints), there exists a real valued function u such that:

$$u(A) > u(B) \quad \Leftrightarrow \quad A \succ B$$
$$u(A) = u(B) \quad \Leftrightarrow \quad A \sim B$$

Expected utility of a Lotery L (outcomes s_i with probabilities p_i):

$$L([p_1, S_1; \cdots ; p_n, S_n]) = \sum_i p_i u(S_i)$$

Proof in [5].
Is a utility u function unique?

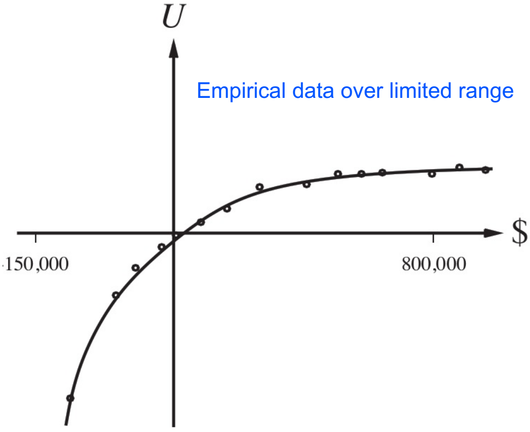
Human utilities

Utility of money

You triumphed in a TV show!

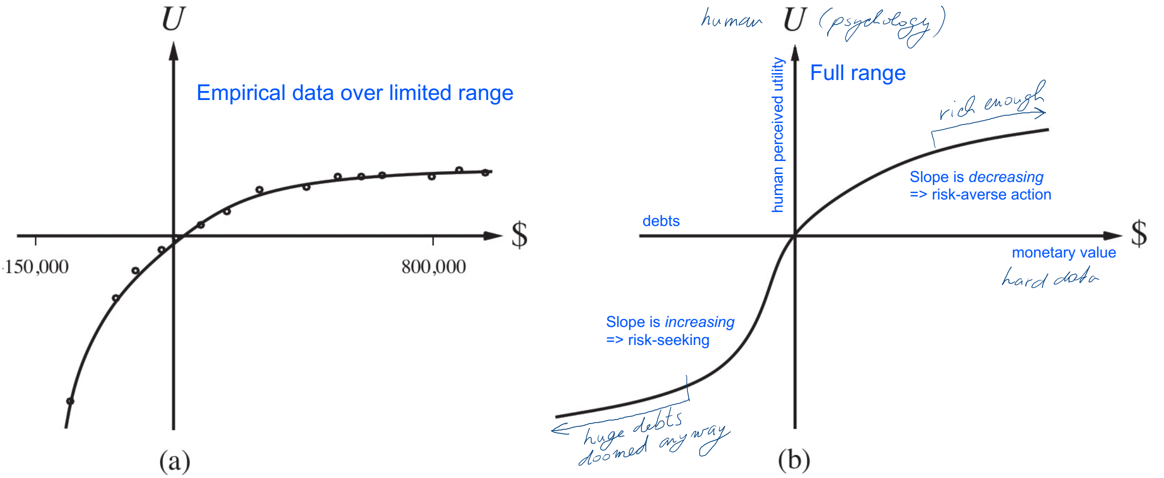
a) Take $\$1,000,000$ ... or
b) Flip a coin and loose all or win $\$2,500,000$

Utility of money: human psychology vs. hard data



(a)

Utility of money: human psychology vs. hard data



References I

Some figures from [3], Chapters 5, 16. Human utilities are discussed in [2]. This lecture has been also greatly inspired by the 7th lecture of CS 188 at http://ai.berkeley.edu as it convenietly bridges the world of deterministic search and sequential decisions in uncertain worlds.

[1] Christopher M. Bishop.
Pattern Recognition and Machine Learning.
Springer Science+Bussiness Media, New York, NY, 2006.
https://www.microsoft.com/en-us/research/uploads/prod/2006/01/Bishop-Pattern-Recognition-and-Machine-Learning-2006.pdf.

[2] Daniel Kahneman.
Thinking, Fast and Slow.
Farrar, Straus and Giroux, 2011.

References II

[3] Stuart Russell and Peter Norvig.
Artificial Intelligence: A Modern Approach.
Prentice Hall, 3rd edition, 2010.
http://aima.cs.berkeley.edu/.

[4] Richard S. Sutton and Andrew G. Barto.
Reinforcement Learning; an Introduction.
MIT Press, 2nd edition, 2018.
http://www.incompleteideas.net/book/the-book-2nd.html.

[5] John von Neumann and Oskar Morgenstern.
Theory of Games and Economic Behavior.
Princeton, 1944.
https://en.wikipedia.org/wiki/Theory_of_Games_and_Economic_Behavior, Utility theorem.