



FCTUC **FACULDADE DE CIÊNCIAS
E TECNOLOGIA**
UNIVERSIDADE DE COIMBRA

Introdução à Inteligência Artificial

2016/2017 – 2º Semestre

Trabalho Prático Nº2: Sokoban

Trabalho realizado por:
Hugo Teixeira Nº 2002126180
Rui Rocha Nº 2010135453

hrafael@student.dei.uc.pt
rmrocha@student.dei.uc.pt

Índice

Introdução	2
Métodos de procura cega.....	3
Algoritmo BFS	3
Algoritmo de pesquisa em profundidade limitada	3
Algoritmo de aprofundamento progressivo	4
BFS vs Aprofundamento Progressivo	4
Nós Visitados	4
Nós Expandidos	5
Métodos de procura informada	6
Heurísticas.....	6
Heurística 1	6
Heurística 2	6
Heurística 3	6
Heurística 4	6
Algoritmo de pesquisa Sôfrega	7
Algoritmo de pesquisa A*	8
A* vs Sôfrega.....	10
Nós Expandidos	10
Nós Visitados	11
Conclusão	12

Introdução

O objetivo deste trabalho é a implementação de alguns métodos de pesquisa cega e de pesquisa informada, para resolver (caso tenham solução) alguns problemas de *Sokoban*. Foi-nos fornecido uma “*Unity Package*”, com a modelação do problema, onde o grupo tinha de implementar os métodos referidos. Na implementação dos métodos de pesquisa informada, desenvolvemos algumas heurísticas, além da fornecida no enunciado, que permitissem melhorar a performance destes algoritmos em relação aos de procura cega. Estes foram testados nalguns mapas fornecidos na “*Unity Package*”.

Métodos de procura cega

Algoritmo BFS

	Visitados	Expandidos	Ações	Tamanho Queue	Solução
Mapa1	22	58	5	9	S
Mapa2	386	1123	8	230	S
Mapa3	134251	382032	29	15861	S
Mapa4	13812	33496	50	700	S
Mapa5	708161	1964594	35	125481	S
Mapa6	71617	158845	89	2943	S
Mapa8	1073204	2802189	93	28441	S

O algoritmo BFS (*Breadth First Search*), encontra sempre uma solução, podendo não ser a melhor.

Algoritmo de pesquisa em profundidade limitada

	Visitados	Expandidos	Ações	Tamanho Stack	Solução	Limite
Mapa1	125	345	5	9	S	5
Mapa2	4214	13196	8	18	S	8
Mapa3						29
Mapa4						50
Mapa5						35
Mapa6						89
Mapa8						93

O algoritmo de procura limitada, não conseguimos obter uma solução para a maior parte dos mapas, por este não encontrar uma solução em tempo útil. Mas em relação ao BFS nota-se uma melhoria em termos do uso da stack em relação à Queue, mas em relação aos nós visitados e expandidos tem uma complexidade enorme.

