|  |  |  |
| --- | --- | --- |
| Grupo 7 | 93686 André Oliveira | 93735 Maria Ribeiro |

**Relatório do Primeiro Projeto de IA**

Heurística:

A **função heurística** utilizada devolve apenas 2 valores: 1 se o objetivo estiver na mesma linha ou 2 se for preciso fazer mais do que 1 movimento.

Resultados Experimentais:

Foram resolvidas várias instâncias do jogo **Ricochet Robots** (Figura 1) executando uma procura em largura primeiro (PLP), uma procura em profundidade primeiro (PPP), uma procura gananciosa e uma procura A\*. Os resultados podem ser encontrados na Tabela 1. Para cada teste foi dado um tempo máximo de 2 minutos, pelo que se o teste não for concluído nesse tempo, foi registado como “*Não terminou em tempo útil*”.

Análise dos Resultados:

Verificamos que a Procura em Profundidade Primeiro, muitas das vezes não termina em tempo útil uma vez que entra em ciclos infinitos como por exemplo, mover sempre o mesmo robot para cima e para baixo, não chegando, portanto, ao objetivo como seria esperado. O único teste em que isto não se verificou foi o do Tabuleiro 4, que indica que foi descoberto primeiro um caminho para a solução do problema do que para um ramo infinito. Este é, portanto, um algoritmo de procura que não se adequa ao problema apresentado, não sendo eficiente nem completo. Como o algoritmo procura em profundidade primeiro, mesmo que encontre uma solução, há uma grande probabilidade de não ser ótima pois poderá já ter explorado vários estados que não são necessários para a resolução do problema.

A Procura em Largura Primeiro demonstrou ser uma das melhores funções de procura uma vez que os tabuleiros são pequenos e apenas precisamos de 4 jogadas no máximo para terminar o jogo. Se fossem necessárias mais jogadas, não seria um bom algoritmo de procura. A quantidade de nós que teriam de ser explorados iria aumentar exponencialmente, levando, por isso, a um elevado tempo de execução e número de nós gerados e explorados. Não é, por isso, uma procura eficiente especialmente para tabuleiros de maiores dimensões ou com maior número de jogadas necessárias para terminar. Porém, como explora nós com menor profundidade primeiro, efetua uma procura completa. Visto que cada ramo da árvore tem um custo unitário, este tipo de procura vai encontrar sempre a solução ótima para o problema.

O algoritmo de procura Greedy foi o que demonstrou melhores resultados. Como é algoritmo de procura não informada, escolhe a ação que parecer ótima localmente, podendo não ser a escolha indicada uma vez não tem em conta as paredes nem a posição dos robôs em volta. Tendo em conta, que o custo de cada jogada é unitário, pode nem sempre encontrar a solução ótima, porém irá encontrar uma solução próxima da ótima num tempo de execução razoável. É, portanto, um algoritmo eficiente, porém, não completo.

A procura A\* era aquela que prometia ser mais eficiente em termos de tempo e memória, ou seja, menos estados gerados e expandidos uma vez que este algoritmo expande o nó que tiver o menor valor de (*custo acumulado + previsão de custo até ao objetivo*). Porém, a heurística utilizada não é ótima, acabando, então, por não ser eficiente neste contexto. Com uma heurística mais adequada seria o algoritmo de escolha e seria, com certeza, eficiente e completo.

A primeira heurística pensada foi devolver a distância de Manhattan entre o objetivo e o respetivo robot. Apesar de ser muito rápida e de expandir poucos nós, não é admissível como é o caso do Tabuleiro 8, em que custo real é 4, porém a heurística iria devolver 5.

Mudamos então para a heurística atual que apenas devolve 2 valores. Devolve 1 se o objetivo estiver na mesma linha ou coluna que o robot correspondente ou 2 caso contrário. Claramente esta não é uma heurística adequada nem ótima. Não verifica se há barreiras a meio nem se o que seria considerado um movimento, na realidade, seria mais devido ao objetivo não se encontrar numa borda ou com outro robot encostado.

