

# JOGOS



prof. Luis Otavio Alvares

# JOGOS

- entre os primeiros domínios de aplicação
- razões
  - problema de definição fácil (regras do jogo)
  - constituem uma tarefa estruturada em que é fácil medir o sucesso ou fracasso



prof. Luis Otavio Alvares

## Vários tipos de jogos

- Jogos de 2 jogadores, com conhecimento completo: damas, xadrez
- Jogos com conhecimento incompleto: poker
- Jogos que incluem elemento de acaso (uso de dados por exemplo): gamão
- ....



prof. Luis Otavio Alvares

## Exemplo: xadrez

- fator médio de ramificação: 35
- número médio de jogadas: 50 para cada jogador.
- a árvore completa de um jogo terá aproximadamente  $35^{100}$  jogadas  $\sim 10^{154}$  nós
- portanto, uma busca cega é inviável, mesmo para realizar o primeiro movimento



prof. Luis Otavio Alvares

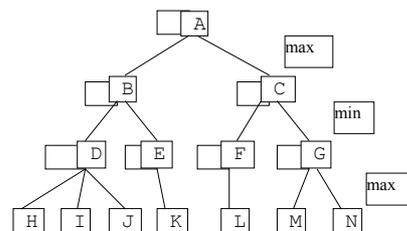
## Raciocínio em jogo de xadrez ?

- MINIMAX - Algoritmo mais usado em jogos de tabuleiro entre dois adversários



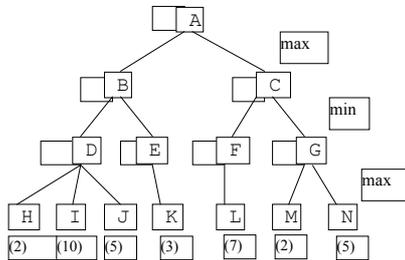
prof. Luis Otavio Alvares

## Uma árvore de jogo



prof. Luis Otavio Alvares

## Aplicando uma função de avaliação nos nós folhas

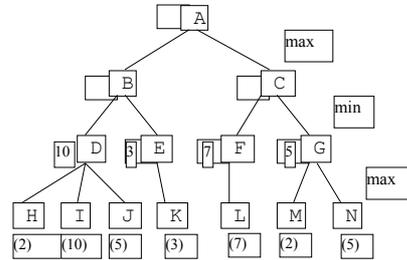


A função de avaliação é em relação ao jogador de maximização



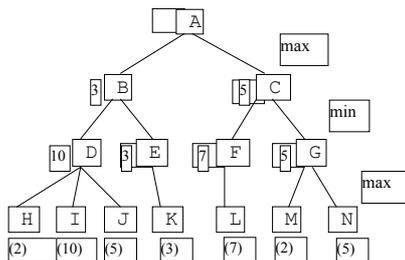
prof. Luis Otavio Alvares

## Propagando as avaliações



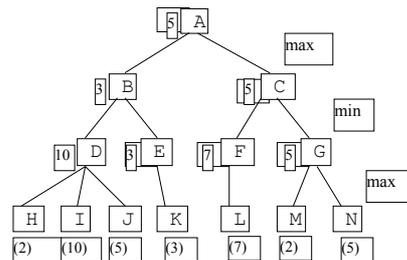
prof. Luis Otavio Alvares

## Propagando as avaliações



prof. Luis Otavio Alvares

## Propagando as avaliações



A jogada seria de "A" para "C"



prof. Luis Otavio Alvares

## Algoritmo Minimax

- GERMOV (posição, jogador) - gera todas as jogadas válidas a partir da situação do jogo definida por *posição* e considerando que a jogada atual é de *jogador*.
- ESTÁTICA (posição, jogador) - retorna um valor que quantifica o estado atual das peças do jogo.
  - A função de avaliação é em relação ao jogador que faria a jogada
  - Quanto maior o valor melhor a situação
- PROFUNDO\_SUFICIENTE (posição, profundidade) - pode considerar vários fatores:
  - número de níveis na árvore
  - um jogador ganhou
  - quão promissor é o caminho
  - quanto tempo ainda há disponível para a jogada