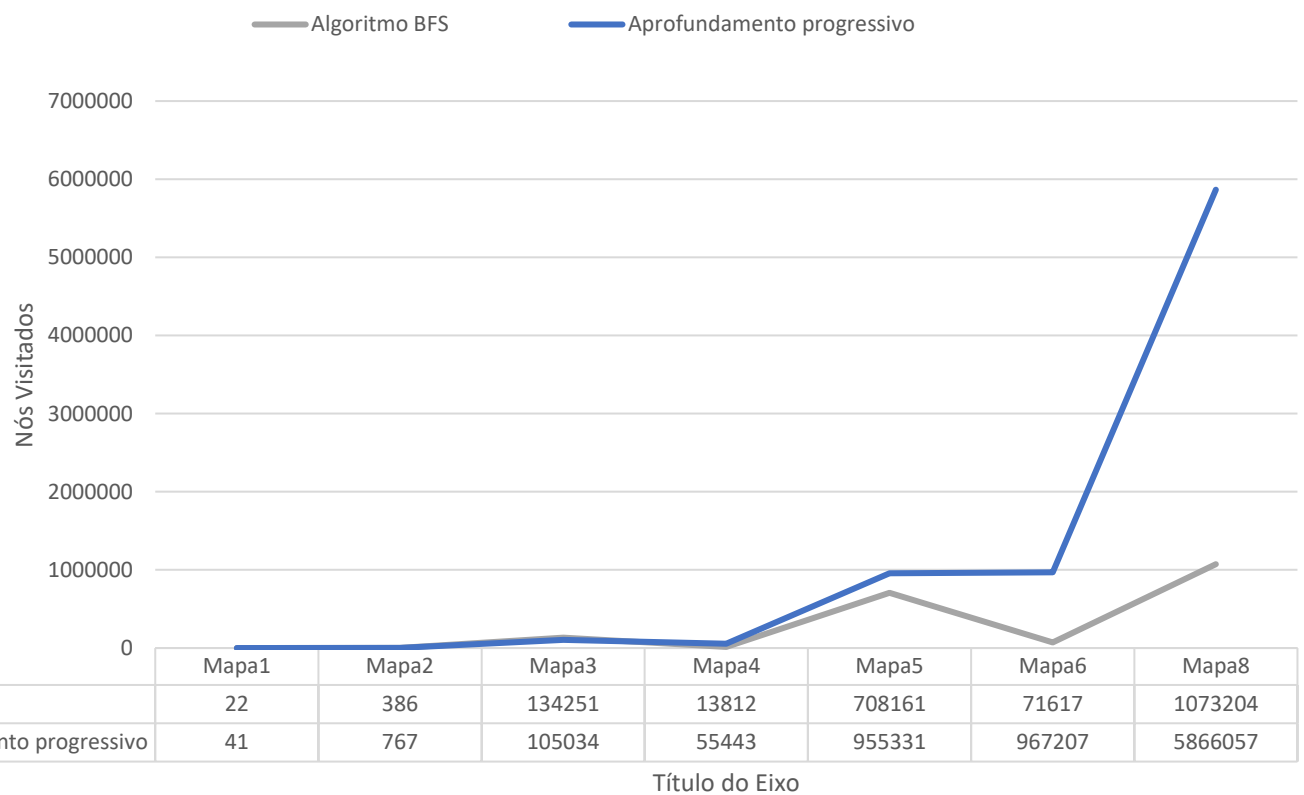
Algoritmo de aprofundamento progressivo

	Visitados	Expandidos	Ações	Tamanho Stack	Solução
Mapa1	41	116	5	4	S
Mapa2	767	2301	10	15	S
Mapa3	105034	281586	35	32	S
Mapa4	55443	141692	60	41	S
Mapa5	955331	2748378	56	54	S
Mapa6	967207	2070370	213	68	S
Mapa8	5866057	15681530	117	73	S

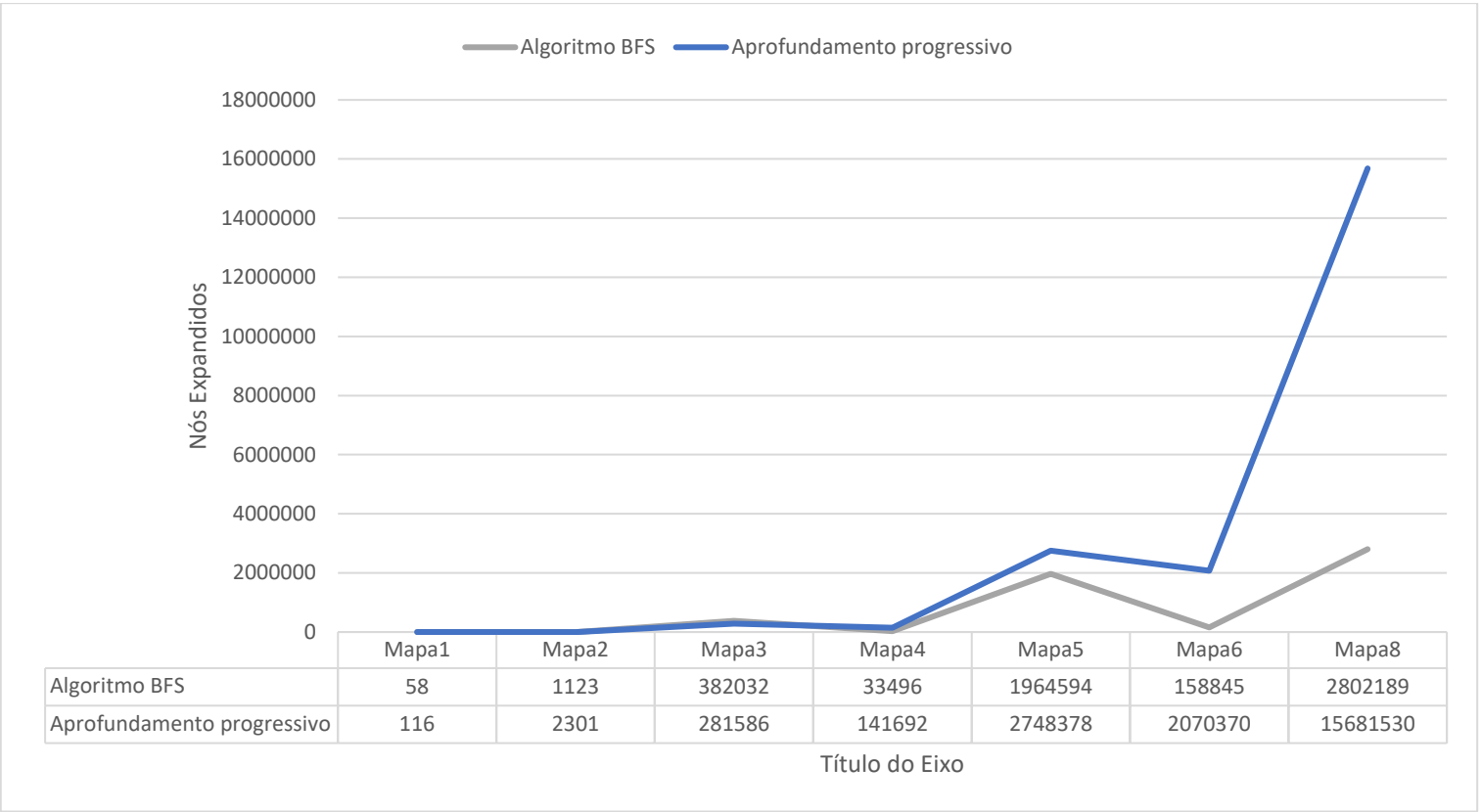
O algoritmo de pesquisa de aprofundamento progressivo encontra as soluções em tempo útil, ao contrário do algoritmo anterior. Em relação ao BFS nota-se uma melhoria em termos do uso da stack em relação à Queue, mas em relação aos nós visitados e expandidos tem uma complexidade elevada.

