**Relatório Inteligência Artificial**

António Marçal 95735

Sofia Sancho 99122

**Introdução**

Neste relatório, pretendemos analisar a solução proposta para o projeto da unidade curricular de Inteligência Artificial (IA) do ano letivo de 2021/2022 (P4).

O problema apresentado tem por objetivo desenvolver um programa em Python que resolva um problema Takuzu, utilizando técnicas de procura de IA.

**Desenvolvimento do Projeto**

Começámos pordesenvolver o código para o projeto a partir das classes parcialmente definidas que tinham sido disponibilizadas pelos professores. Na nossa versão inicial, apenas os testes com tabuleiros de menor dimensão efetivamente terminavam, sendo que os outros pareciam ter um tempo de execução inaceitavelmente elevado, mesmo usando os tipos e procura mais eficientes.

Para resolver este problema, otimizámos o código para que o método Takuzu.actions analizasse se existia pelo menos uma ação que fosse a única possibilidade para uma determinada posição do tabuleiro. Nesse caso, o método devolvia apenas essa ação, impedindo que as árvores de procura gerassem um número muito elevado de nós, o que por sua vez diminuiu drasticamente o tempo de execução dos testes e a memória utilizada. Nos casos em que tal ação obrigatória não existia, o método passava a devolver apenas as duas possibilidades de ação para a primeira posição livre.

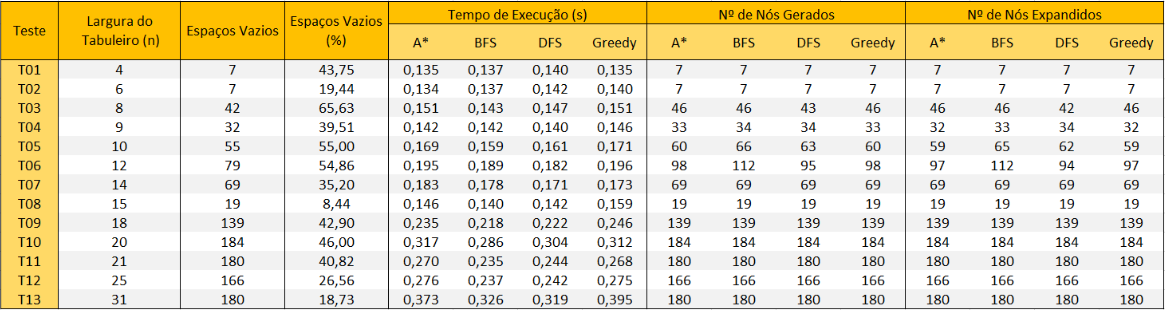
Mesmo depois desta otimização, ainda foram necessárias várias melhorias aos diversos algoritmos para atingir o tempo de execução requerido, consistindo estas maioritariamente em reestruturações do código e na elminação de repetições.

**Avaliação Experimental dos Resultados**

Após termos atingido o que acreditamos ser o tempo de execução ideal, decidimos testar o nossa proposta de solução, utilizando vários tipo de procuras diferentes, tanto informadas como não informadas. Para comparar os algoritmos, utilizámos os seguintes critérios:

* Tempo de Execução
* Número de Nós Gerados
* Número de Nós Expandidos

Os resultados desta experiência foram obtidos utilizando os testes públicos disponibilizados pelos professores, e os resultados podem ser vistos na tabela em anexo.



Como podemos observar, os diversos tipos de procura testados tiveram tempos de execução bastante diferentes, pois para além da expectável tendência a aumentar consoante a dimensão do tabuleiro, também mostram diferenças em relação uns aos outros.

Comparando os tempos de execução, podemos concluir que, no caso geral, as procuras A\* e greedy\_search são consideravelmente mais lentas do que as procuras não informadas, com a exceção apenas dos testes com tabuleiros mais pequenos. Inicialmente, isto foi um fenómeno inesperado, visto que a procura informada deveria estar mais otimizada, ou seja, o seu tempo de execução também deveria ser mais baixo. Após considerarmos mais atentatmente o motivo por trás desta observação, concluimos que se deve ao uso de uma heurística pouca útil, ou seja, as procuras A\* e greedy\_search verificaram mais condições que a DFS e a BFS, mas não obtiveram um grande ganho com isso.

Quanto às diferenças entre os tipos de procuram informada, vemos que A\*, nalgumas situações, tem ganhos consideráveis em relação à greedy\_search, sendo que, noutras instâncias, os tempos de execução das duas são bastante semelhantes.

Quanto às procuras não informadas, a BFS mosta uma ligeira melhoria comparativamente à DFS, mas sem aparentar ter uma relação consistente com o tamanho do tabuleiro ou o número de espaços livres.