

COMP0271: Inteligência Artificial

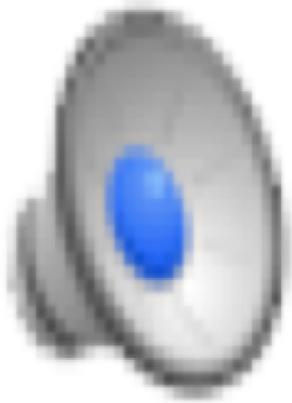
Busca em ambiente com adversários (parte 02)



Professor: Hendrik Macedo

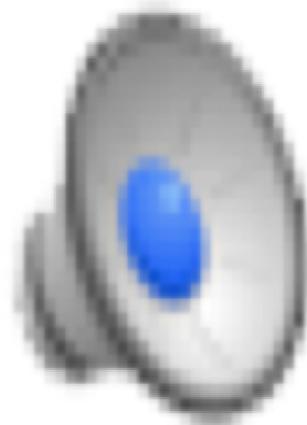
Universidade Federal de Sergipe, Brasil

Como pensa seu adversário?



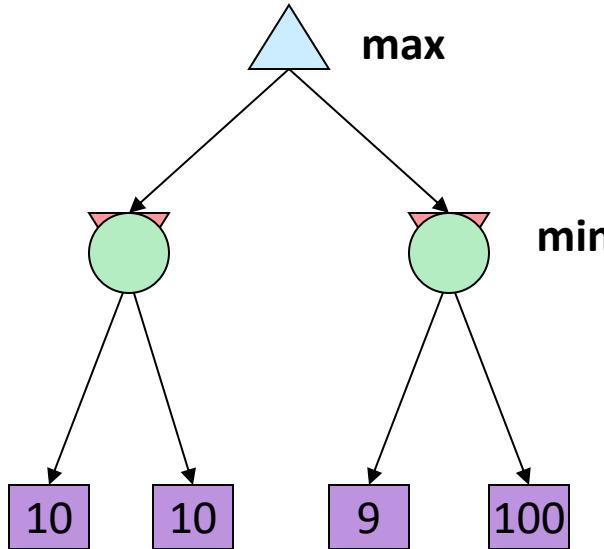
Maximizando a Utilidade mediante
reflexão sobre adversários experts!!!

Como pensa seu adversário?



Maximizando a Utilidade mediante
reflexão sobre **probabilidades!!!**

Adversário → Probabilidade



Nova árvore de jogo:
resultados incertos controlados por probabilidade, não por um adversário!

Expectimax

```
def value(state):
```

 if the state is a terminal state: return the state's utility

 if the next agent is MAX: return max-value(state)

 if the next agent is EXP: return exp-value(state)

```
def max-value(state):
```

 initialize $v = -\infty$

 for each successor of state:

$v = \max(v, \text{value}(\text{successor}))$

 return v

```
def exp-value(state):
```

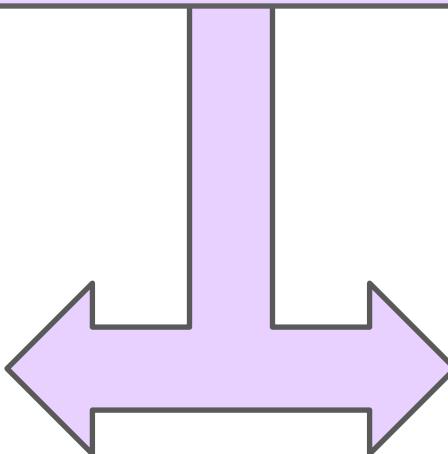
 initialize $v = 0$

 for each successor of state:

