Grupo 3 80832 Margarida Ferreira 81805 Duarte David

Relatório do Primeiro Projeto de IA 2017-2018

Modelação do Problema:

Pretende-se resolver o problema de encontrar uma solução para uma variante do jogo *Same Game*. Para tal, modela-se o problema como uma procura, onde cada configuração do tabuleiro é um potencial nó. Se um movimento válido (retirar um grupo com mais de um elemento) numa dada configuração gera outra configuração, então o nó correspondente à primeira pode gerar o nó correspondente à segunda. A função de custo é definida como sendo o custo anterior incrementado de um. A função heurística utilizada é uma contagem do número de grupos ainda existentes no estado (contando com grupos de um único elemento). Em caso de empate, explora-se primeiro os nós gerados mais cedo.

A função heurística escolhida consiste na contagem de grupos ainda presentes no tabuleiro. Esta heurística corresponde à contagem do número de movimentos necessários para obter uma solução numa versão simplificada do problema, em que qualquer grupo pode ser removido (mesmo que tenha apenas um elemento) e não há compactação vertical nem horizontal. Esta heurística não é admissível e por este motivo não pode ser garantida a optimalidade da procura A\*.

Resultados Experimentais:

Foram resolvidas várias instâncias do jogo *Same Game* (ver Figura 1) com procura em profundidade primeiro, procura gananciosa e procura A\*, tendo sido obtidos os seguintes resultados:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Tabuleiro** | **Tamanho** | **Número de cores** | **Procura** | **Nós gerados** | **Nós expandidos** | **Tempo(s)** |
| **1** | 5 x 4 = 20 | 2 | PPP | 0 | 1 | 0.000128 |
| Gananciosa | 0 | 1 | 0.000210 |
| A\* | 0 | 1 | 0.000193 |
| **2** | 5 x 4 = 20 | 3 | PPP | 7 | 4 | 0.000820 |
| Gananciosa | 6 | 3 | 0.001177 |
| A\* | 7 | 4 | 0.001397 |
| **3** | 4 x 10 = 40 | 5 | PPP | 74701 | 74702 | 9.943855 |
| Gananciosa | 74701 | 74702 | 19.267899 |
| A\* | 74701 | 74702 | 20.780052 |
| **4** | 4 x 10 = 40 | 3 | PPP | 85 | 54 | 0.011061 |
| Gananciosa | 59 | 42 | 0.014429 |
| A\* | 43 | 24 | 0.010846 |
| **5** | 4 x 10 = 40 | 5 | PPP | 3123363 | 3123308 | 416.189735 |
| Gananciosa | 256 | 319 | 0.089831 |
| A\* | 91 | 16 | 0.026579 |
| **6** | 4 x 10 = 40 | 4 | PPP | 6 | 7 | 0.001569 |
| Gananciosa | 6 | 7 | 0.002442 |
| A\* | 6 | 7 | 0.002406 |
| **7** | 6 x 5 = 30 | 3 | PPP | 6 | 3 | 0.001168 |
| Gananciosa | 6 | 3 | 0.002035 |
| A\* | 15 | 10 | 0.004267 |
| **8** | 6 x 5 = 30 | 3 | PPP | 7 | 3 | 0.001220 |
| Gananciosa | 7 | 3 | 0.002383 |
| A\* | 8 | 4 | 0.002664 |
| **9** | 15 x 10 = 150 | 3 | PPP | 351 | 44 | 0.118043 |
| Gananciosa | 190 | 16 | 0.180421 |
| A\* | 191 | 17 | 0.176785 |
| **10** | 15 x 10 = 150 | 4 | PPP | Não terminou em tempo útil | | |
| Gananciosa | 443 | 33 | 0.498485 |
| A\* | 1224 | 811 | 1.241444 |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Tabuleiro 1**  1 2 1 2 1  2 1 2 1 2  1 2 1 2 1  2 1 2 1 2 | **Tabuleiro 2**  1 2 2 3 3  2 2 2 1 3  1 2 2 2 2  1 1 1 1 1 | **Tabuleiro 3**  3 1 3 2  1 1 1 3  1 3 2 1  1 1 3 3  3 3 1 2  2 2 2 2  3 1 2 3  2 3 2 3  5 1 1 3  4 5 1 2 | | **Tabuleiro 4**  3 1 3 2  1 1 1 3  1 3 2 1  1 1 3 3  3 3 1 2  2 2 2 2  3 1 2 3  2 3 2 3  2 1 1 3  2 3 1 2 | **Tabuleiro 5**  1 1 5 3  5 3 5 3  1 2 5 4  5 2 1 4  5 3 5 1  5 3 4 4  5 5 2 5  1 1 3 1  1 2 1 3  3 3 5 5 |
| **Tabuleiro 7**  1 1 3 3 3 3  2 2 3 3 3 3  2 2 3 3 3 3  2 2 3 3 3 3  2 2 3 3 3 3 | **Tabuleiro 8**  3 3 0 3 3 3  2 2 0 3 3 3  2 2 0 3 3 3  2 2 1 3 3 3  2 2 1 3 3 3 |
| **Tabuleiro 9**  1 3 2 1 2 1 2 2 1 2 2 1 1 3 1  1 3 3 2 1 2 2 2 3 1 2 1 2 3 1  1 1 1 2 3 2 3 3 2 2 3 1 1 3 1  1 2 2 2 3 3 3 3 1 2 1 2 1 3 2  1 3 1 3 2 2 2 2 3 1 1 2 3 2 1  1 1 2 2 2 1 1 3 2 1 2 3 1 3 1  3 1 3 2 2 2 3 3 3 1 3 3 2 1 1  3 2 1 2 1 3 1 2 1 2 3 1 1 3 3  2 3 1 2 3 3 1 2 3 3 3 2 1 1 1  2 2 1 1 2 1 2 2 1 1 3 2 2 2 2 | | | **Tabuleiro 10**  4 4 4 2 3 2 2 2 2 1 2 3 4 4 3  1 4 4 2 1 2 4 1 1 2 4 4 3 4 1  2 3 2 2 2 4 3 2 2 2 2 1 4 2 3  3 3 3 4 2 2 4 3 3 4 4 2 4 1 1  1 3 3 4 3 1 1 1 1 1 1 2 4 2 3  4 2 3 3 1 3 3 2 2 4 3 3 2 1 4  3 1 2 1 2 3 2 4 2 4 4 3 4 4 2  3 4 3 1 1 3 4 4 1 2 3 1 1 4 2  1 3 1 3 2 1 2 1 3 4 4 3 2 3 1  1 2 1 2 2 2 1 3 2 4 3 2 1 2 1 | | **Tabuleiro 6**  4 4 4 2  4 4 4 3  4 4 4 1  4 4 4 4  4 4 4 2  4 4 4 4  4 4 4 3  4 4 4 3  4 4 4 4  4 4 4 2 |
| *Figura 1: São representados os vários tabuleiros a testar* | | | | | |

