

Matheus Rhoden, 277948, Turma A
Rodrigo Cardoso Buske, 206526, Turma B
Txai Wieser, 217052, Turma B

INF01048 - Inteligência Artificial - UFRGS

Trabalho 1 - Parte 2

Bibliotecas

Utilizamos apenas as bibliotecas nativas, **sys**, **time** e a biblioteca disponibilizada pelo professor **common/board**. Portanto o código só precisa estar na mesma pasta que o server, igual ao *randomplayer*.

Implementação

Função de avaliação:

Para heurística que decide o score de um movimento, resolvemos utilizar a quantidade de pontos ao fim do movimento e a posição das peças, dando maior valor para peças nas bordas já que essas têm menos posições de captura possíveis, às vezes até sendo impossível de capturar.

Estratégia de parada:

Para a parada decidimos usar duas abordagens, uma profundidade máxima e um tempo máximo de execução. A profundidade máxima tenta garantir que vários "galhos" diferentes do grafo sejam explorados, ao invés de se aprofundar de mais em poucos.

Já o tempo máximo busca garantir que não seja excedido o tempo máximo de execução permitido pelo servidor. O tempo apenas impede a expansão de novos nodos, por isso deixamos

um bom tempo extra em relação ao tempo total permitido, para que a recursão possa finalizar, retornando os valores calculados para cada nodo.

Melhorias possíveis:

Experimentamos bastante com a heurística, inclusive testando duas formas completamente diferentes de calcular ela, o problema é que novamente a comparação da execução entre as duas não parecia fazer muito sentido com as nossas expectativas, então certamente poderíamos melhorar mais a heurística em si e outros valores usados, como o tempo máximo de execução. Mas estava complicado de testar e não tínhamos como garantir que deixando um tempo muito apertado ele não passaria o tempo máximo nos testes do professor.

Dificuldades:

Tivemos bastante dificuldade de testar a implementação, às vezes parecia que nosso player estava indo muito bem, outras vezes ele perdia ou empatava com o random, estava muito inconsistente e tivemos dificuldade de entender porque. Eventualmente percebemos que estávamos inicializando alfa e beta com os valores invertidos, ou seja, alfa estava recebendo infinito e beta -infinito, por isso estava causando resultados tão estranhos. Ainda assim tivemos que fazer vários testes e experimentações, já que tem muitos fatores que podem afetar o resultado final, incluindo problemas com a linguagem em si.

Bibliografia

[The Anatomy of a Game Program](#)

[An Analysis of Heuristics in Othello](#)

[An intelligent Othello player combining machine learning and game specific heuristics](#)