$p = \text{probability}(\text{successor})$

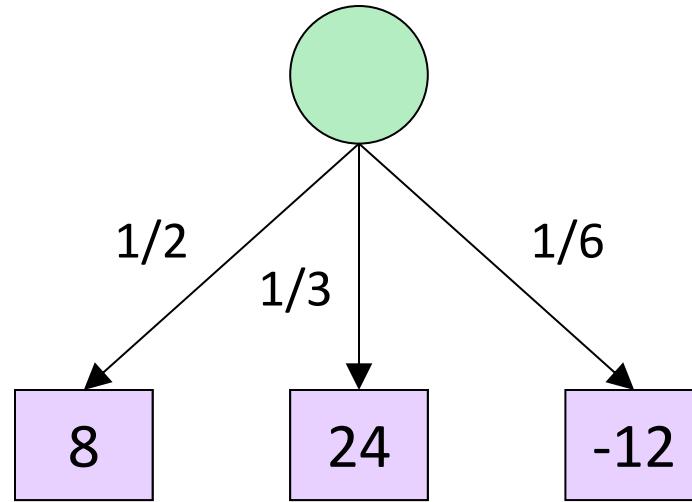
$v += p * \text{value}(\text{successor})$

 return v



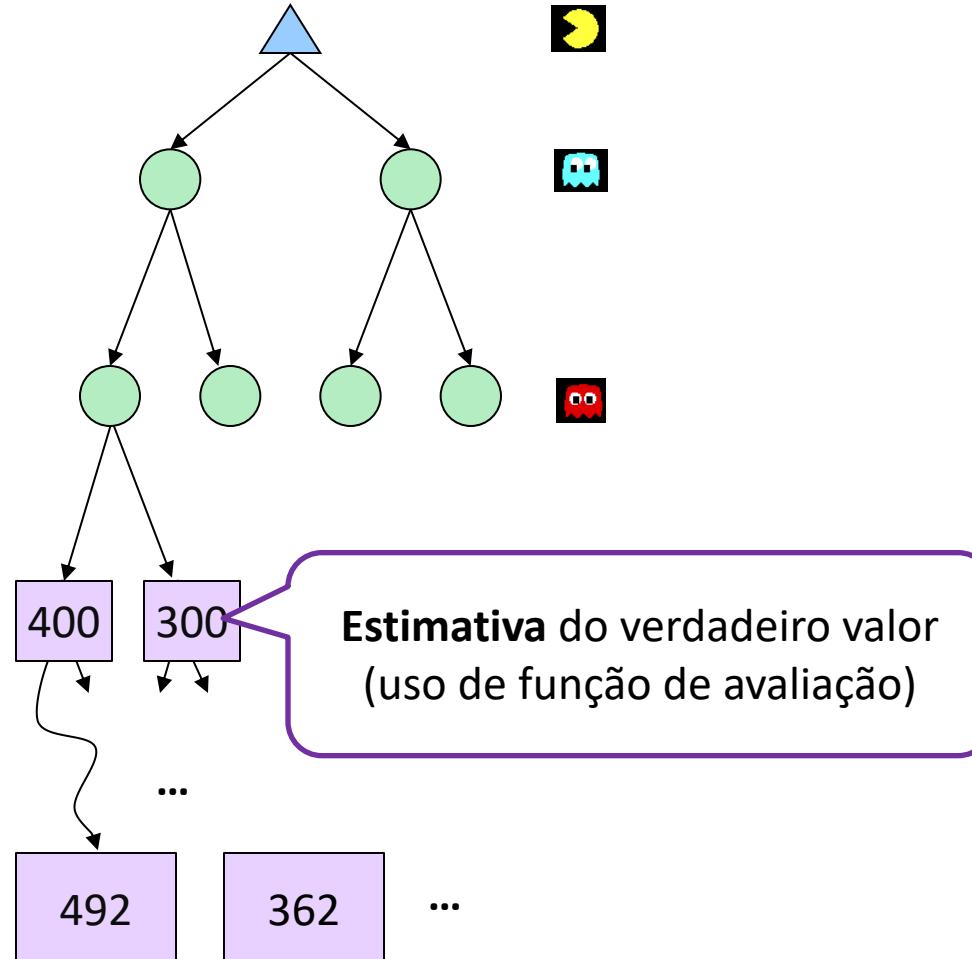
Expectimax

```
def exp-value(state):  
    initialize v = 0  
    for each successor of state:  
        p = probability(successor)  
        v += p * value(successor)  
    return v
```



