

Inteligência Artificial

Primeiro Trabalho Prático

Participantes:

Henrique Sousa, up201906681

Mateus Silva, up201906232

Melissa Silva, up201905076

Professora:

Ana Paula Rocha

Especificação do Trabalho

- A presente apresentação demonstra o progresso já feito no primeiro trabalho prático da unidade curricular *Inteligência Artificial*;
- O trabalho poderia debruçar-se sobre um de três temas: **Métodos de Pesquisa Heurísticos para Jogos de 1 Jogador**, **Métodos de Pesquisa Adversarial para Jogos de Tabuleiro de 2 Jogadores** e **Metaheurísticas para Problemas de Optimização/Decisão**;
- Optámos pelo segundo tema, dentro do qual escolhemos o jogo 2F, *Lines of Action*.

Formulação do Problema como Problema de Pesquisa

- O jogo *Lines of Action* pode representar-se com uma lista 2D com as dimensões do tabuleiro. O estado inicial, S_i , encontra-se ao lado, onde: 0 é uma posição vazia, 1 é uma peça preta e 2 é uma peça branca;
- Estado Final: todas as peças de um jogador ficam ligadas ou um jogador fica só com uma peça.

```
[0, 1, 1, 1, 1, 1, 1, 0]
[2, 0, 0, 0, 0, 0, 0, 2]
[2, 0, 0, 0, 0, 0, 0, 2]
[2, 0, 0, 0, 0, 0, 0, 2]
[2, 0, 0, 0, 0, 0, 0, 2]
[2, 0, 0, 0, 0, 0, 0, 2]
[2, 0, 0, 0, 0, 0, 0, 2]
[0, 1, 1, 1, 1, 1, 1, 0]
```

Operador	Précondições	Efeitos	Custo
$mover(P, esquerda)$	$P_x > 0$	$P_x = P_x - N$	1
$mover(P, direita)$	$P_x < 7$	$P_x = P_x + N$	1
$mover(P, cima)$	$P_y > 0$	$P_y = P_y - N$	1
$mover(P, baixo)$	$P_y < 7$	$P_y = P_y + N$	1
$mover(P, diagonalPEsquerda)$	$P_x > 0 \wedge P_y > 0$	$P_x = P_x - N \wedge P_y = P_y - N$	1
$mover(P, diagonalPDireita)$	$P_x < 7 \wedge P_y < 7$	$P_x = P_x + N \wedge P_y = P_y + N$	1
$mover(P, diagonalSEsquerda)$	$P_x > 0 \wedge P_y < 7$	$P_x = P_x - N \wedge P_y = P_y + N$	1
$mover(P, diagonalSDireita)$	$P_y < 7 \wedge P_x < 7$	$P_x = P_x + N \wedge P_y = P_y - N$	1

