
Algoritmos para Jogos

— Introdução ao MCTS —

O que é um jogo?

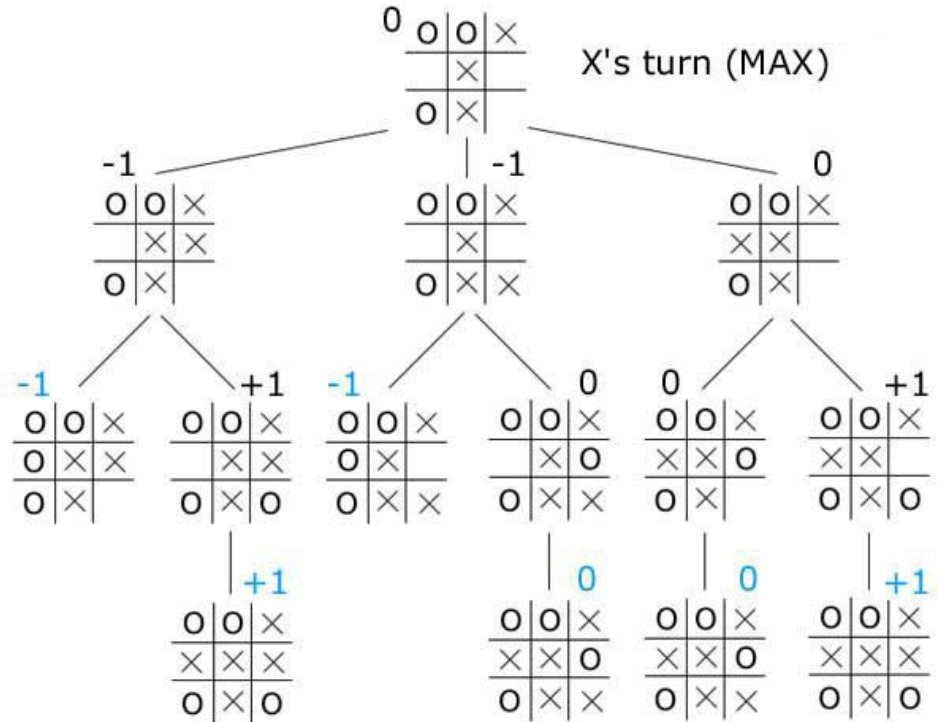
Formalmente podemos definir um jogo como um problema de procura com os seguintes elementos:

- S_0
- Jogador(**S**)
- Jogadas(**S**)
- Jogar(**S, A**)
- Terminal(**S**)
- Utilidade(**S, P**)

Algoritmos para jogos

Min-Max

- Procura em profundidade até encontrar um estado final
- O Jogador Max procura maximizar a sua jogada
- Precisamos de explorar a árvore toda para retornar uma resposta

