

# Introdução à Inteligência Artificial

## PVP 5B – Procura Adversa MiniMax

José Coelho,  
2023



PVP 5B – Procura Adversa de José Coelho é disponibilizado sob a Licença *Creative Commons-Atribuição - NãoComercial-Compartilhaqual 4.0 Internacional*

# Índice

1. Algoritmo MiniMax
  1. Exemplo, 2 jogadores
  2. Exemplo, 3 jogadores
2. Algoritmo MiniMax com cortes Alfa / Beta
  1. Exemplo, 2 jogadores
  2. Exemplo, 2 jogadores ordenado
3. Procura limitada
4. Avaliação heurística

# Algoritmo MiniMax

- Recebe um estado e retorna uma ação e utilidade  
 $MiniMax: S \rightarrow A \times (P \rightarrow \mathbb{Z})$

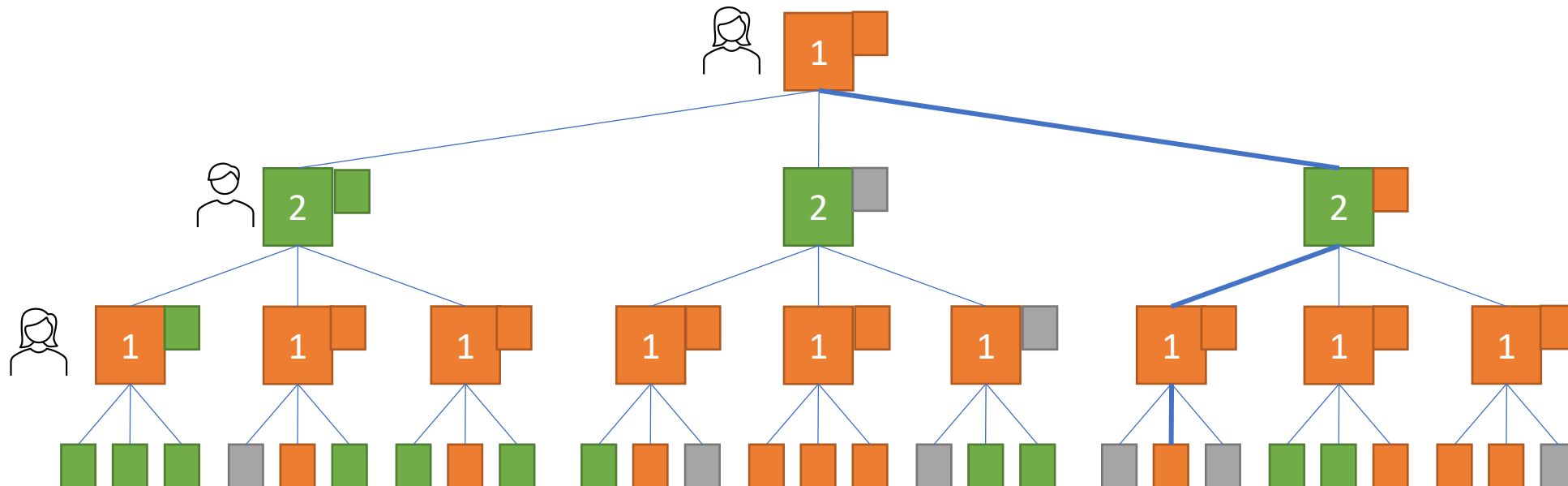
- $MiniMax(s)$ :
  1.  $teste(s) = 1$ , retornar  $(\_, util(s))$
  2. Caso contrário:  $\forall a \in A$ 
    1.  $(\_, f) \leftarrow MinMax(exe(s, a))$
    2. Se  $f_{jog(s)} > mf_{jog(s)}$  então  $mf \leftarrow f, ma \leftarrow a$
  3. Retornar  $(ma, mf)$