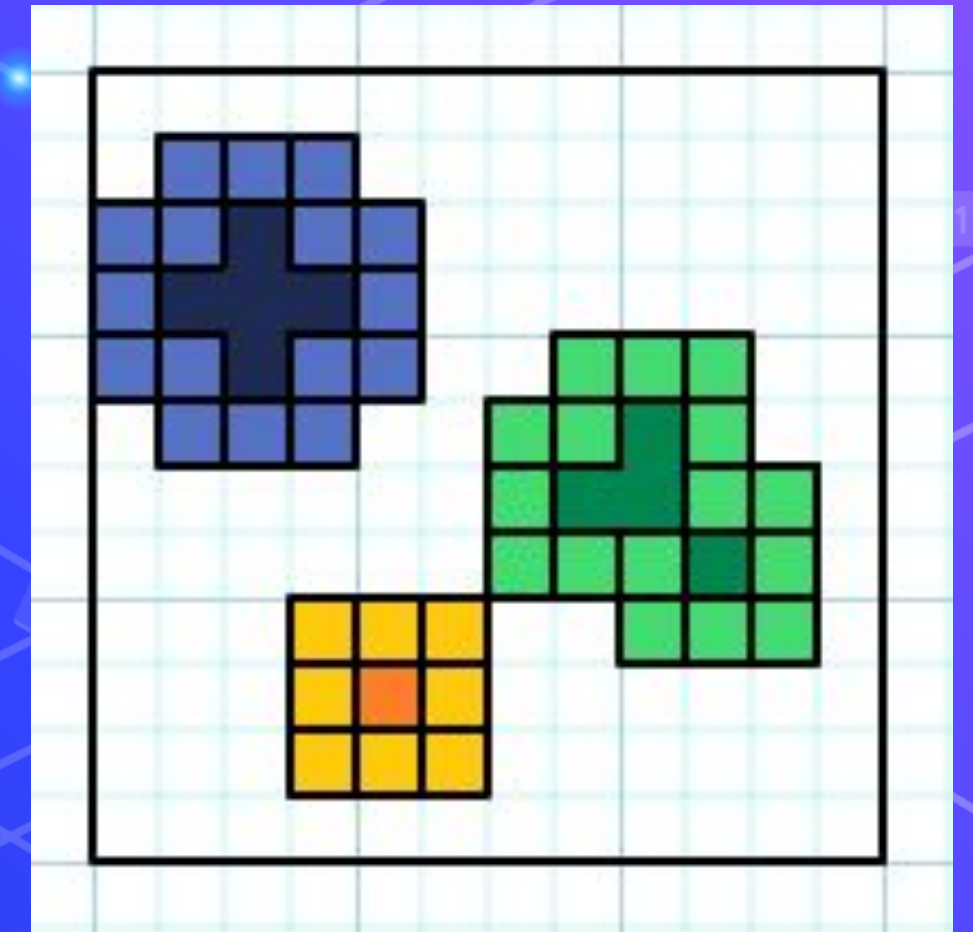
$$H_1(n) = \text{contagem}(P)$$

$$H_2(n) = \text{contagem}(\text{Grupo})$$

$$H_3(n) = \text{contagem}(\text{Jogadas}_{\text{capturas}}(J_i))$$

$$H_4(n) = \text{tamanho}(\text{Pegada}(\text{Grupo}_i, J_1)) - \text{tamanho}(\text{Pegada}(\text{Grupo}_j, J_2))$$

- As heurísticas concebidas encontram-se formalizadas acima:
 - A primeira é a contagem do número de peças de cada jogador;
 - A segunda conta ajuntamentos - grupos de peças da mesma cor;
 - A terceira é a contagem de jogadas que resultam numa captura de peça para cada jogador;
 - A última considera o tamanho das pegadas (figura à direita) dos grupos no tabuleiro (o tamanho das pegadas do jogador corrente menos o tamanho das pegadas do adversário);



Considerámos que, quanto mais complexo o conceito da heurística, mais difícil seria um jogo contra ela. 1 é a mais simples, 4 a mais complexa.