Estas heurísticas poderiam ser melhoradas se fosse tida em conta as posições dos outros robots e das paredes para um cálculo de um valor de custo mais aproximado do real.

Conclusões Finais:

Ao comparar todos os tipos de procura contemplados na Tabela 1, podemos em primeira instância descartar a Procura em Profundidade Primeiro como solução ideal para este problema, uma vez que raramente termina em tempo útil.

Devido à heurística utilizada, a procura A\* também não é a função de procura mais eficiente.

Apesar de a Procura em Largura Primeiro ocupar mais espaço, é o algoritmo que encontra a solução ótima. O algoritmo Greedy apresenta um menor escalamento de tempo de execução em relação ao tamanho do tabuleiro e um menor número de nós gerados. Em ambos os testes, há um grande consumo de memória. E a eficiência é muito semelhante entre ambas.

Para a submissão, a procura escolhida para a versão final do projeto, foi a Procura Iterativa em Profundidade, uma vez que é uma procura completa e ótima, sendo que o custo de todos os ramos é 1, e que, no geral, apresentou tempos de execução e espaço ocupado mais reduzidos.

Anexos:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Tabuleiro** | **Tamanho** | **Procura** | **Nós gerados** | **Nós expandidos** | **Tempo (s)** |
| **1** | 4x4 = 16 | PLP | 1756 | 268 | 0.67506 |
| PPP | Não terminou em tempo útil | | |
| Gananciosa | 1554 | 207 | 0.65872 |
| A\* | 2663 | 393 | 1.88998 |
| **2** | 5x5 = 25 | PLP | 68 | 8 | 0.00232 |
| PPP | Não terminou em tempo útil | | |
| Gananciosa | 19 | 2 | 0.00066 |
| A\* | 215 | 24 | 0.02493 |
| **3** | 6x6=36 | PLP | 323 | 33 | 0.02692 |
| PPP | Não terminou em tempo útil | | |
| Gananciosa | 53 | 5 | 0.00598 |
| A\* | 1357 | 139 | 0.52013 |
| **4** | 3x3 = 9 | PLP | 6 | 1 | 0.00000 |
| PPP | 6 | 1 | 0.00000 |
| Gananciosa | 6 | 1 | 0.00000 |
| A\* | 26 | 5 | 0.00099 |
| **5** | 3x3 = 9 | PLP | 12 | 3 | 0.00100 |
| PPP | Não terminou em tempo útil | | |
| Gananciosa | 9 | 2 | 0.00099 |
| A\* | 22 | 6 | 0.00099 |
| **6** | 4x4 = 16 | PLP | 90 | 12 | 0.00201 |
| PPP | Não terminou em tempo útil | | |
| Gananciosa | 22 | 3 | 0.00099 |
| A\* | 471 | 62 | 0.06383 |
| **7** | 4x4 = 16 | PLP | 346 | 44 | 0.03589 |
| PPP | Não terminou em tempo útil | | |
| Gananciosa | 78 | 9 | 0.00499 |
| A\* | 1066 | 140 | 0.36913 |
| **8** | 5x5 = 25 | PLP | 1098 | 265 | 0.83517 |
| PPP | Não terminou em tempo útil | | |
| Gananciosa | 60 | 7 | 0.00299 |
| A\* | 2384 | 329 | 1.65310 |

**Tabela 1:** Resultados obtidos com as diversas funções de procura.

|  |  |  |  |
| --- | --- | --- | --- |
| **Tabuleiro 1** | **Tabuleiro 2** | **Tabuleiro 3** | **Tabuleiro 4** |
| **Tabuleiro 5** | **Tabuleiro 6** | **Tabuleiro 7** | **Tabuleiro 8** |

***Imagem 1.*** *Visualização dos testes*