# Exemplo MiniMax, 2 jogadores

*MiniMax(s):*

1.  $teste(s) = 1$ , retornar  $(\_, util(s))$
2. Caso contrário:  $\forall a \in A$ 
  1.  $(\_, f) \leftarrow MinMax(exe(s, a))$
  2. Se  $f_{jog(s)} > mf_{jog(s)}$  então  $mf \leftarrow f, ma \leftarrow a$
3. Retornar  $(ma, mf)$

■ Vitória 1   
■ Vitória 2   
■ Empate



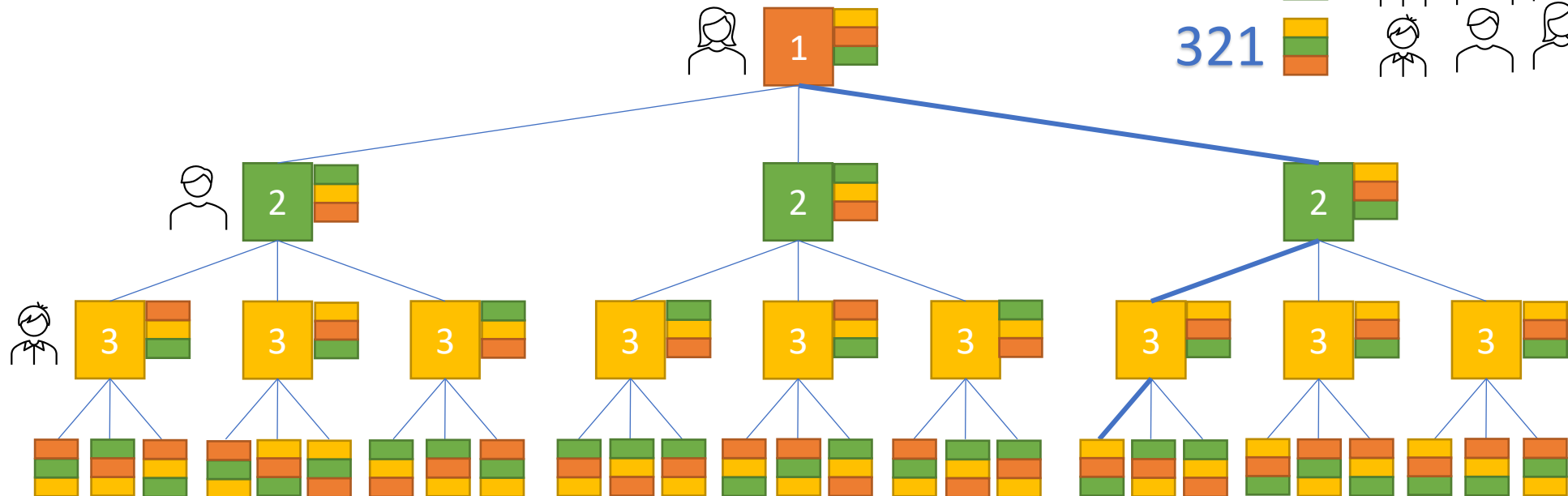
# Exemplo MiniMax, 3 jogadores

*MiniMax(s):*

1.  $teste(s) = 1$ , retornar  $(_, util(s))$
2. Caso contrário:  $\forall a \in A$ 
  1.  $(_, f) \leftarrow MinMax(exe(s, a))$
  2. Se  $f_{jog(s)} > mf_{jog(s)}$  então  $mf \leftarrow f, ma \leftarrow a$
3. Retornar  $(ma, mf)$

Vitória/Empate/Derrota:

123				
132				
231				
213				
312				
321				



# Algoritmo MiniMax, cortes alfa/beta

- Recebe um estado e retorna uma ação e utilidade

$$MiniMaxAB: S \times (P \rightarrow \mathbb{Z}) \rightarrow A \times (P \rightarrow \mathbb{Z})$$

- $MiniMaxAB(s, \alpha\beta)$ :