prof. Luis Otavio Alvares

```

MINIMAX (posição, profundidade, jogador)
1- se profundo_suficiente (posição, profundidade)
então retorna estrutura:
    valor ← estática (posição, jogador)
    caminho ← nil
senão SUCESSORES ← GERMOV (posição, jogador)
2- se SUCESSORES = ∅
então retorna estrutura:
    VALOR ← ESTÁTICA (posição, jogador)
    CAMINHO ← nil
senão MELHOR_CONTAGEM ← valor mínimo de ESTÁTICA (x,y)
para cada elemento SUC de SUCESSORES faça:
    RESULTADO_SUC ← MINIMAX (SUC, profundidade+1, OPOSTO (jogador))
    NOVO_VALOR ← - VALOR.RESULTADO_SUC
se NOVO_VALOR > MELHOR_CONTAGEM
então MELHOR_CONTAGEM ← NOVO_VALOR
    MELHOR_CAMINHO ← SUC + CAMINHO.RESULTADO_SUC
fim_se
fim_paracada
3- retorna estrutura:
    VALOR ← MELHOR_CONTAGEM
    CAMINHO ← MELHOR_CAMINHO
    
```



prof. Luis Otavio Alvares

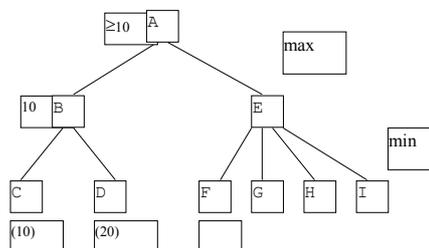
## Corte Alfa-Beta

Um aperfeiçoamento do algoritmo minimax corresponde a não pesquisar um ramo da árvore que comprovadamente não pode levar a um resultado melhor que o atual.



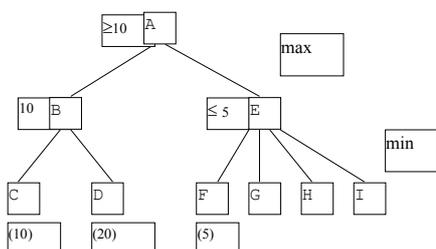
prof. Luis Otavio Alvares

## Corte Alfa-beta (cont.)



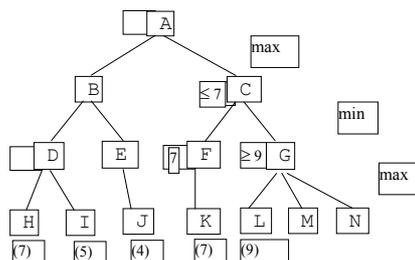
prof. Luis Otavio Alvares

## Corte Alfa-Beta (cont.)



prof. Luis Otavio Alvares

## Corte Alfa-Beta (cont.)



prof. Luis Otavio Alvares

## Refinamentos adicionais [RIC 94]

- **Esperando por quietude:** não parar a busca no meio de uma troca de peças, por exemplo
- **Busca secundária:** após decidir qual o melhor movimento, investigar este movimento 2 jogadas além do verificado inicialmente, para ter certeza que não há uma "armadilha"
- **Usar movimentos de livros:** por exemplo, aberturas e encerramentos
- **Não gerar todos os movimentos possíveis,** só os plausíveis



prof. Luis Otavio Alvares

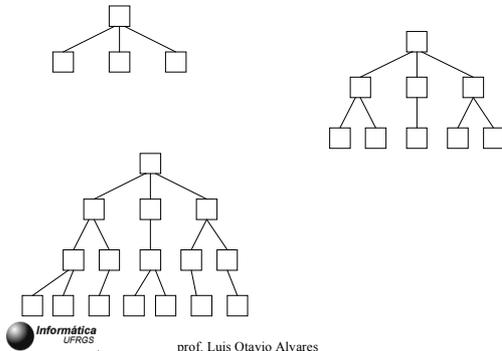
## Aprofundamento iterativo

- Procedimento usado para evitar o problema de jogar com tempo definido
- base: encontrar a melhor solução com profundidade 1, depois com profundidade 2 e assim sucessivamente. Ao terminar o tempo disponível, apresenta-se a melhor solução encontrada até o momento



prof. Luis Otavio Alvares

## Aprofundamento iterativo



prof. Luis Otavio Alvares

## Problemas do Minimax

- Baseia-se fortemente em que o oponente escolherá sempre o melhor movimento
  - aceitável em situação de vitória
  - em situação de derrota pode ser melhor arriscar que o oponente cometerá um erro



prof. Luis Otavio Alvares

## Alguns jogos específicos

Os melhores programas de jogos jogam, em geral, muito bem, mas eles utilizam técnicas bastante diferentes das técnicas humanas:

