|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| NColxLin | Nºnos | Nº lig | Soluçao 1º Prof | Custo  Tempo | Soluçao 1º Larg | Custo  Tempo | Soluçao A\* | Custo  Tempo | Soluçao  Better 1ºProf | Custo  Tempo |
| 3x3 | 9 | 24 | [cel(1, 1), cel(2, 1), cel(3, 1), cel(3, 2), cel(2, 2), cel(1, 2), cel(1, 3), cel(2, 3), cel(3,3)] | Custo:  9  Tempo:  0.00016 | [cel(1, 1), cel(2, 1), cel(3, 1), cel(3, 2), cel(3, 3)] | Custo:  5  Tempo:  7.7e-5 | [cel(1, 1), cel(2, 2), cel(3, 3)] | Custo:  2.83  Tempo:  6.10e-5 | [cel(1, 1), cel(2, 1), cel(3, 1), cel(3, 2), cel(3, 3)] | Custo:  5  Tempo:  0.0002 |
| 4x4 | 16 | 48 | [cel(1,1), cel(2,1), cel( 3,1), cel(4,1), cel(4,2), cel(3,2), cel(2,2), cel(1 ,2), cel(1,3), cel(2,3), cel(3,3), cel(4,3), cel(4, 4)) ] | Custo:  13  Tempo:  0.00027 | [cel(1, 1), cel(2, 1), cel(3, 1), cel(4, 1), cel(4, 2), cel(4, 3), cel(4, 4)] | Custo:  7  Tempo:  0.0011 | [cel(1, 1), cel(2, 2), cel(3, 3), cel(4, 4)] | Custo:  4.24  Tempo:  0.00016 | [cel(1, 1), cel(2, 1), cel(3, 1), cel(4, 1), cel(4, 2), cel(4, 3), cel(4, 4)] | Custo:  7  Tempo:  0.0055 |
| 5x5 | 25 | 80 | [cel(1, 1), cel(2, 1), cel(3, 1), cel(4, 1), cel(5, 1), cel(5, 2), cel(4, 2), cel(3, 2), cel(2, 2), cel(1, 2), cel(1, 3), cel(2, 3), cel(3, 3), cel(4, 3), cel(5, 3), cel(5, 4), cel(4, 4), cel(3, 4), cel(2, 4), cel(1, 4), cel(1, 5), cel(2, 5), cel(3, 5), cel(4, 5), cel(5, 5)] | Custo:  25  Tempo:  0.00020 | [cel(1, 1), cel(2, 1), cel(3, 1), cel(4, 1), cel(5, 1), cel(5, 2), cel(5, 3), cel(5, 4), cel(5, 5)] | Custo:  9  Tempo:  0.008 | [cel(1, 1), cel(2, 2), cel(3, 3), cel(4, 4), cel(5, 5)] | Custo:  5.66  Tempo:  0.00065 | [cel(1, 1), cel(2, 1), cel(3, 1), cel(4, 1), cel(5, 1), cel(5, 2), cel(5, 3), cel(5, 4), cel(5, 5)] | Custo:  9  Tempo:  0.228 |
| 3x4 | 12 | 34 | [cel(1, 1), cel(2, 1), cel(3, 1), cel(3, 2), cel(2, 2), cel(1, 2), cel(1, 3), cel(2, 3), cel(3, 3), cel(3, 4)] | Custo:  10  Tempo:  0.0002 | [cel(1, 1), cel(2, 1), cel(3, 1), cel(3, 2), cel(3, 3), cel(3, 4)] | Custo:  6  Tempo:  0.00028 | [cel(1, 1), cel(2, 2), cel(2, 3), cel(3, 4)] | Custo:  3.83  Tempo:  0.00025 | [cel(1, 1), cel(2, 1), cel(3, 1), cel(3, 2), cel(3, 3), cel(3, 4)] | Custo:  6  Tempo:  0.00067 |
| 3x5 | 15 | 44 | [cel(1, 1), cel(2, 1), cel(3, 1), cel(3, 2), cel(2, 2), cel(1, 2), cel(1, 3), cel(2, 3), cel(3, 3), cel(3, 4), cel(2, 4), cel(1, 4), cel(1, 5), cel(2, 5), cel(3, 5)] | Custo:  15  Tempo:  0.0002 | [cel(1, 1), cel(2, 1), cel(3, 1), cel(3, 2), cel(3, 3), cel(3, 4), cel(3, 5)] | Custo:  7  Tempo:  0.00062 | [cel(1, 1), cel(1, 2), cel(1, 3), cel(2, 4), cel(3, 5)] | Custo:  4.82  Tempo:  0.00067 | [cel(1, 1), cel(2, 1), cel(3, 1), cel(3, 2), cel(3, 3), cel(3, 4), cel(3, 5)] | Custo:  7  Tempo:  0.00185 |
| 4x5 | 20 | 62 | [cel(1, 1), cel(2, 1), cel(3, 1), cel(4, 1), cel(4, 2), cel(3, 2), cel(2, 2), cel(1, 2), cel(1, 3), cel(2, 3), cel(3, 3), cel(4, 3), cel(4, 4), cel(3, 4), cel(2, 4), cel(1, 4), cel(1, 5), cel(2, 5), cel(3, 5), cel(4, 5)] | Custo:  20  Tempo:  0.00024 | [cel(1, 1), cel(2, 1), cel(3, 1), cel(4, 1), cel(4, 2), cel(4, 3), cel(4, 4), cel(4, 5)] | Custo:  8  Tempo:  0.0023 | [cel(1, 1), cel(1, 2), cel(2, 3), cel(3, 4), cel(4, 5)] | Custo:  5.24  Tempo:  0.00046 | [cel(1, 1), cel(2, 1), cel(3, 1), cel(4, 1), cel(4, 2), cel(4, 3), cel(4, 4), cel(4, 5) | Custo:  8  Tempo:  0.0197 |
| 4x3 | 12 | 34 | [cel(1, 1), cel(2, 1), cel(3, 1), cel(4, 1), cel(4, 2), cel(3, 2), cel(2, 2), cel(1, 2), cel(1, 3), cel(2, 3), cel(3, 3), cel(4, 3)] | Custo:  12  Tempo:  7.89e-5 | [cel(1, 1), cel(2, 1), cel(3, 1), cel(4, 1), cel(4, 2), cel(4, 3)] | Custo:  6  Tempo:  0.00017 | [cel(1, 1), cel(2, 2), cel(3, 2), cel(4, 3)] | Custo:  3.83  Tempo:  0.00026 | [cel(1, 1), cel(2, 1), cel(3, 1), cel(4, 1), cel(4, 2), cel(4, 3)] | Custo:  6  Tempo:  0.000781 |