1.  $teste(s) = 1$ , retornar  $(\_, util(s))$
2. Caso contrário:  $\forall a \in A$ 
  1.  $(\_, f) \leftarrow MiniMaxAB(exe(s, a), ab)$
  2. Se  $f_{jog(s)} > mf_{jog(s)}$  então  $mf \leftarrow f, ma \leftarrow a$
  3. Se  $\alpha\beta_{jog(s)} < f_{jog(s)}$  então  $\alpha\beta_{jog(s)} \leftarrow f_{jog(s)}$
  4. Se  $\exists_{j \in P, j \neq jog(s)} \alpha\beta_j \geq f_j$  então CORTE!
3. Retornar  $(ma, mf)$

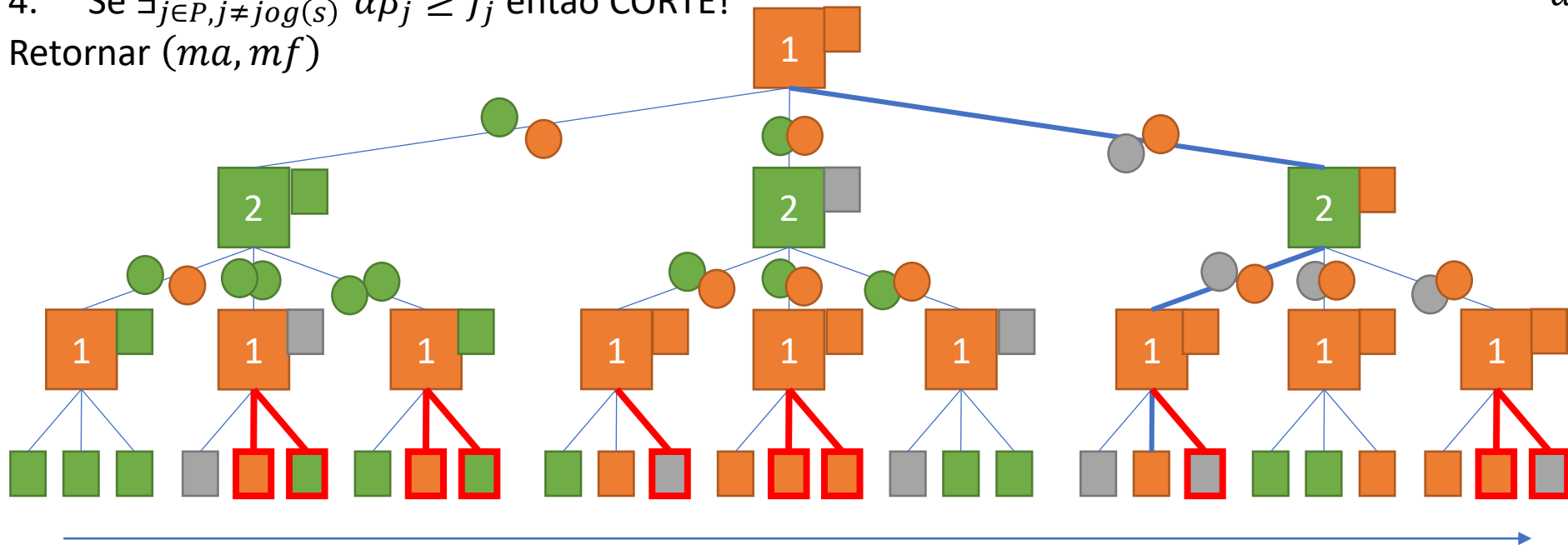
# Exemplo MiniMaxAB, 2 jogadores

*MiniMaxAB*( $s, \alpha\beta$ ):

1.  $teste(s) = 1$ , retornar  $(\_, util(s))$
2. Caso contrário:  $\forall a \in A$ 
  1.  $(\_, f) \leftarrow MinMaxAB(exe(s, a), ab)$
  2. Se  $f_{jog(s)} > mf_{jog(s)}$  então  $mf \leftarrow f, ma \leftarrow a$
  3. Se  $\alpha\beta_{jog(s)} < f_{jog(s)}$  então  $\alpha\beta_{jog(s)} \leftarrow f_{jog(s)}$
  4. Se  $\exists j \in P, j \neq jog(s) \ \alpha\beta_j \geq f_j$  então CORTE!
3. Retornar  $(ma, mf)$