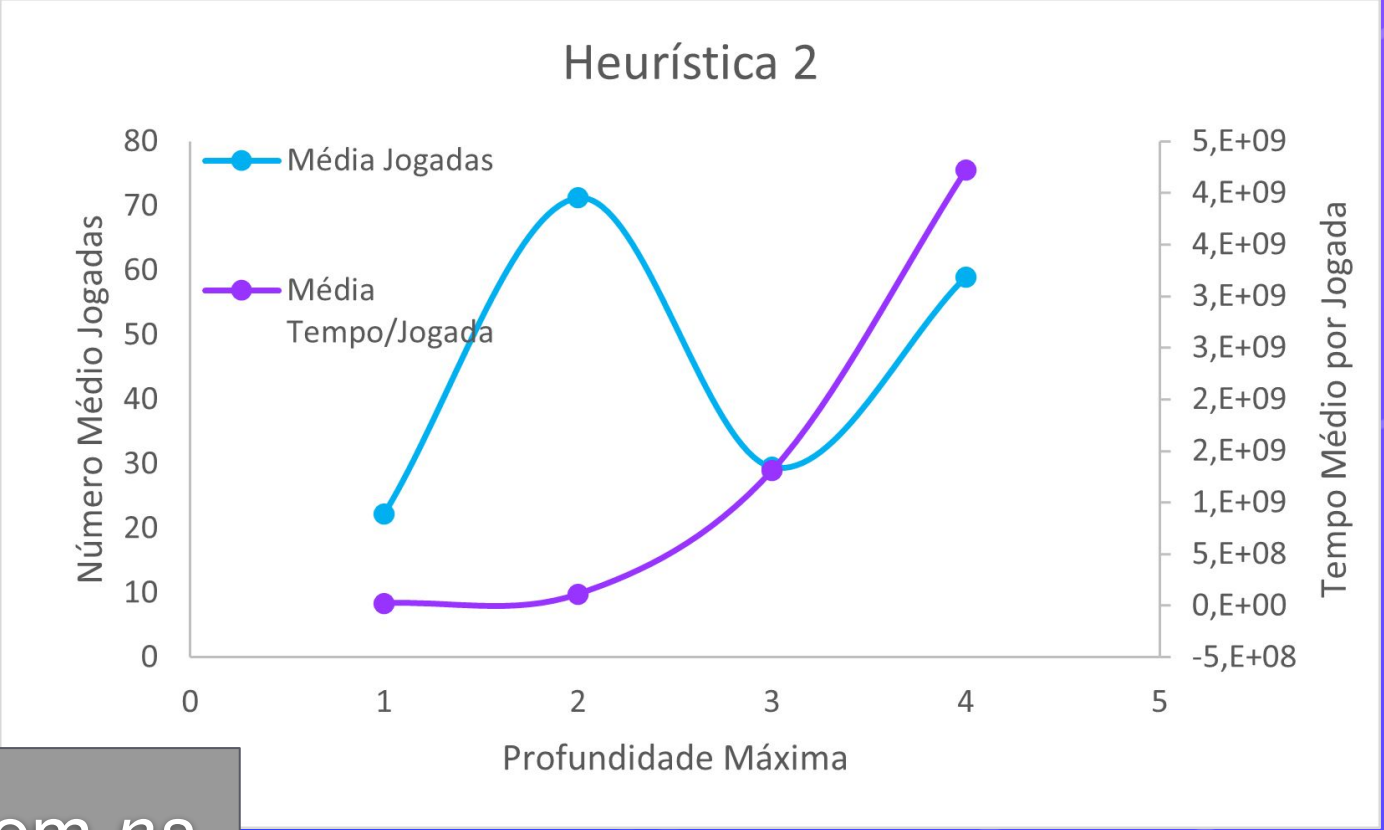
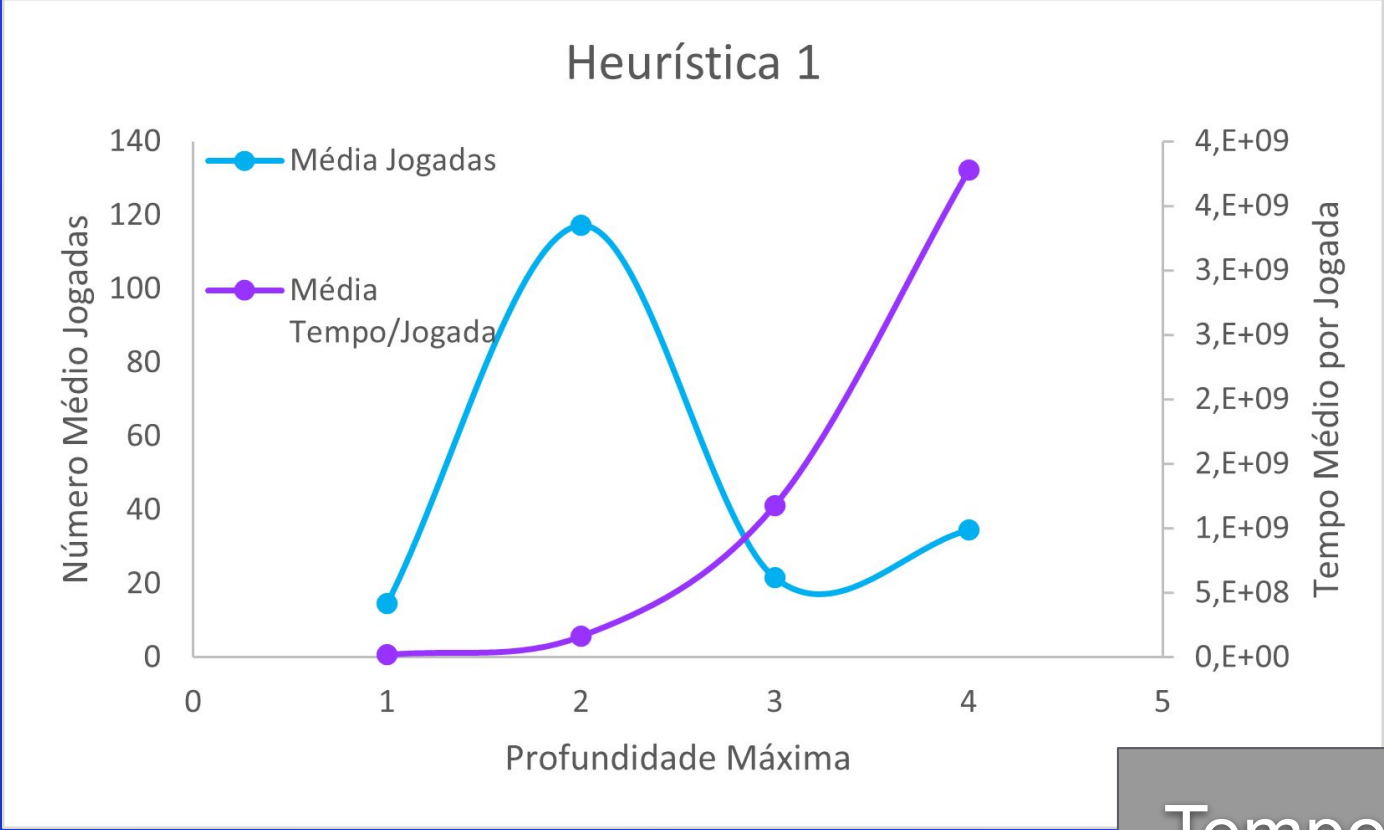
Abordagem

