
Universidade Federal Fluminense – UFF
Instituto de Computação
Departamento de Ciência da Computação

Jogo de Damas usando Minimax com
profundidade limitada e avaliação heurística e
Minimax com poda alfa beta
Primeiro projeto de Inteligência Artificial

Alunos: Bruno Cezar Paiva Dos Santos
Marcos Vinicius Moura Dos Reis
Rafael Duarte Campbell de Medeiros
Raphael Guizan Goncalves Baptista

Professora: Aline Marins Paes Carvalho

Niterói – RJ
Maio / 2019

Sumário

1	Apresentando o Software	2
1.1	Interface visual do jogo e seu funcionamento	2
1.2	Regras do jogo	2
1.2.1	Peças e Movimentações	2
1.2.2	Definições	3
1.2.3	Promoção a Dama	3
1.2.4	Captura	3
1.2.5	Condição de Empate	4
1.3	Propriedades de ambiente	4
2	Algoritmos de Busca	4
2.1	Minimax com poda alfa-beta	4
2.2	Minimax com avaliação heurística e profundidade limitada	5
3	Exemplo de execução	5
4	Repositório	5

1 Apresentando o Software

O trabalho apresentado seguirá os parâmetros descritos para a **primeira opção**, descrevendo um jogo com dois algoritmos de busca diferentes. Para o trabalho, o jogo a ser apresentado será o jogo de **damas** brasileiro, com os algoritmos de **Minimax com poda alfa & beta** e **Minimax com avaliação heurística e profundidade limitada**.

1.1 Interface visual do jogo e seu funcionamento

A interface visual será bastante simplificada, baseada exclusivamente no teclado e tela. O jogador poderá ver uma matriz que representa o campo em questão e cada célula será populada com um indicador:

- **b** - peça simples preta
- **B** - peça preta dupla
- **w** - peça branca simples
- **W** - peça branca dupla
- **.** - espaço vazio preto
- **-** - espaço vazio branco

As células serão definidas por um par constituído de um número (**L**) indicando a linha, variando de 0 a 7, e outro número (**N**) indicando a coluna, variando de 0 a 7. Vale lembrar que as linhas são contadas de baixo para cima e as colunas, da esquerda para a direita. Os pares são dados na forma (L,N), i.e. coluna e linha (e.g. [1,1] ou [1, 7]).

Modelo

<Coluna da posição atual> <Linha da posição atual>
<Coluna da posição final> <Linha da posição final>

Exemplo

>> 1 1
>> 5 3

Caso o usuário submeta como posição inicial uma célula não ocupada por uma de suas peças, o *software* o notificará até que receba uma posição inicial válida; analogamente, o *software* também o notificará caso a posição final passada não coincida com um movimento válido para a peça em questão.

1.2 Regras do jogo

1.2.1 Peças e Movimentações

O jogo de Damas tem dois participantes que usam um tabuleiro para jogar. Com o tabuleiro deve ser posicionado de modo que a grande diagonal comece do lado esquerdo de cada jogador, a primeira casa à esquerda de cada jogador será preta. O jogador que estiver jogando com as peças brancas começa o jogo, podendo dar o primeiro lance.

Os jogadores alternam a vez até o fim do jogo. As peças comuns só podem se movimentar para a frente, para uma casa preta livre na próxima linha, diagonal à sua casa atual. As damas podem se movimentar em diagonal para frente e para trás para qualquer casa livre, desde que o caminho esteja livre. O jogo termina quando todas as peças de um jogador forem capturadas ou quando este não puder mais fazer nenhum lance válido.

- Peças - 24 peças, 12 brancas e 12 pretas.
- Tabuleiro - tabuleiro de 64 casas, claras e escuras.
- Distribuição - 12 peças da mesma cor para cada jogador, posicionadas nas casas escuras, ocupando as três linhas mais próximas de cada jogador.
- Objetivo - Capturar todas as peças do oponente ou deixá-lo impossibilitado de mover.

1.2.2 Definições

- Dama - peça simples promovida a dama
- Grande diagonal - a maior linha formada diagonalmente pelas casas escuras.
- Lance - o deslocamento de uma peça para outra casa.
- Captura - um lance onde uma peça passa por cima de uma peça adversária que está entre a sua casa e a sua casa de destino.
- Tomada em cadeia - uma lance onde uma peça captura duas ou mais peças sucessivamente.

1.2.3 Promoção a Dama

Quando uma peça comum chega na ultima linha do tabuleiro (primeira linha do adversário), ela será promovida a dama, ganhando mais liberdade de movimento.

1.2.4 Captura

A captura das peças é obrigatória, ou seja, se na vez de um jogador ele puder capturar uma peça adversária, ele deve fazê-lo. As peças comuns só podem capturar para frente e para trás peças adversárias adjacentes às suas. As damas podem capturar peças distantes na sua diagonal, desde que o caminho entre a dama e a peça capturada esteja livre, e o caminho entre a peça capturada e a casa de destino da dama esteja também livre.

