

Relatório de Resultados – Busca de Rotas em Grafos (5x5)

1. Introdução

O objetivo do experimento foi avaliar algoritmos clássicos de busca de rotas em um grafo 5x5 com pesos e obstáculos. Foram testados:

- **BFS**
- **DFS**
- **Dijkstra**
- **Greedy Best-First Search (GBFS)** com heurísticas Manhattan e Euclidiana
- **A*** com heurísticas Manhattan e Euclidiana

O ponto de partida foi $(0, 0)$ e o destino $(4, 4)$.

2. Resultados Obtidos

Algoritmo	Heurística	Caminho	Custo	Nós Expandidos	Tempo (ms)
A*	Euclidiana	9 nós	18	13	0
A*	Manhattan	9 nós	18	13	0
BFS	–	9 nós	18	13	1
DFS	–	9 nós	18	9	0
Dijkstra	–	9 nós	18	13	1
GBFS	Euclidiana	9 nós	18	9	0
GBFS	Manhattan	9 nós	18	9	0

3. Análise de Heurísticas

- Para **A***, o uso de heurística (Manhattan ou Euclidiana) não alterou o caminho nem o custo, mas manteve o número de nós expandidos eficiente (13 nós).
 - Para **GBFS**, a heurística direcionou a busca de forma eficiente, reduzindo o número de nós expandidos para 9.
 - Conclusão: **Heurísticas foram determinantes para eficiência**, mas o caminho ótimo foi o mesmo neste grafo pequeno.
-

4. Comparação de Performance

- **DFS** expandiu menos nós (9) que BFS e Dijkstra (13), mas depende fortemente da ordem de exploração, podendo não ser confiável em grafos maiores ou mais complexos.
 - **GBFS** foi o algoritmo mais rápido (0 ms) e com menor expansão de nós graças à heurística.
 - **A*** combinou custo mínimo e heurística, garantindo eficiência sem perder a precisão do caminho.
 - **BFS e Dijkstra** tiveram desempenho semelhante neste grafo, expandindo 13 nós e custando 1 ms.
-

5. Impacto do Tamanho do Grafo

- Em grafos maiores, DFS pode explorar caminhos inúteis, aumentando drasticamente nós expandidos e tempo.
 - BFS e Dijkstra escalariam linearmente com o aumento do número de nós.
 - Algoritmos heurísticos (GBFS e A*) mantêm expansão de nós menor, mostrando vantagem em grafos grandes.
-

6. Observações e Padrões

- Todos os algoritmos encontraram o mesmo **custo total do caminho (18)**.
 - Heurísticas reduzem significativamente nós expandidos e tempo de execução, sem alterar o custo em grafos simples.
 - O número de nós visitados por DFS é menor neste caso, mas o algoritmo é menos previsível em cenários maiores ou mais complexos.
-

7. Conclusão

- **Heurísticas foram determinantes** para eficiência em GBFS e A*.
- **A*** é o algoritmo mais balanceado, garantindo caminho ótimo e bom desempenho.
- Em grafos pequenos, todos os algoritmos encontraram o mesmo caminho.
- Em grafos maiores, algoritmos com heurística serão mais escaláveis.