- Nós determinamos as melhores posições a analisar por um processo de identificação de estruturas (*pattern matching*), que é um processo realizado em paralelo
- A exploração de um conjunto de jogadas possíveis, como realizado pelo computador, não é paralelizável



prof. Luis Otavio Alvares

## XADREZ

- Não há uma predominância clara do computador ou do homem. Em 1997 pela primeira vez um campeão de xadrez perdeu para um computador
- Site:  
[www.chess.ibm.com/meet/html/d.3.html](http://www.chess.ibm.com/meet/html/d.3.html)



prof. Luis Otavio Alvares

## História do Deep Blue

- 1985 Hsu e Campbell na CMU (50.000 p/s - SUN)
- 1986 primeira competição (2v, 1e, 2d)
- 1988 Deep Thought (720.000 p/s - SUN4)
- 1989 ganha campeonato mundial de programas
- 1989 Hsu e Campbell entram na IBM
- 1989 joga com Kasparov
- 1991 Deep Thought II (7 milhões de pos/seg)
- conhecimento obtido do grande mestre Joel Benjamin
- vários anos e muitos milhões de dólares
- 256 processadores específicos
- analisa 200 milhões de posições por segundo



prof. Luis Otavio Alvares

## Deep Blue

- Para analisar 8 jogadas a frente (4 de cada jogador), seria preciso analisar

$35^8 \approx 2,2 \times 10^{12}$  jogadas, o que levaria cerca de **3 horas** por um programa que avaliasse 200 milhões de posições por segundo em busca cega



prof. Luis Otavio Alvares

## Deep Blue x IA

- **Does Deep Blue use artificial intelligence?**  
The short answer is "no." Earlier computer designs that tried to mimic human thinking weren't very good at it. No formula exists for intuition. So Deep Blue's designers have gone "back to the future". Deep Blue relies more on computational power and a simpler search and evaluation function.



prof. Luis Otavio Alvares

## Deep Blue

- "At the heart of Deep Blue's ability to play chess is its evaluation function. The evaluation function is an algorithm that measures the "goodness" of a given chess position. "
- "Deep Blue's strengths are the strengths of a machine. **It has more chess information** to work with than most computers and all but a few chess masters. "



prof. Luis Otavio Alvares

## Deep Blue

- "it operates much like a turbocharged "expert system," drawing on vast resources of stored information (For example, a database of opening games played by grandmasters over the last 100 years) and then calculating the most appropriate response to an opponent's move."



prof. Luis Otavio Alvares

## Deep Blue: Função de avaliação

Considera 4 fatores: peças, posição, segurança do Rei e perda de tempo

- **peças**: peão vale 1, bispo e cavalo 3, torre 5 e rainha 9. Mas possuir os dois bispos no fim do jogo vale mais que 6...
- **posição**: avalia a disposição das peças no tabuleiro, por exemplo, o número de casas seguras que elas podem ocupar
- **segurança do Rei**
- **perda de tempo**: é relacionado com o aspecto posição mas se preocupa com a disputa pelo controle do tabuleiro



prof. Luis Otavio Alvares

## DAMAS

- Predominância nas disputas: computador
- programa clássico: [Samuel 1963]
  - armazenava valores encontrados para as jogadas, para utilizar em jogos posteriores
  - função de avaliação:
$$c_1 t_1 + c_2 t_2 + \dots + c_{16} t_{16}$$
 $t_i$  é o valor de uma característica: domínio do centro, mobilidade, etc
  - os valores  $c_i$  mudam com a experiência



prof. Luis Otavio Alvares

## DAMAS

- Referência:
  - Programa: Chinook
  - Site: [www.cs.ualberta.ca/~chinook](http://www.cs.ualberta.ca/~chinook)



prof. Luis Otavio Alvares

## GO



prof. Luis Otavio Alvares

## GO

- Criado na China ~3000AC
- Citado por Confúncio 600AC
- É um dos poucos jogos em que a predominância é humana e esta diferença ainda é muito grande.



prof. Luis Otavio Alvares

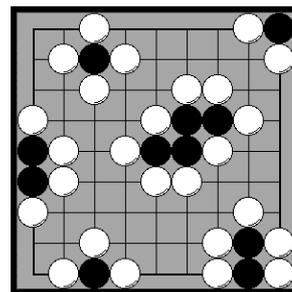
## GO

- É praticado com dois jogadores, em um tabuleiro 19 por 19 (interseções). O objetivo do jogo é conquistar o maior território possível
- Regras: cada jogador coloca uma peça por vez no tabuleiro; as peças cercadas por peças adversárias são retiradas do tabuleiro.
- Programa: Handtalk
- Site: [www.webwind.com/go](http://www.webwind.com/go)



prof. Luis Otavio Alvares

## GO



prof. Luis Otavio Alvares