$$v = (1/2)(8) + (1/3)(24) + (1/6)(-12) = 10$$

Limitando a profundidade no Expectimax



Expectimax-2048-AI

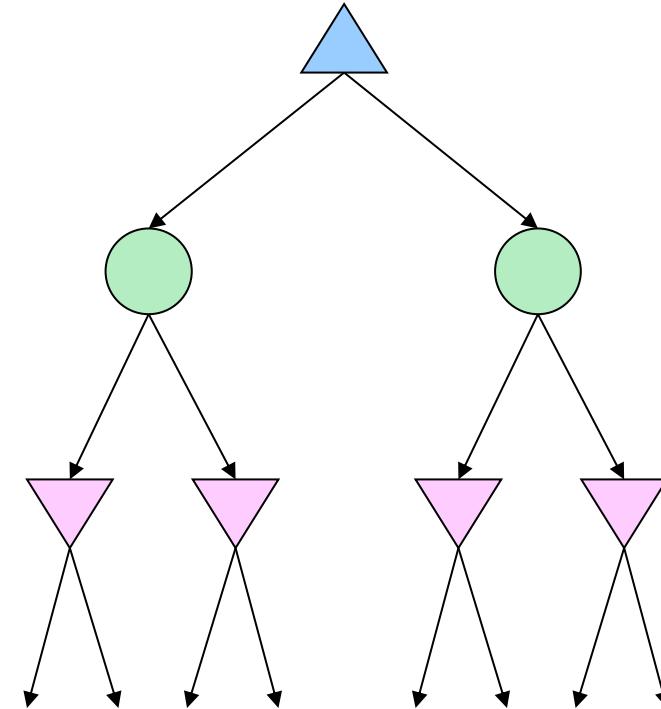
<https://github.com/andhofmt/Expectimax-2048-AI>



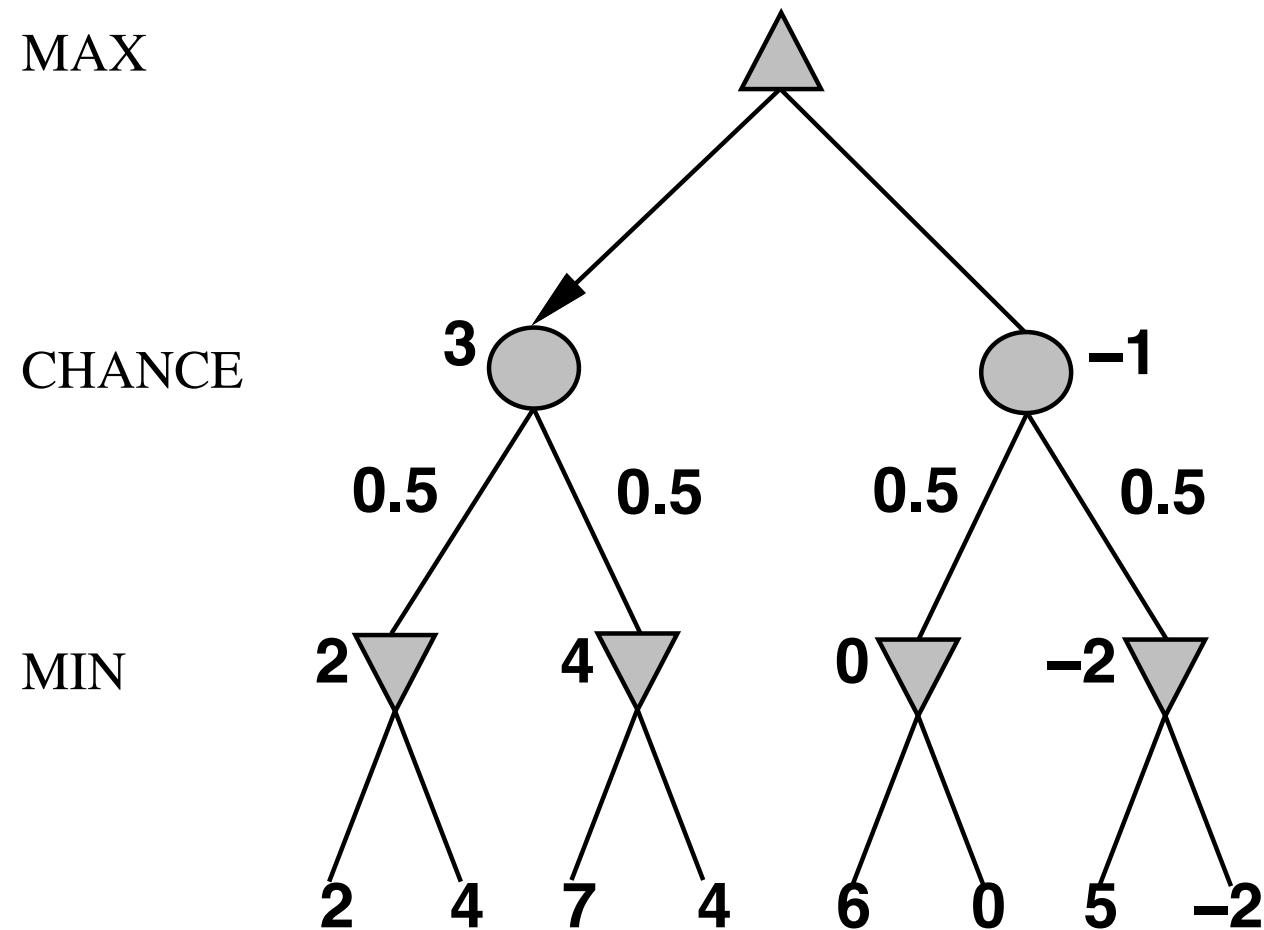
Agente consegue frequentemente vencer o jogo ao utilizar Expectimax e função de avaliação com heurísticas!