■ Vitória 1  
■ Vitória 2  
■ Empate

$\alpha \bigcirc \bigcirc \beta$



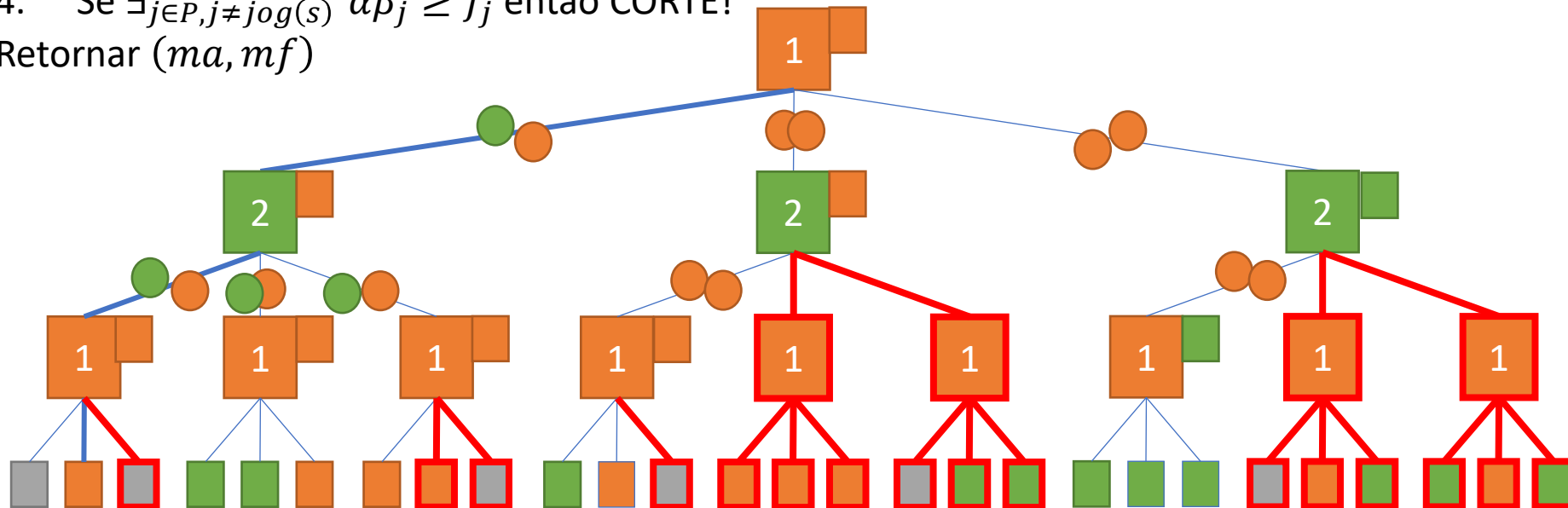
# Exemplo MiniMaxAB, 2 jogadores, ordenado

*MiniMaxAB*( $s, \alpha\beta$ ):

1.  $teste(s) = 1$ , retornar  $(\_, util(s))$
2. Caso contrário:  $\forall a \in A$ 
  1.  $(\_, f) \leftarrow MinMaxAB(exe(s, a), ab)$
  2. Se  $f_{jog(s)} > mf_{jog(s)}$  então  $mf \leftarrow f, ma \leftarrow a$
  3. Se  $\alpha\beta_{jog(s)} < f_{jog(s)}$  então  $\alpha\beta_{jog(s)} \leftarrow f_{jog(s)}$
  4. Se  $\exists_{j \in P, j \neq jog(s)} \alpha\beta_j \geq f_j$  então CORTE!
3. Retornar  $(ma, mf)$