Uma imagem com texto, diagrama, captura de ecrã, Tipo de letra

Descrição gerada automaticamente

Como podemos verificar nos gráficos de dimensão-custo, o custo tende a aumentar quantas mais são as colunas e linhas da matriz.

Podemos concluir que também que seja a matriz 4x3 ou 3x4, por exemplo não tende a afetar o custo. Apenas quando se usa o algoritmo dfs podemos verificar uma diferença de custo, isto acontece devido a matriz conter mais colunas, e como este algoritmo tenta percorrer primeiro na horizontal.

Uma imagem com texto, número, Tipo de letra, diagrama

Descrição gerada automaticamente

Nestes gráficos podemos ver a comparação entre a dimensão e o tempo de execução do algoritmo. Como podemos verificar o que mais é afetado pelo aumento da dimensão da matriz é o tempo de execução já que a quantidade de ligações aumenta em conjunto com a dimensão.

Não só é possível verificar que o algoritmo dfs é o menos afetado pelo aumento da dimensão, mas é igualmente possível afirmar que tanto o algoritmo better dfs e A\* são os mais afetados pelo aumento de dimensão, podemos concluir que ao aumentar ainda mais a dimensão, o tempo necessário para executar estes será cada vez mais exorbitante.

Depois de fazer vários testes descobrimos também que a partir de grafos com cerca de 50 nós, deixa de ser possível procurar o caminho, já que a tentativa de usar os algoritmos é parada devido a um stack overflow. O ultimo grafo que foi possível testar demoraou cerca de 8/9 minutos.