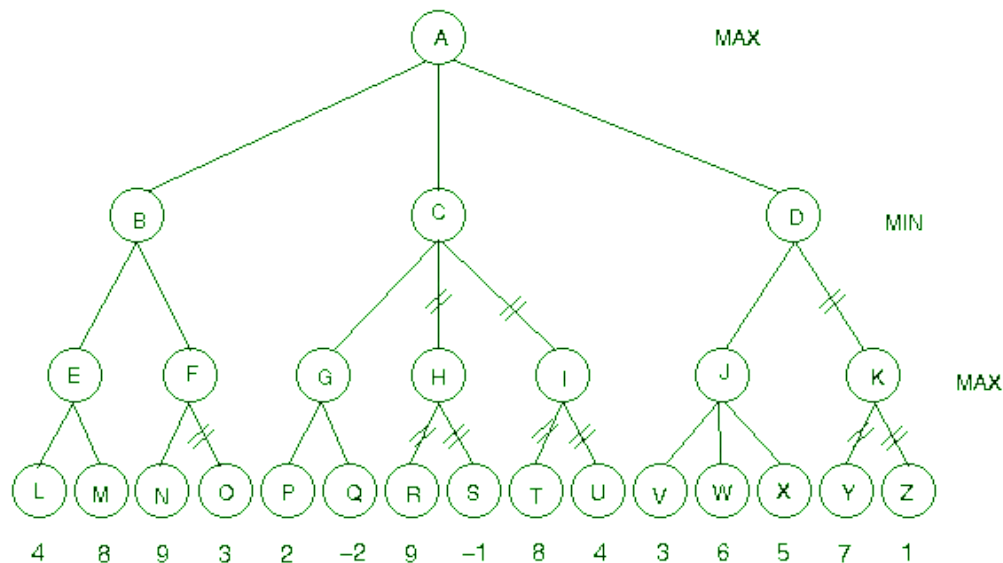


Complexidade do Min-Max

- Para uma profundidade d e um fator de ramificação b :
 - Complexidade temporal - $O(b^d)$
 - Complexidade espacial - $O(b \times d)$

Corte Alpha Beta

- Tenta diminuir os nós explorados
- Guarda o valor mínimo e máximo
- Não explora uma sub-árvore que não melhore o valor atual



Complexidade do Corte Alpha Beta

Para uma profundidade d e um fator de ramificação b :

- **Pior Caso:**
 - Complexidade temporal - $O(b^d)$
 - Complexidade espacial - $O(b \times d)$
- **Explorando sempre o melhor caminho:**
 - Complexidade temporal - $O(b^{d/2})$
- **Média :**
 - Complexidade temporal - $O(b^{3d/4})$

Monte Carlo Tree Search

O MCTS é um algoritmo que resolve o problema de pesquisa com adversários.

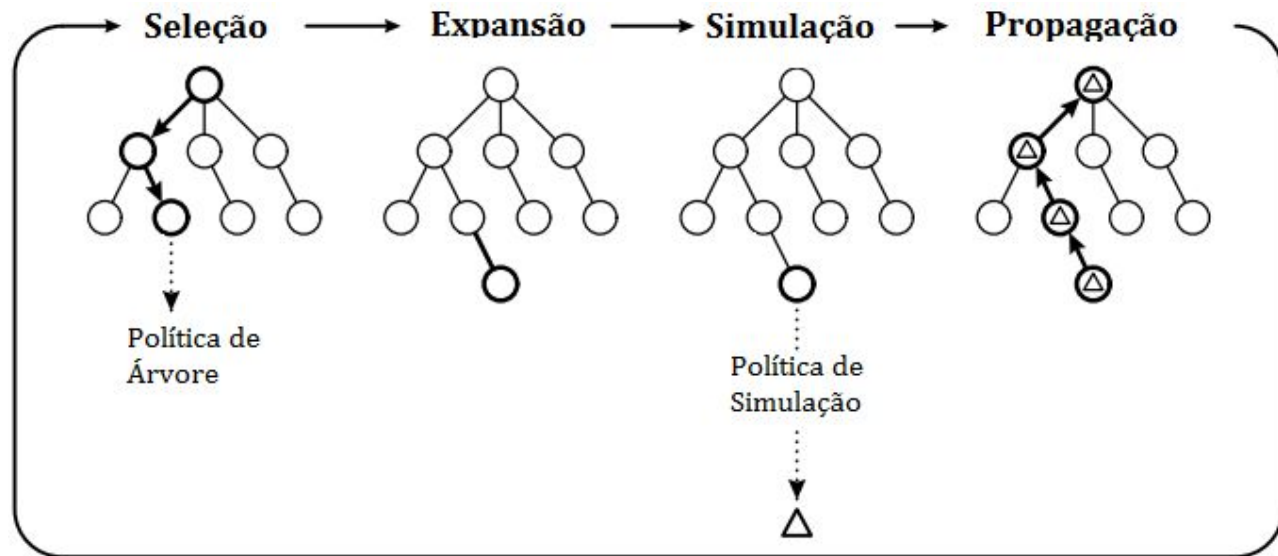
Este algoritmo pode ficar a correr indefinidamente até retornar uma resposta ótima, ou pode-se limitar o tempo de execução ou limitar as iterações do algoritmo, para garantir que retorna uma resposta.

A cada iteração, o algoritmo realiza quatro passos:

Monte Carlo Tree Search

Seleção - utilizando a política de árvore, procura um nó expansível.