BFS vs Aprofundamento Progressivo

Nós Visitados



Nós Expandidos



Métodos de procura informada

Ao contrário dos métodos de procura cega, estes métodos assumem que têm informação que os “ajuda” no processo de pesquisa. Estes métodos de procura informada têm, para além do custo desde o estado inicial até ao estado alvo ($g(n)$), uma informação adicional que nos dá uma estimativa do custo do nó corrente até ao nó solução ($h(n)$).

Heurísticas

A função que nos dá uma estimativa do custo de um nó corrente até ao nó solução é chamada de heurística. Para este trabalho desenvolvemos 3 heurísticas para além da que nos é fornecida no enunciado, que também se encontra implementada.

Heurística 1

Esta heurística foi a recomendada no enunciado e é dada pelo número de caixas fora da posição final/objetivo. Para obter esta heurística é verificado o número de caixas total e depois é decrementado este número, quando as caixas se encontram na posição final num dado momento, sendo retornado esse valor.

Por exemplo caso exista um mapa com 5 caixas, e num dado momento estão 2 caixas já numa posição final, a função heurística irá conter o valor 3.

É uma heurística admissível pois o custo real será sempre superior ao custo obtido nesta heurística.

Heurística 2

A heurística da distancia da diagonal, resulta do calculo da distancia na diagonal do bloco ao seu objetivo. Esta heurística é admissível, sendo comprovado nos resultados experimentais apresentados.

Heurística 3

Heurística da distancia de manhattan, para qualquer bloco, é o resultado da soma da distancia da posição inicial a posição da solução do bloco. Para este resultado admite-se para qualquer movimento, a heurística é admissível. Assim sendo o objectivo desta heurística será deslocar para a posição mais próxima do objectivo do problema.

Heurística 4

Esta heurística é a soma da distância euclidiana entre cada caixa e a posição final mais próxima. Para implementar esta heurística são percorridas todas as caixas, e é calculado e somada para cada caixa a distância euclidiana a posição objetivo mais próxima. Caso tenhamos 2 caixas em que a primeira está a distância euclidiana 3 e 5 dos objetivos, e a segunda a 6 e 2, a função heurística irá conter a soma de 3 com 2.

Esta heurística é também admissível, facto que é também comprovado pelos testes feitos.

Algoritmo de pesquisa Sôfrega

O algoritmo de procura sôfrega funciona, escolhendo o nó da fronteira de uma árvore de procura, sendo o nó mais promissor estipulado pelo resultado da função de $h(n)$. Este algoritmo deixa a árvore de procura ordenada, escolhendo o valor mais baixo do nó escolhido, estando mais próximo hipoteticamente mais próximo da solução do problema.