Análise dos Resultados:

Existem dois casos distintos a ter em conta: o caso em que o problema tem solução e o caso em que não tem.

Nos problemas sem solução, é sempre percorrida toda a árvore de procura, pelo que não se espera uma melhoria com o uso de procura informada (antes pelo contrário, visto que o cálculo da função heurística tem custos). Tal observou-se nos problemas 1, 3 e 6, que não têm solução. O número de estados gerados/expandidos é idêntico nas três procuras, mas o tempo das procuras informadas é aproximadamente o dobro (o número de chamadas a board\_find\_groups duplica).

Dentro dos problemas com solução, distinguem-se os problemas triviais (aqueles em que qualquer sequência de ações válidas chega ao estado objetivo – Problemas 7 e 8). Nestes, a PPP termina com sucesso quando chega ao fim do primeiro ramo, enquanto que as outras podem ir efetuando a procura em mais do que um ramo. Por esta razão, observa-se que as procuras informadas demoram mais do que a PPP. Observa-se ainda que a procura A\* demora mais do que a procura Gananciosa, uma vez que o peso da função de custo na A\* faz com que esta se comporte de forma mais semelhante a uma procura de custo uniforme.

Nos restantes problemas (2, 4, 5, 9 e 10) observa-se que em quase todos os casos as procuras informadas são *muito* mais rápidas do que a PPP (Isto deve se ao facto de esta não ser guiada por uma função de custo - como a procura em custo uniforme - nem por uma função heurística -como as procuras informadas), e que a A\* é bastante mais rápida do que a procura gananciosa (uma vez que o peso da função de custo a incentiva a afastar-se da origem). As exceções a estes casos são o problema 2 (que é quase trivial), o problema 9 (em que a A\* e a Gananciosa demoram mais ou menos o mesmo tempo, embora sejam as duas bastante melhores) e o problema 10 (onde a procura Gananciosa é bastante mais rápida do que a A\*). Concluímos que em problemas com muitas soluções, a procura Gananciosa revela-se melhor do que a A\*, pois a função de custo penaliza a exploração de nós mais longe da origem, obrigando a explorar mais ramos da árvore de procura.

Em relação à completude, temos a garantia de que as Procuras A\* e gananciosa são completas (embora não levem sempre à solução ótima). Temos também a garantia, devido à natureza do problema, que a PPP é completa, uma vez que cada configuração do tabuleiro de jogo apenas transita de modo a diminuir o número total de peças (a monotonia estrita de um dos parâmetros da configuração do tabuleiro garante a não repetição de nós num ramo da árvore de procura).

Considerações Adicionais

Para além da heurística utilizada, considerou-se a heurística que conta apenas os grupos de tamanho superior a 2, mas verificou-se que atrasava significativamente a procura e não penalizava estados com vários elementos sozinhos (que não podem ser removidos). Foi ainda considerada a heurística correspondente à contagem de número de cores ainda presentes no tabuleiro. Apesar de esta heurística ser admissível, verificou-se que as procuras se tornam substancialmente mais demoradas, pois o seu valor está muito afastado do valor real (h<<h\*), pois pode haver muitos grupos de cada cor, que têm de ser eliminados em jogadas distintas.

Todas estas heurísticas podem ser calculadas em tempo linear (em função do número de casas do tabuleiro).

Conclusões

Observou-se que a utilização de procura informada com a função heurística escolhida acelerou bastante a procura nos casos não triviais e satisfazíveis. Destes, os casos com um maior número de soluções eram mais facilmente resolvidos com a procura gananciosa, sendo que aqueles com menos soluções (e muitos “becos-sem-saída”) se resolviam mais depressa com a procura A\*, pois, devido à influência da função de custo, explora mais homogeneamente a árvore de procura.