■ Vitória 1  
■ Vitória 2  
■ Empate

$\alpha \bigcirc \bigcirc \beta$





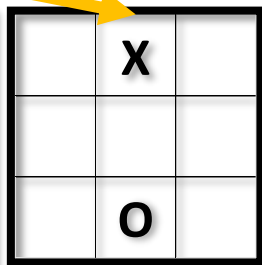
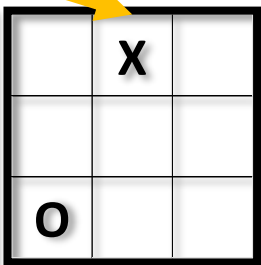
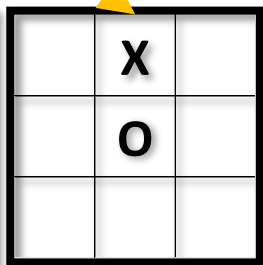
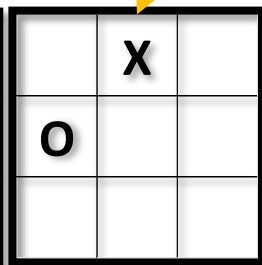
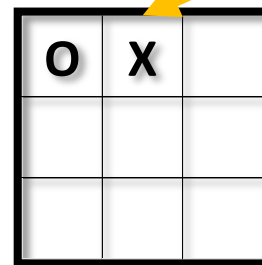
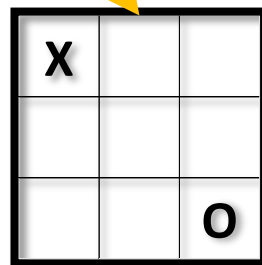
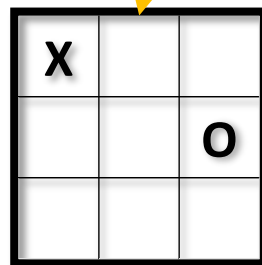
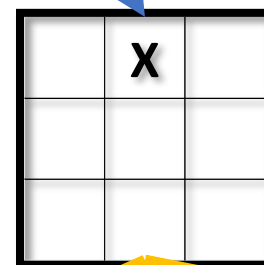
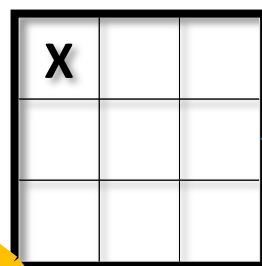
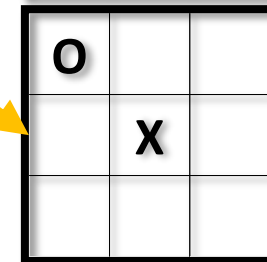
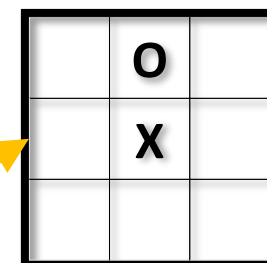
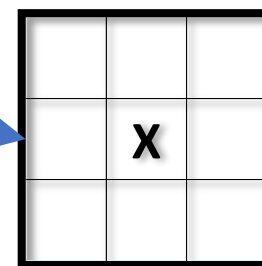
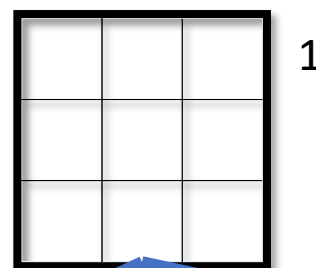
# Procura limitada

- Avaliação heurística
  - Distância ao objetivo mais próximo
  - Extrair as características relevantes, e fazer soma ponderada
  - Afinar com aprendizagem automática
- Paragem da procura
  - Não parar enquanto a heurística não estiver estável
  - Zona de estabilidade com movimentos restritos
    - Apenas ações que possam alterar grandemente a heurística
- Procura iterativa
  - Permitir utilizar o resultado da iteração anterior, para ordenar
  - Utilização de hashtable, para poupar recálculos entre iterações

# Avaliação heurística



- $10 \cdot \text{ameaças}(1) + \text{ameaças}(2)$



# Recursos utilizados

- Microsoft Power Point
- Clipchamp, voz de síntese Duarte
- Vimeo
- Russell, S. J. & Norvig, P. (2010). Artificial intelligence: A modern approach (3rd ed). Prentice Hall.