Sôfrega Heurística 1

	Visitados	Expandidos	Ações	Tamanho Lista	Solução
Mapa1	19	50	5	9	S
Mapa2	30	96	8	34	S
Mapa3	3506	9568	37	350	S
Mapa4	9099	21967	52	500	S
Mapa5	1767	4641	39	386	S
Mapa6	9542	21360	89	767	S
Mapa8	28793	73864	97	3128	S

Sôfrega Heurística 2

	Visitados	Expandidos	Ações	Tamanho Lista	Solução
Mapa1	19	50	5	9	S
Mapa2	24	77	10	28	S
Mapa3	3467	9230	35	163	S
Mapa4	4834	11641	66	152	S
Mapa5	31861	84477	47	1559	S
Mapa6	4826	10513	123	113	S
Mapa8	6794	17741	131	183	S

Sôfrega Heurística 3

	Visitados	Expandidos	Ações	Tamanho Lista	Solução
Mapa1	19	50	5	9	S
Mapa2	24	77	10	28	S
Mapa3	3382	9004	35	163	S
Mapa4	2099	5087	66	93	S
Mapa5	31861	84477	47	1559	S
Mapa6	4753	10386	121	165	S
Mapa8	6615	17217	131	221	S

Sôfrega Heurística 4

	Visitados	Expandidos	Ações	Tamanho Lista	Solução
Mapa1	19	50	5	9	S
Mapa2	24	77	10	28	S
Mapa3	3467	9230	35	163	S
Mapa4	4834	11641	66	152	S
Mapa5	31861	84477	47	1559	S
Mapa6	4826	10513	121	113	S
Mapa8	6794	17741	131	183	S

Analisando os resultados a cima descritos, verificamos resultados interessantes. Para o mesmo Algoritmo, e com heurísticas diferentes, encontramos, que temos uma heurística que expandido e visitando mais nós, encontra uma solução ótima, fazendo a solução do problema em menos movimentos. Por outro lado, encontramos uma heurística que expandindo menos a arvore e apresenta uma solução, mas fazendo mais movimentos no cenário do jogo

*Algoritmo de pesquisa A**

O algoritmo A* e um algoritmo que combina o algoritmo de procura Sôfrega com o custo uniforme. A função de ativação tem o formato $f(n) = g(n) + h(n)$, onde $g(n)$ e o custo uniforme entre o estado inicial ao estado do nó n, e $h(n)$ e o valor da heurística aplicada no algoritmo.

A Heurística 1*

	Visitados	Expandidos	Ações	Tamanho Lista	Solução
Mapa1	21	35	5	9	S
Mapa2	187	564	8	139	S
Mapa3	126745	362926	29	16900	S
Mapa4	13126	31822	50	697	S
Mapa5	442260	1234389	35	87813	S
Mapa6	71123	157717	89	3009	S
Mapa8	966704	2523898	93	34455	S

A Heurística 2*

	Visitados	Expandidos	Ações	Tamanho Lista	Solução
Mapa1	21	55	8	9	S
Mapa2	128	386	8	90	S
Mapa3	123701	354202	29	18268	S
Mapa4	10054	24540	50	738	S
Mapa5	107845	302854	35	32485	S
Mapa6	54556	120524	89	2692	S
Mapa8	966704	2523898	93	34455	S

A Heurística 3*

	Visitados	Expandidos	Ações	Tamanho Lista	Solução
Mapa1	21	55	5	9	S
Mapa2	119	357	8	79	S
Mapa3	120261	344846	29	17576	S
Mapa4	9254	22585	50	689	S
Mapa5	107845	302854	35	32485	S
Mapa6	54178	119738	89	2842	S
Mapa8	785801	2047037	93	27065	S

