

Relatório 1º Projecto de IA

Grupo 2

Resultado dos testes:

teste	medida	DFS	Greedy	A*
4x5 com 2 cores sem solução	gerados	0	0	0
	expandidos	1	1	1
	tempo(s)	0	0,0005	0
	Solução?	não	não	não
4x5 com 3 cores	gerados	6	6	6
	expandidos	3	3	3
	tempo(s)	0.0001	0.0009	0.0015
	Solução?	sim	sim	sim
10x4 com 3 cores sem solução	gerados	74701	74701	74701
	expandidos	74702	74702	74702
	tempo(s)	19.95	27.68	27.38
	Solução?	não	não	não
10x4 com 3 cores	gerados	205	59	59
	expandidos	181	42	42
	tempo(s)	0.06	0.02	0.02
	Solução?	sim	sim	sim
10x4 com 5 cores	gerados	404	115	97
	expandidos	346	58	36
	tempo(s)	0.11	0.04	0.03
	Solução?	sim	sim	sim

Nota: o primeiro nó é contado como expandido mas não como gerado por isso em casos limites (sem solução ou apenas um nó existente) pode haver mais um expandido do que gerados.

Completude

- Nos exemplos acima os três algoritmos são completos, uma vez que encontraram sempre a solução dos problemas com solução.
- Apesar da DFS e o algoritmo Greedy não serem completos por poderem entrar em ciclos, como os problemas testados e qualquer tipo de jogo "same_game" não geram árvores de procura infinitas (pois não há maneira de passar para um estado anterior) a solução é sempre encontrada portanto consideramos como completos neste problema.

Eficiência:

- No pior caso (sem solução) os algoritmos expandem todos os nós.
- Para tabuleiros de dimensões reduzidas, o número de nós expandidos é semelhante a todos os algoritmos.
- Para tabuleiros de dimensão superior, o número de **nós** expandidos para a **DFS** é **superior** ao número de nós expandidos para os algoritmos **Greedy** e **A***.

Para todos os exemplos, nenhum algoritmo expandiu menos nós que o algoritmo A*, o que confirma a propriedade do algoritmo A* (A* é optimalmente eficiente).

Heurística $h(n)$:

- **$h(n)$** = número de grupos (prioriza tabuleiros com menor grupos de cores).
- Heurística **não admissível** (otimista) para este problema pois é NP-Completo.
- Como alternativa é usada uma heurística com **erro reduzido**, que consequentemente reduz o tempo de execução do algoritmo.
- Observando os resultados obtidos para o número de nós expandidos para o Greedy e para o **A***, podemos concluir que a **heurística** escolhida é **boa**.

No algoritmo Greedy, podem ser expandidos estados cujo custo do caminho do estado inicial até ao estado é já bastante elevado (problema que pode ser reduzido com outra heurística).

Impacto do número de cores e dimensões do tabuleiro:

- O aumento de número de grupos aumenta o número de estados/nós expandidos, aumentando assim o espaço e tempo de execução dos algoritmos.
- As dimensões do tabuleiro têm um maior impacto no número de nós expandidos do que o número de cores.
- O número de cores aumenta o número mínimo de nós a expandir, mais significativamente em tabuleiros de pequenas dimensões.

Comparação: ($h(n)$ = número de grupos e $g(n)$ - custo do nó inicial até n)

A*:

- Função de avaliação: **$f(n) = g(n) + h(n)$**
- **Menor** número de **nós** expandidos (optimalmente eficiente).
- **Evita** expandir nós que têm um **custo** muito elevado ($g(n)$).
- Tempo de execução sempre **semelhante** ou **inferior** ao algoritmo **Greedy** e **DFS**.

Greedy:

- Função de avaliação: **$f(n) = h(n)$** . (Considera apenas a custo do caminho do nó inicial até um nó terminal).
- Menor tempo de execução e profundidade da árvore de procura inferior à gerada pelo algoritmo DFS, para problemas grandes o suficiente.

DFS:

- Função de avaliação **$f(n) = g(n)$** . (Considera apenas o custo do caminho desde o estado inicial até ao nó n).