

Inteligência Artificial

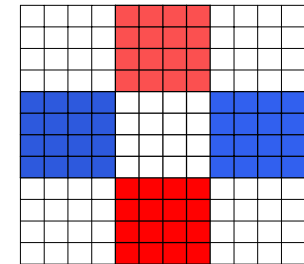
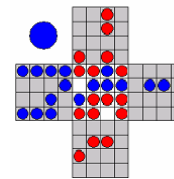
Exemplos de Jogos

Prof. Paulo Martins Engel

1

Xadrez chinês

- O objetivo de cada jogador é passar todas as suas peças para o quadrado oposto. Movimenta-se uma peça a cada jogada.
- O movimento é sempre para uma posição livre adjacente ou pode-se saltar quando há outra peça adjacente (sua ou do adversário) e uma posição livre logo após a peça. Este salto pode ser múltiplo (vários saltos seguidos).
- As peças de um jogador não podem entrar nos quadrados inicial e final do adversário.
- Os movimentos podem ser horizontal e/ou vertical; não podem ser em diagonal.



2

Xadrez chinês

Exemplo de função de avaliação:

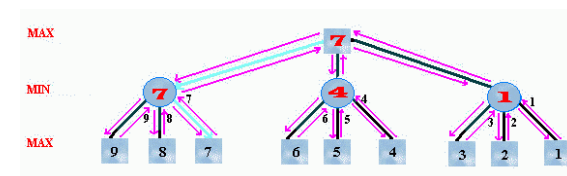
- o valor das peças é de acordo com a sua posição: 12 para a última linha (chegada), 11 para a anterior e assim sucessivamente;
- situações de bloqueio do adversário recebem uma contagem extra: +1, +2 ou +3, conforme a posição da barreira.

3

3

Minimax

- Largamente utilizado.
- Gera todos os movimentos possíveis.
- Ordem de complexidade w^d .
- Efeito horizonte.
- Assume que cada jogador fará o melhor lance.
- É uma estimativa.



4

Busca e Conhecimento

Busca

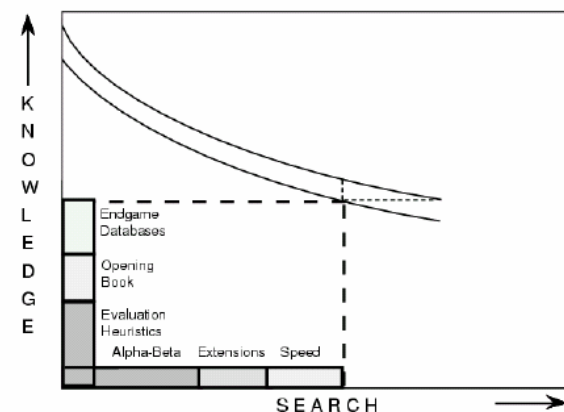
- Minimax
- Corte Alfa-Beta
- Ordenação
- Tabela de Transposição
- Velocidade do Código
- Forward Pruning
- Extensões de Busca
- Pensar na Vez do Oponente

Conhecimento

- Função de Avaliação
- Banco de Dados
- Livro de Abertura

5

Busca versus Conhecimento



6

Aprofundando a Busca

- Aumentar tempo de busca
- Aumentar a capacidade do hardware
- Otimizar o código do programa
- Pensar na vez do oponente
- Corte alfa-beta
- Ordenação da árvore
- Tabela de Transposição

7

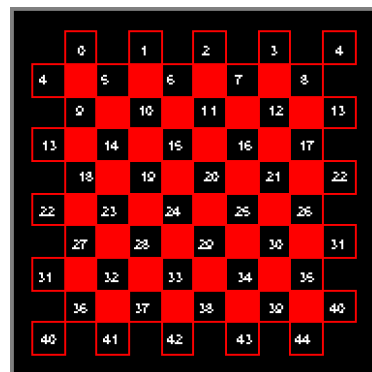
Tabela de Transposição

- Funciona como uma “cache” de posições já analisadas.
- Utilizada para ordenar a árvore.
- Reutiliza escores.
- Utilizada para fazer o computador pensar na vez do adversário.
- Redução de ~40% no tamanho da árvore.
- Desempenho aumenta conforme a profundidade da busca.

8

Bitboards

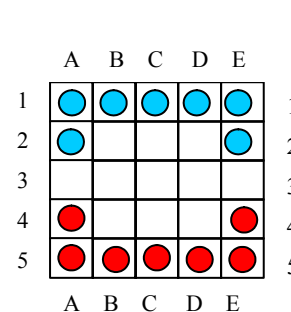
- Utilizado largamente.
- Permite operações lógicas entre tabuleiros.
- Funções para descobrir captura e mobilidade de peças são mais velozes.
- Pré-cálculo de tabelas de bitboards.



9

Exemplo do jogo “isola todos”

- **Descrição do jogo:**
- O jogo é jogado com sete peças para cada jogador, num tabuleiro 5 x 5.
- O objetivo é ser o primeiro a ir da posição inicial, na qual todas as peças podem se mover, até uma posição na qual nenhuma peça da sua cor possa ser movida. Não há captura de peças.



- **Mobilidade**
- Uma peça só pode ser movimentada quando ela estiver em contato, horizontalmente, verticalmente, ou diagonalmente, com no mínimo uma outra peça da mesma cor.
- Na posição inicial, todas as peças podem ser movimentadas (ver figura).
- Quando uma peça puder se movimentar, ela pode andar numa linha reta qualquer número de quadrados sem pular sobre outras peças ou mudar de direção.