- Abordámos o projeto começando pela elaboração do jogo em si (modo humano vs. humano) e só depois elaborámos a inteligência artificial - conseguimos implementar todas as heurísticas propostas;
- Acerca das heurísticas, acreditámos que as heurísticas mais simples fossem menos eficazes - igualámos eficácia ao número de jogadas necessárias para um jogo terminar;
- Para avaliar a velocidade e “peso” das heurísticas, guiámo-nos pelo tempo necessário para obter uma jogada - acreditámos que as heurísticas de conceito mais complexo fossem mais lentas/pesadas;
- Acerca de operadores, os diagonais foram mais difíceis de concluir.

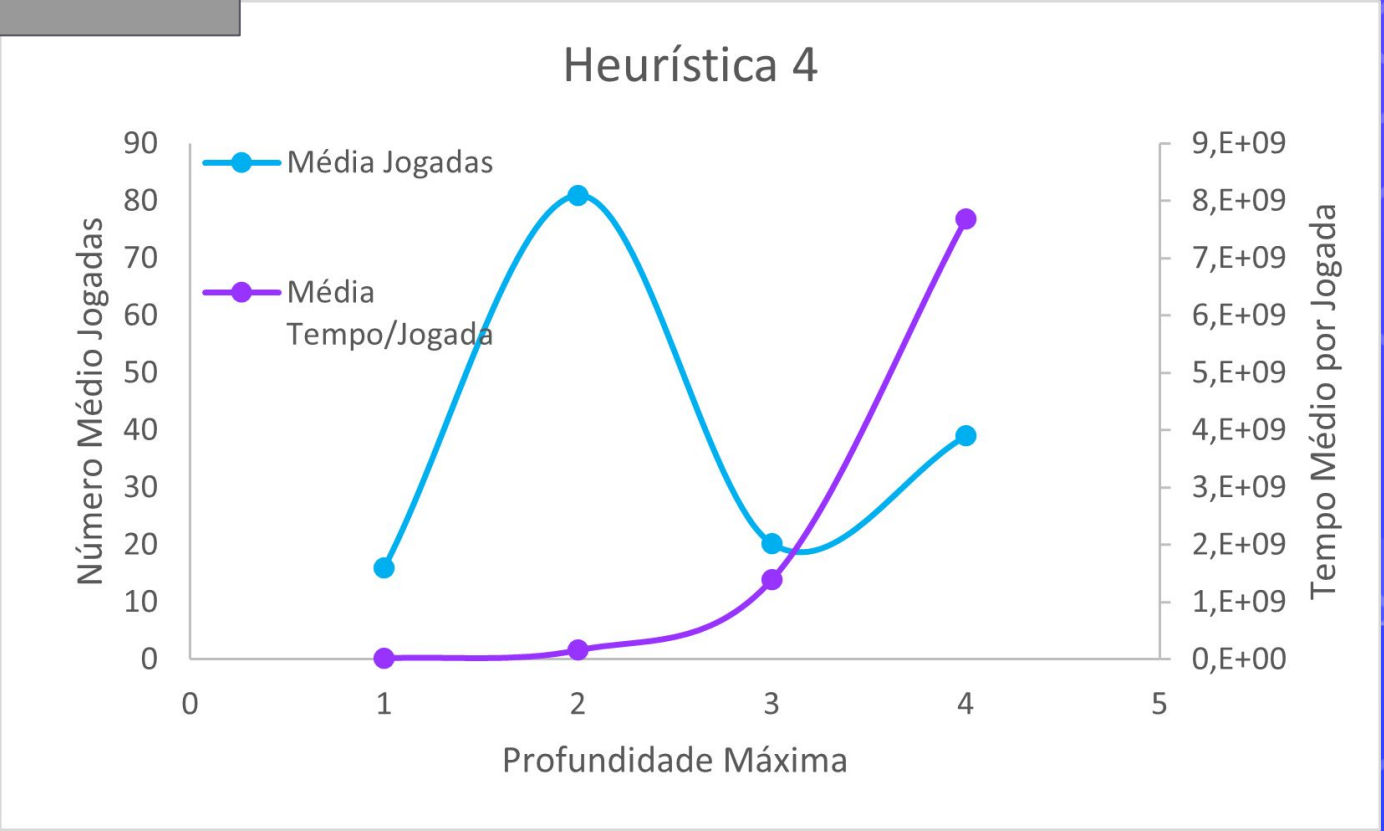
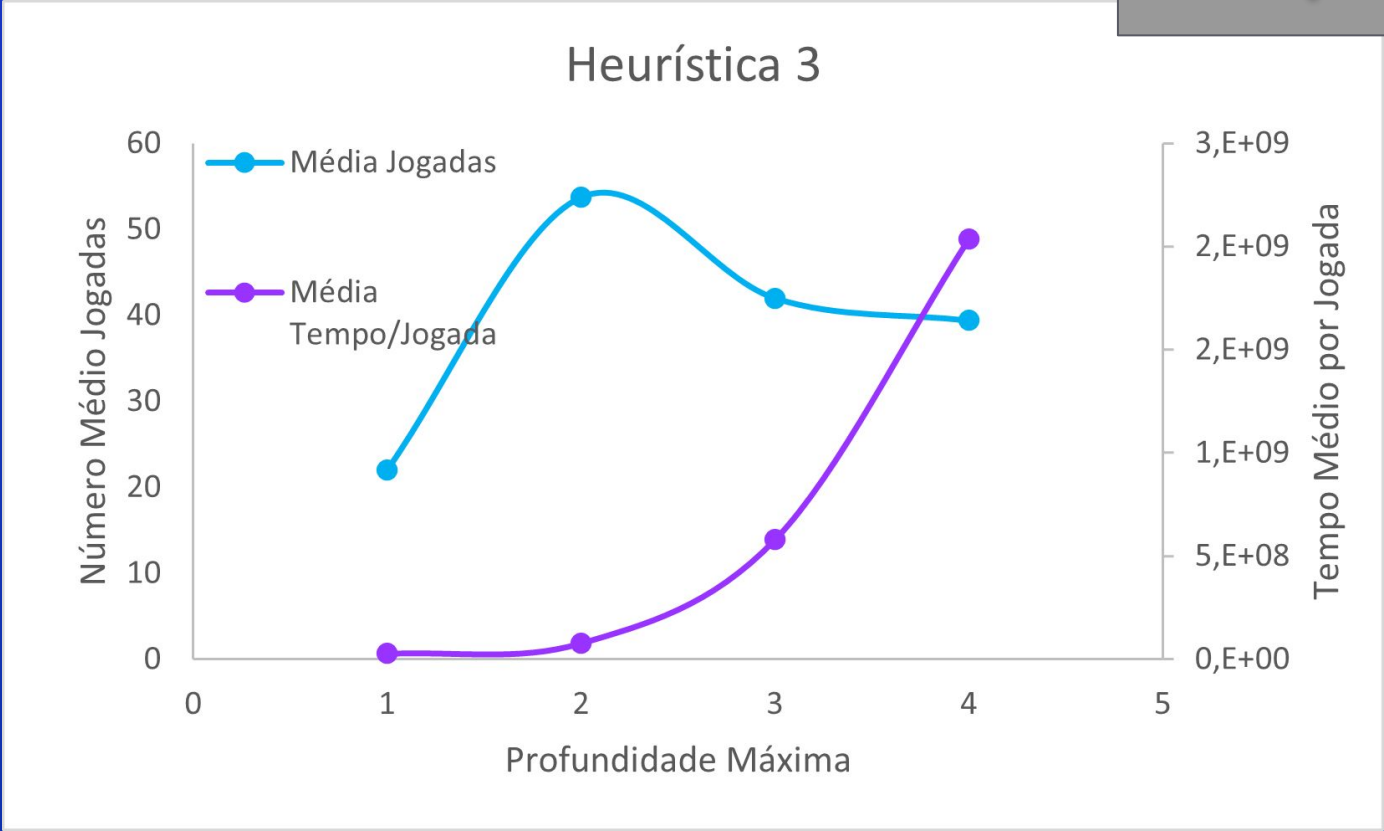
Algoritmos Implementados