Quando uma captura é realizada, a peça capturada é removida do tabuleiro. Quando uma peça faz uma captura e fica numa posição onde é possível fazer outra captura, ela deve fazê-la na mesma jogada, realizando uma tomada em cadeia. A peça deve fazer isso até que não seja mais possível capturar peça alguma neste lance.

1.2.5 Condição de Empate

- A partir de qualquer ponto da partida se ocorrerem 20 (vinte) lances sucessivos de Damas, sem captura ou movimento de peças comuns.
- Se uma mesma posição se produzir pela terceira vez, cabendo ao mesmo jogador o lance será considerado um empate.
- Se um jogador possuir três damas contra uma dama do outro jogador, será considerado empate se o jogador com o maior numero de peças não conseguir obter vitória em vinte lances.

1.3 Propriedades de ambiente

O jogo de Damas, escolhido para o trabalho, tem as seguintes características:

- Completamente observável;
- Determinístico;
- Episódico;
- Discreto;
- Estático;
- Multiagente.

2 Algoritmos de Busca

Os algoritmos utilizados no desenvolvimento do jogo são o **Minimax com poda alfa-beta** e o **Minimax com avaliação heurística e profundidade limitada**.

O algoritmo Minimax calcula a decisão minimax a partir do estado corrente. Ela utiliza uma computação recursiva simples dos valores minimax de cada estado sucessor. Os valores são calculados utilizando uma função utilidade que retorna um valor numérico para os estados terminais.

2.1 Minimax com poda alfa-beta

A poda alfa-beta é um artifício para calcular a decisão minimax correta sem examinar todos os nós na árvore do jogo, tornando o minimax mais eficiente.

Para o nosso jogo de Damas, o valor numérico que a função utilidade retorna foi calculado da seguinte forma(para uma IA jogando com as brancas):

$$((whitedoubles - blackdoubles) * 3) + (capturedBlacks - capturedWhites)$$

whitedoubles = quantidade de damas brancas.

blackdoubles = quantidade de damas pretas.

capturedWhites = quantidade de peças brancas capturadas.

capturedblacks = quantidade de peças pretas capturadas.

Com a poda alfa-beta, o algoritmo consegue garantir um resultado igual ao Minimax convencional, mas diminuindo significativamente a quantidade de nós que serão visitados.

Como o jogo de Damas tem espaços de busca que entram em loop e implicam em empate, tornou-se mais prático colocar um limite de profundidade similar a de um empate, como se fosse um fim de jogo.

2.2 Minimax com avaliação heurística e profundidade limitada

Este algoritmo é semelhante ao anterior, mas tem sua profundidade limitada, não indo até os nós terminados sempre. Também não realiza poda e, no lugar da função de utilidade, usa uma função heurística mais completa.

No desenvolvimento do jogo, o limite da profundidade é definido na instância da classe e fica salvo na variável `DEPTH_LIMIT`. Deste modo, a recursão sempre conta a profundidade e verifica, logo no início, se está em uma profundidade dentro do limite; caso não esteja, retorna a heurística da posição.

A heurística é usada para o movimento e a posição final da jogada, levando em conta tanto seu desempenho (no quesito de capturar e virar dama) quanto de sua situação futura (pode capturar ou ser capturada no futuro). O valor numérico é dado pela expressão:

$$(2 + canCapture - canBeCaptured) + (captured * 3) + (becomeDouble * 9)$$

`canCapture` = quantidade de peças que o jogador pode capturar

`canBeCaptured` = quantidade de peças do oponente que podem capturar o jogador.

`willCapture` = quantidade de peças que o jogador irá capturar

`becomeDouble` = recebe 1 caso a peça vire dama, ou 0 caso contrário.

3 Exemplo de execução

O jogo tem uma classe *log* que gera arquivos guardando informações relevantes sobre os jogos e as decisões tomadas em cada rodada.

Para cada jogador e cada jogo, é gerado um arquivo que recebe a data, horário, cor da peça e jogador (IAHeurística, IAPoda ou Jogador); esse arquivo guarda o tempo para tomar a decisão, número de nós visitados, número de nós podados e qual o movimento realizado (e suas implicações, como captura ou virar dama). Ao fim, o *log* indica vitória, derrota ou empate.

Como o *software* permite realizar vários jogos em sequência, é gerado um *log* que guarda quem são os jogadores para cada cor e, para cada jogo, o tempo total de execução e quem foi vitorioso (ou empate).

Para este trabalho, foram realizados algumas jogadas e os arquivos de *log* estão encaminhados anexos aos relatórios.

4 Repositório

O jogo completo e alguns exemplos de *log* para as combinações podem ser encontrados no seguinte repositório: <https://github.com/rafaeldcampbell/CheckersWithAI>