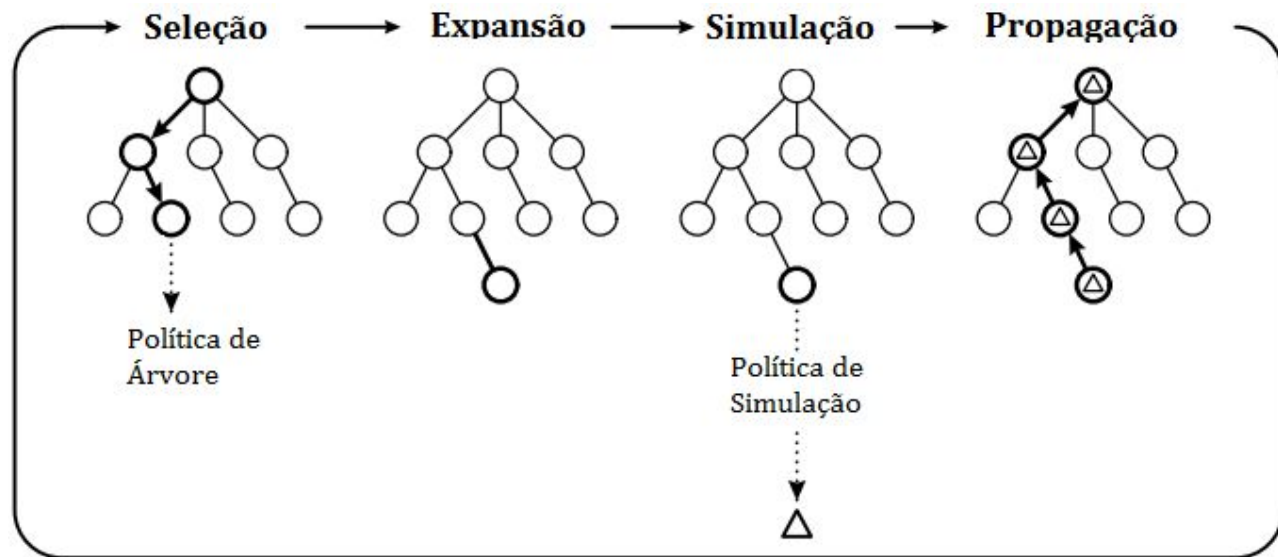
Expansão - são gerados um ou mais filhos, do nó selecionado no passo anterior, e adicionados à árvore.



Monte Carlo Tree Search

Simulação - utilizando a política de simulação, é encontrada uma utilidade para os nós adicionados.

Propagação - os resultados das simulações são propagados pela árvore



Monte Carlo Tree Search

Algoritmo 1: Algoritmo genérico MCTS

function MCTS(S_0)

 Cria no-raiz V_0 a partir do estado S_0

while *Dentro dos limites computacionais do*

$V_1 \leftarrow$ Política_de_Arvore(V_0)

$\Delta \leftarrow$ Política_de_Simulação($S(V_1)$)

 Propagar(Δ, V_1)

return Melhor_Filho(V_0)

end function

Políticas de árvore e simulação

Política de árvore - determina que nó vai ser escolhido e expandido, durante os passos de seleção e expansão.

Política de simulação - escolhe um caminho a partir de um nó não terminal até um nó terminal (simula um jogo)

As políticas mais simples são a escolha aleatória dos nós, contudo o MCTS apresenta melhores resultados com políticas ajustadas ao problema.

MCTS-UCT

O MCTS tenta aproximar o valor de cada nó ao seu valor real, para isso vai construindo uma árvore parcial. A árvore construída depende da maneira como os nós são selecionados.

O UCT (Upper Confidence Bound for Trees) é uma política de árvore que trata o problema da escolha de um nó, como um problema Multi-Armed Bandit.

O valor dado pelo UCT, para cada nó, corresponde a um valor aproximado da recompensa, calculado a partir das simulações.

MCTS-UCT

Assim sendo o UCT procura um nó j , que maximize: $UCT = \bar{X}_j + C_p \sqrt{\frac{2 \ln n}{n_j}}$

- n corresponde ao número de vezes que o nó atual foi visitado
- n_j corresponde ao número de vezes que o nó-filho j foi visitado
- \bar{X}_j é a recompensa média do nó-filho j
- C_p é uma constante

O princípio está em que se n_j tender para ∞ o seu valor vai tender para 0.

Dúvidas