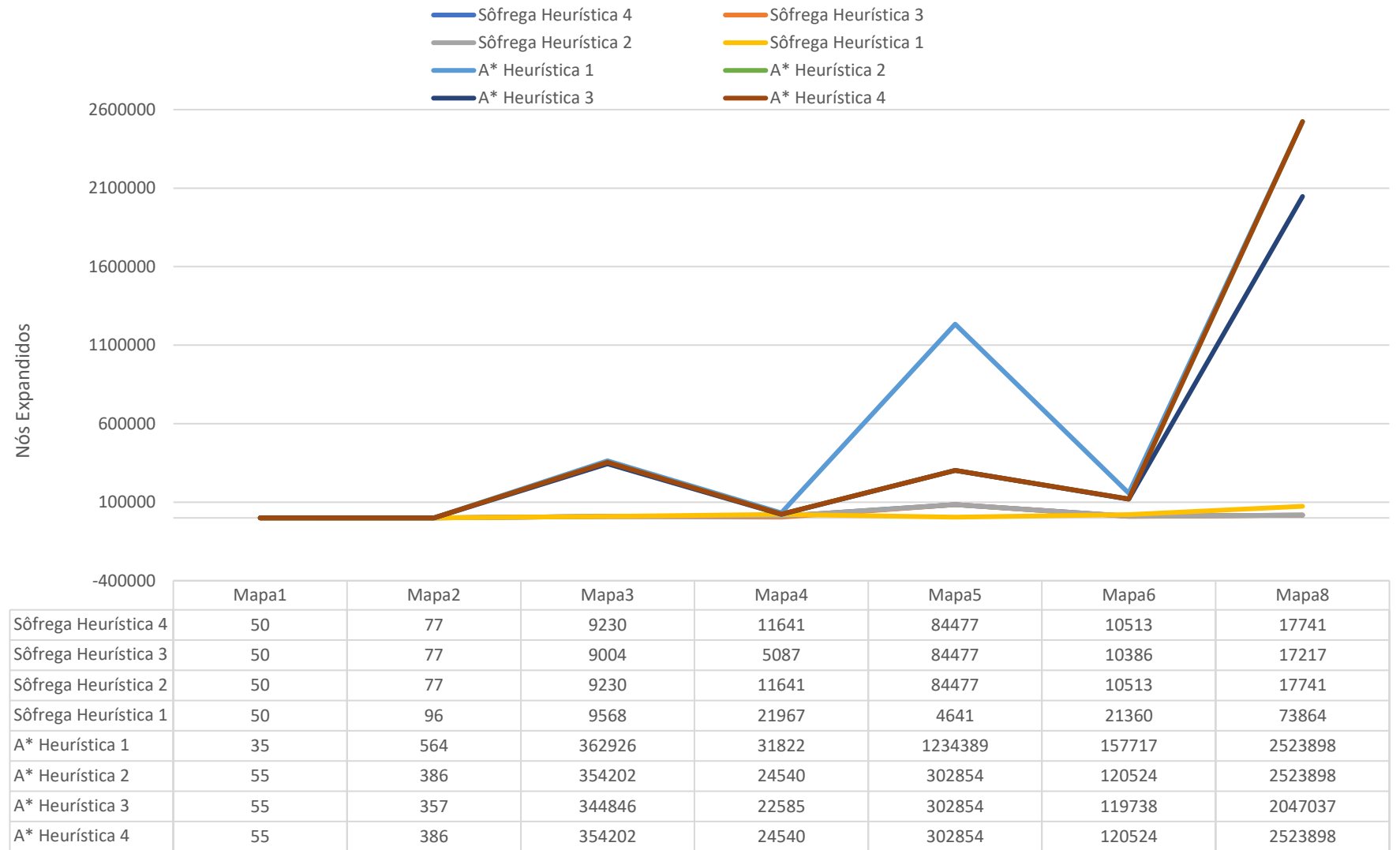
A Heurística 4*

	Visitados	Expandidos	Ações	Tamanho Lista	Solução
Mapa1	21	55	8	9	S
Mapa2	128	386	8	90	S
Mapa3	123701	354202	29	18268	S
Mapa4	10054	24540	50	738	S
Mapa5	107845	302854	35	32485	S
Mapa6	54556	120524	89	2692	S
Mapa8	966704	2523898	93	34455	S