10

Características do Isomaster

- Linguagem C
- Tabuleiro representado por bitboard (32 bits)
- Operadores booleanos (and, or, xor, shift, negate) utilizados

Azul: 1 1 1 1 1
1 0 0 0 1
0 0 0 0 0
0 0 0 0 0
0 0 0 0 0

Azul: 00000001111110001000000000000000

11

Bitboards

Azul: 1 1 1 1 1 1 0 0 0 0 1
0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0

Vermelho: 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0
1 0 0 0 0 0 0 0 0 0 0
1 1 1 1 1 1 1 1 1 1 1

Em notação binária com 32 bits (e em notação hexadecimal):

Azul: 00000001111110001000000000000000 (0x01F88000)
Vermelho: 0000000000000000000000001000111111 (0x0000023F)

12

Representação das posições

- Define-se cada posição por uma constante:

```
#define A1 0x01000000: 0000 0001 0000 0000 0000 0000 0000 0000
```

...

```
#define E5 0x00000001 : 0000 0000 0000 0000 0000 0000 0000 0001
```

	A	B	C	D	E
1	1	0	0	0	0
2	0	0	0	0	0
3	0	0	0	0	0
4	0	0	0	0	0
5	0	0	0	0	0

A1

	A	B	C	D	E
1	0	0	0	0	0
2	0	0	0	0	0
3	0	0	0	0	0
4	0	0	0	0	0
5	0	0	0	0	1

E5

13

Representação de configurações

- Inicial Azul: A1 OR B1 OR C1 OR D1 OR E1 OR A2 OR E2
- Lance: A1 – C3 : Lance A1 OR C3
- Executar o lance: Azul = Azul XOR Lance

	A	B	C	D	E
1	1	1	1	1	1
2	1	0	0	0	1
3	0	0	0	0	0
4	0	0	0	0	0
5	0	0	0	0	0

AZUL

	A	B	C	D	E
1	1	0	0	0	0
2	0	0	0	0	0
3	0	1	0	0	0
4	0	0	0	0	0
5	0	0	0	0	0

A1 OR C3

	A	B	C	D	E
1	0	1	1	1	1
2	1	0	0	0	1
3	0	1	0	0	0
4	0	0	0	0	0
5	0	0	0	0	0

A1 XOR Lance

14

Função de Avaliação

- **3 aspectos básicos**
 - 1. Número de peças isoladas
 - 2. Número de peças solitárias
 - 3. Zonas do Tabuleiro

Azul: 4 3 4 3 4
3 1 2 1 3
5 3 6 3 5
4 2 3 2 4
5 4 5 4 5

15

Banco de Dados de Fim de Jogo

Características

- Contém informação perfeita.
- Pode ser atingido mesmo em posições da abertura.
- Indispensável para a construção de um jogo de damas.
- Construído com o algoritmo de análise retroativa.

Vantagens

- Proporciona cortes na busca.
- Melhora a qualidade da estimativa.
- Garante vitórias ou empates em finais de partidas.
- Mestres humanos memorizam finais de até 5 peças.
- Eventualmente será responsável pela resolução do jogo de damas.

16

Banco de Dados - Problemas

- Tempo de cálculo.
- Espaço de armazenamento.
- “Swaps” entre disco e memória.
- Como garantir o progresso ?
- Como diferenciar dois empates ? E duas derrotas ?

Tamanho do Banco(peças)	Número de Posições	Tamanho em bytes	Tempo de Cálculo (Pentium Pro 200)
2	3.488	872	3 s
3	199.520	48 Kb	5 m
4	6.408.836	1,6 Mb	3h e 20m
5*	134.000.000	33 Mb	3 dias
6*	2.144.000.000	536 Mb	48 dias
7*	30.016.000.000	7,5 Gb	22 meses

17

Módulos

- Gerador de Movimentos
- Minimax
- Função de Avaliação
- Tabela de Transposição
- Banco de Dados de Fim de Jogo

Módulo	Tempo Utilizado	Linhas de Código
Gerador de Movimentos	22 %	500
Minimax	21 %	300
Função de Avaliação	3 %	100
Tabela de Transposição	20 %	300
Banco de Dados	2 %	700
Outras Funções	32 %	3000

18

Técnicas Utilizadas

- Bitboards.
- Corte alfa-beta.
- Tabela de transposição.
- Banco de dados com posições de 4 ou menos peças.
- Pensa na vez do oponente.
- Detecção de Repetições.
- ETC (Enhanced Transposition Cutoff)

19

Busca - Resultados Práticos

- +2 ply : Corte alfa-beta.
- +1ply : Tabela de transposição.
- +1ply : Pensar na vez do adversário.

Redução do fator de ramificação de ~5 para ~2,3.

- +1 ply ao dobrar o número de posições analisadas.
- +5 ply, aumento de ~64 vezes.
- +10 ply, aumento de ~4000 vezes.

20

Banco de dados

- Ponto forte do jogo.
- 6408836 posições armazenadas.
- 2 bits por entrada.
- Começa a ser utilizado efetivamente em posições com 10 ou menos pedras.
- Proporciona excelente desempenho para finais de 6 ou menos pedras.

21

Avaliação

- Fases do jogo
 - Fraco desempenho na abertura.
 - Bom meio de jogo.
 - Excelente fim de jogo.
- 95% de acerto para posições propostas.
- Bom desempenho contra outros programas.
- Ganhos materiais X Ganhos posicionais
- Controle do tabuleiro.
- A “forçada”.

Programa	Livro de Abertura	Tabela de Transposição	Banco de Fim de Jogo	Vitórias do Programa	Empates	Vitórias do Panzer Checkers
WinBraz 3D	Sim	Sim	Não	6	2	-
Plus 800	Não	Sim	Sim	-	-	8
DamasMaster	Não	Não	Não	-	1	7
IaDamas	Não	Não	Não	-	-	8
DamaX	Não	Não	Não	-	3	5

22