António Cruz: ist195536
Martim Santos: ist95638

Relatório IA P3 2022

Procura Não Informada:

DFS	Tempo (sec)	Memória (Kb)	# Nós Gerados	# Nós Expandidos
Input1.txt	0.571	28712	5	5
Input2.txt	0.580	28792	40	39
Input3.txt	0.563	28856	27	27
Input4.txt	0.600	29112	174	174
Input5.txt	0.648	30336	782	780
Input6.txt	0.589	29596	233	228
Input7.txt	0.604	30184	409	403
Input8.txt	0.584	29280	125	116
Input9.txt	0.583	29292	125	116
Input10.txt	0.588	29123	110	105

BFS	Tempo (sec)	Memória (Kb)	# Nós Gerados	# Nós Expandidos
Input1.txt	0.598	28772	5	5
Input2.txt	1.040	28868	153	153
Input3.txt	0.675	28768	27	27
Input4.txt	0.597	28992	174	174
Input5.txt	0.640	29600	799	799
Input6.txt	0.648	29784	541	541
Input7.txt	0.686	30256	606	606
Input8.txt	0.619	29160	187	187
Input9.txt	0.631	29168	187	187
Input10.txt	0.619	29148	174	174

Nota: Os valores apresentados nas procuras não informadas são aceitáveis, no entanto, ao tentar resolver tabuleiros Numbrix de maior dimensão e com menos valores iniciais já colocados, tornam-se impraticáveis.

Procura Informada:

Greedy	Tempo (sec)	Memória (Kb)	# Nós Gerados	# Nós Expandidos
Input1.txt	0.574	28776	5	5
Input2.txt	0.595	29032	146	145
Input3.txt	0.573	28852	27	27
Input4.txt	0.597	29036	77	74
Input5.txt	0.555	28792	40	36
Input6.txt	0.668	30256	438	430
Input7.txt	0.784	29336	170	162
Input8.txt	0.656	29472	112	97
Input9.txt	0.626	29180	112	97
Input10.txt	0.638	29076	89	79

A*	Tempo (sec)	Memória (Kb)	# Nós Gerados	# Nós Expandidos
Input1.txt	0.566	28800	5	5
Input2.txt	0.608	29056	139	136
Input3.txt	0.564	28864	27	27
Input4.txt	0.597	28888	44	39
Input5.txt	0.579	29324	266	252
Input6.txt	0.673	30656	508	500
Input7.txt	0.679	29260	128	119
Input8.txt	0.691	29396	158	151
Input9.txt	0.638	29424	158	151
Input10.txt	0.597	29192	122	116

TÉCNICO LISBOA

António Cruz: ist195536
Martim Santos: ist95638

Discussão Dos Resultados:

Observando os resultados obtidos e com o auxílio dos conhecimentos teóricos, concluímos que a procura não informada é extremamente ineficiente para o problema em questão.

Sem a utilização de uma heurística, o conjunto de nós gerados e explorados cresce de forma acentuada e invalida a eficiência destas procuras (tanto a DFS como a BFS).

De forma a tornar estas procuras exequíveis, decidimos restringir o número de ações que cada estado considera como possíveis (o que permitiu que estas procuras apresentassem resultados satisfatórios para tabuleiros de tamanho reduzido), cortando as seguintes ações:

- Uma ação que coloque um número *num* numa posição que esteja a uma distância de Manhattan de *prev_num* e *next_num* superior à diferença numérica entre estes e *num* (Sendo *prev_num* (resp. *next_num*) o valor mais alto (resp. baixo) dos valores mais baixos (resp. altos) que *num* já colocados no tabuleiro);
- Uma ação que coloque um número *num* numa posição que não tenha posições vizinhas livres suficientes para colocar os números adjacentes a *num*
- Uma ação que, ao colocar o número na respetiva posição, bloqueie os vizinhos dessa posição, isto é, impeça os vizinhos de futuramente colocar os seus adjacentes à sua volta (utiliza-se a mesma lógica que o ponto anterior)
- Uma ação que, ao ser executada, dê aso a um zero isolado (um zero que não é passível de ser preenchido)

Tanto a DFS como a BFS são completas neste caso, já que o espaço de estados é infinito (não há ciclos).

Notamos, após a aplicação das restrições, que a Depth First Search apresenta uma maior eficiência tanto em termos de memória como em termos de tempo (para o problema em questão) do que a Breadth First Search (este facto é principalmente verificado em tabuleiros de maiores dimensões, pois para os tabuleiros testados nos testes públicos, os valores são extremamente semelhantes como se pode comprovar pelas figuras acima).

Embora a implementação das restrições acima apresentadas tenha reduzido substancialmente os consumos de tempo e memória das procuras não informadas, estas continuam a ser pouco eficientes, pelo que é necessário recorrer a procuras informadas (nomeadamente os algoritmos de Greedy Search e A* Search).

Com esse intuito, foi desenvolvida uma heurística que visa, como qualquer heurística, guiar a procura na direção da solução. Foram testadas várias opções, algumas das quais apresentamos de seguida.

TÉCNICO LISBOA

António Cruz: ist195536
Martim Santos: ist95638

Histórico de Heurísticas:

- Começámos por experimentar a heurística mais básica em que valorizávamos os tabuleiros que tivessem o menor número de espaços por preencher (o que não provou ser muito útil);
- De seguida, decidimos valorizar ações que colocassem números em zonas com poucos espaços brancos em redor, de forma a tornar o tabuleiro menos esparso e evitar a formação de "ilhas de zeros", sendo uma "ilha de zeros" um conjunto de zeros adjacentes que não são passíveis de ser preenchidos (estão isolados)
- Por fim, a heurística que trouxe melhores resultados consiste nas duas anteriores, dando prioridade a uma face à outra multiplicando por um coeficiente, e ainda uma DFS que procura "ilhas de zeros" com, no máximo, 4 zeros

A procura Greedy não é, na maior parte dos casos, completa, pois pode entrar em ciclos, no entanto, com a heurística implementada, o tradeoff de completude por mais eficiência temporal é benéfico e apresenta melhores resultados. Ademais, a heurística desenvolvida teve em atenção a prevenção de criação de ciclos, pelo que se torna mais viável a utilização desta procura.

No caso do algoritmo A* Search, a heurística implementada não provou ser boa o suficiente para que esta procura possa tirar o maior proveito e, como tal, apresenta resultados não satisfatórios.

A heurística por nós apresentado não é admissível, logo conclui-se que a procura não é ótima.

Repáramos que ao alterar o coeficiente do número de zeros na heurística, os valores oscilam bastante nos nós gerados e explorados em tabuleiros maiores para esta procura.

Conclusões Finais:

As procuras não informadas ofereceram uma boa visão inicial dum problema crucial em procuras em Inteligência Artificial que se prende com a dimensão do espaço de estados.

Como solução para este obstáculo, restringimos as ações, tornando assim as procuras não informadas numa boa arma para resolver tabuleiros de pequena e média dimensão (até 10x10 sensivelmente, dependendo do quão vazio e esparso está o tabuleiro inicial) de forma extremamente eficiente.

Torna-se, no entanto, inconcebível a utilização deste tipo de procuras para tabuleiros de grande dimensão e, para tal, desenvolvemos uma heurística que permitiu a utilização de procuras informadas.

A procura Greedy mostrou ser a procura mais eficiente no geral (tanto em memória como em tempo), ficando a procura A* aquém das expectativas por termos desenvolvido uma heurística menos positiva.