- Como esperado para pesquisa adversarial, implementámos o algoritmo *Minimax*, especialmente o seu derivado, *Negamax* que melhora o primeiro tirando proveito da característica de soma-zero de jogos adversariais;
- Usámos cortes alfa-beta para maior velocidade, usando um *best accept* adaptado tendo em conta a profundidade limitada e a possibilidade de ciclos: quando múltiplas opções são igualmente boas, escolhe-se uma aleatoriamente;
- Permitimos ao jogador escolher a profundidade do algoritmo através da interface com o jogo.

Resultados



Tempos medidos em *ns*.



Resultados

- Todas as heurísticas sofrem um grande aumento - que estabiliza em maiores profundidades - no salto de profundidade 1 para 2, provavelmente por este afetar imenso a capacidade de decisão;
- O tempo médio por jogada, como esperado, e em todas as heurísticas, aumenta proporcionalmente com o aumento da profundidade;
- A heurística com melhor compromisso entre número médio de jogadas e tempo médio por jogada é a terceira, que lida com jogadas que causem captura;
- Tal faz sentido devido à captura ser uma jogada que tende a afetar significativamente o estado do jogo.

Resultados

Heurística	Vitórias	Derrotas	Detalhes das Vitórias
H ₁	0	15	Nenhuma
H ₂	10	5	5x (1v2), 1x (2v3), 4x (2v4)
H ₃	13	2	5x (1v3), 4x (2v3), 4x (2v4)
H ₄	7	8	5x (1v4), 1x (2v4), 1x (3v4)

Mais uma vez, a terceira heurística prova-se melhor.

NOTA: Não existem empates por definição nas regras do jogo.

Conclusões

- Como esperado, as heurísticas de conceito mais complexo provaram-se mais eficazes (menor nº de jogadas) que as demais;
- Por outro lado, mas ainda expectável, profundidades mais baixas resultavam em tempos por jogada menores;
- O algoritmo *Minimax* provou-se simples de implementar e, mesmo assim, capaz de responder aos requisitos;
- Acreditamos ter cumprido todos os objetivos propostos para o trabalho.

Bibliografia:

https://en.wikipedia.org/wiki/Lines_of_Action, consultado a 24 de março de 2022
<https://boardspace.net/loa/english/index.html>, consultado a 24 de março de 2022
<https://gamerules.com/rules/lines-of-action/>, consultado a 7 de abril de 2022