Expectiminimax

- Ambiente funciona como um jogador extra que se movimenta após cada min/max
- Cada nó computa a combinação apropriada de seus filhos



Expectiminimax



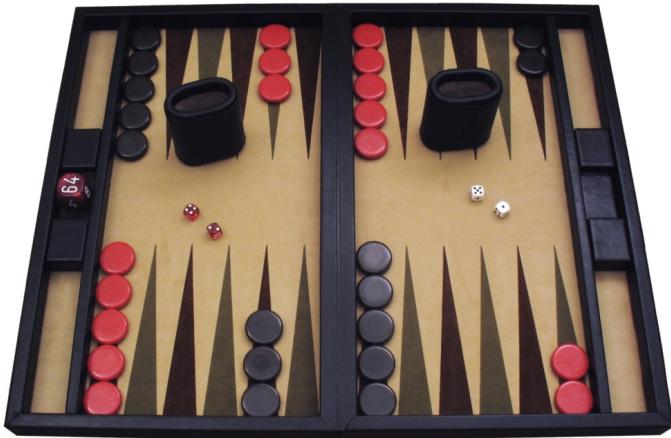
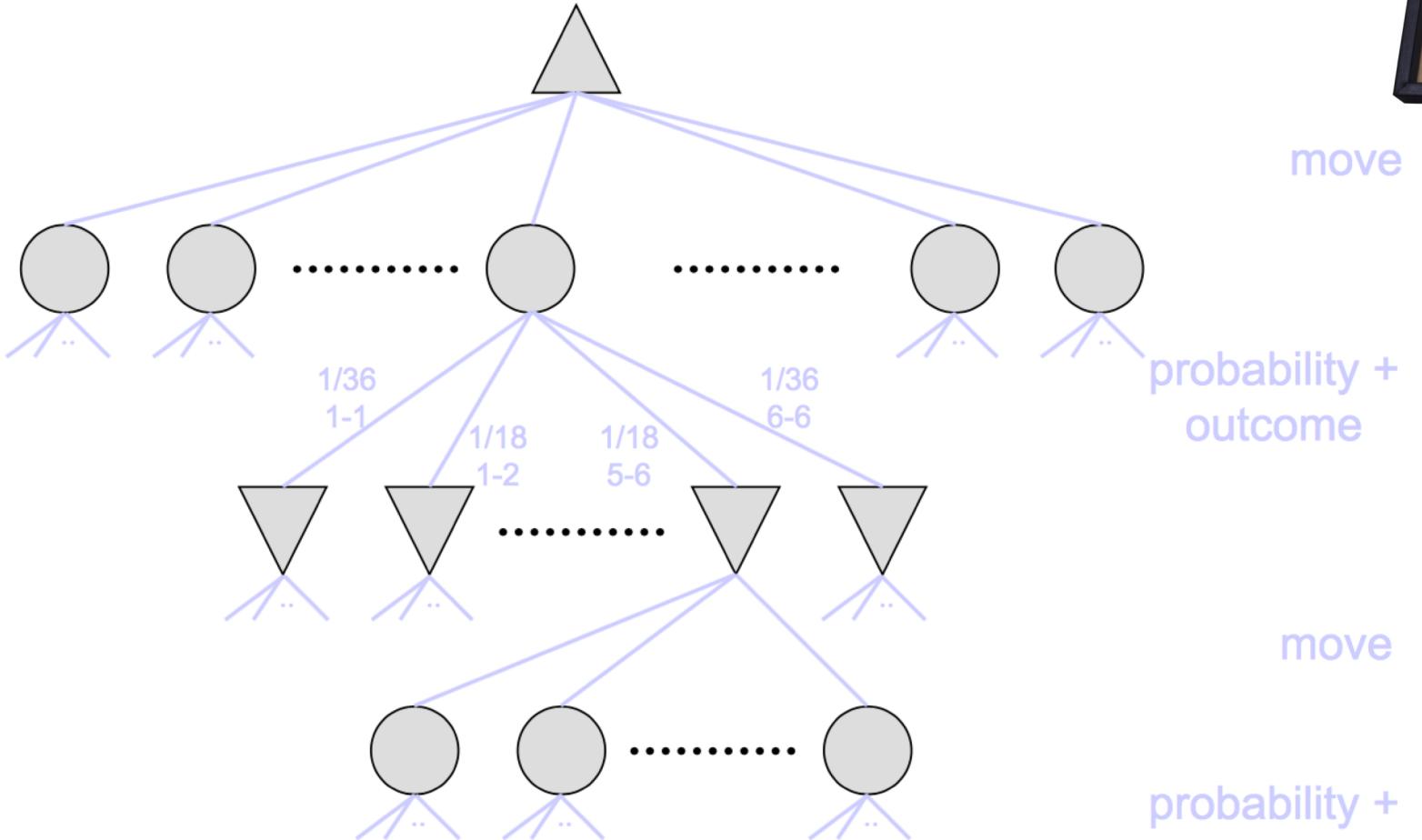
Gamão

MAX

CHANCE

MIN

CHANCE



move

probability +
outcome

move

probability +
outcome

Gamão

- time complexity: $\mathcal{O}(b^m n^m)$
 - b : maximal number of possible moves
 - n : number of possible outcomes for the random element
 - m : maximal search depth
- example: backgammon
 - average b is around 20
(but can be up to 4000 for doubles)
 - $n = 21$
 - about three ply depth is feasible



Para saber mais....

- Tesauro, G. (1995). Temporal difference learning and TD-Gammon. *Communications of the ACM*, 38(3), 58-68.-
 - profundidade 2
 - EVAL muito boa => nível campeão mundial



Para saber mais...

- Avila, M. V., & Macedo, H. T. (2012). Evolution-Assisted Multicriteria Decision Making for a No-Limit Texas Hold'em Poker Agent. *European Journal of Scientific Research*, 67(2), 272-282.
 - publicado com aluno de TCC;
 - abordagem híbrida.
- M. Bowling et al. Heads-up limit hold'em poker is solved. *Science* 347, 145-149. DOI: 10.1126/science.1259433 (2015).
 - Falo sobre isso em <http://www.saense.com.br/2016/03/o-jogador-perfeito-de-poker/>



Hendrik Macedo

Escreve sobre *Inteligência Artificial* no Saense.

<http://www.saense.com.br/autores/artigos-publicados-por-hendrik-macedo/>