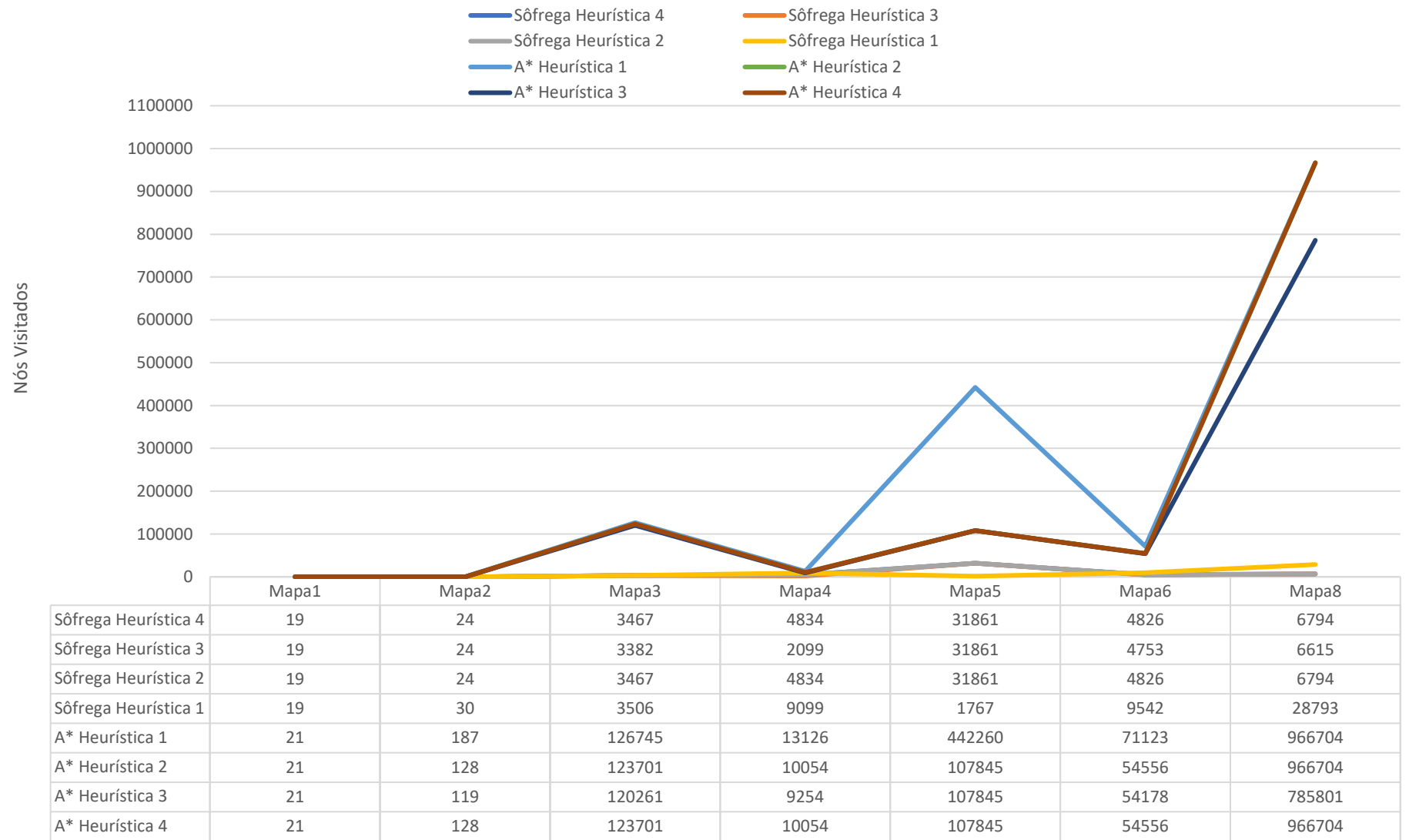
Após feitas, varias experiencias com este algoritmo, e com as heurísticas que foram implementadas, verifica-se que só a heurística 3, não expande a arvore de procura e não visita o mesmo numero de nós que as outras 3 heurísticas, apresentando uma solução com o mesmo numero de movimentos que as outras 3 heurísticas, usadas. Sendo uma heurística admissível para o problema apresentado, não desperdiçando memoria a expandir a árvore.

A* vs Sôfrega

Nós Expandidos



Nós Visitados



Conclusão

Os algoritmos estudados apresentam sempre uma solução com sucesso. Destes algoritmos alguns são discriminadores, como é o caso do algoritmo em largura primeiro, do progressivo e do A* (que só é completo se a heurística for admissível).

Tivemos dificuldades em termos de performances dos algoritmos, o que implicou que o grupo não conseguiu recolher os resultados para todos os mapas disponibilizados. Como é o caso do A* que encontra sempre o melhor caminho, se a heurística for admissível, mas tem um custo computacional muito elevado. Enquanto o algoritmo de pesquisa sôfrega pode não encontrar a melhor solução, mas em termos de complexidade computacional é mais eficiente.

O que nos leva a concluir que a escolha dos algoritmos depende do problema, e para que tipo de uso ou recolha de dados que se necessita. Por isso, a escolha do algoritmo depende do tipo de problema que temos em mão e da solução que pretendemos.