



UNIVERSIDADE DE SÃO PAULO
INSTITUTO DE CIÊNCIAS MATEMÁTICAS E DE
COMPUTAÇÃO
Disciplina SCC-0630 – Inteligência Artificial

Trabalho 2 - Análise de Métodos de Busca em Grafos

GABRIEL TAKESHI MIYAKE BATISTELLA Nº USP: 11232198

VÍTOR FERNANDO RINALDINI Nº USP: 11232305

Docente:

ALNEU DE ANDRADE LOPES

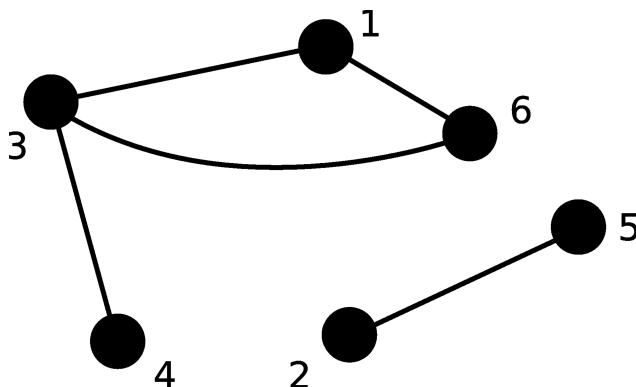
Julho de 2021

1 Introdução

Grafos são estruturas matemáticas formadas por um conjunto de vértices (ou nós) V que são ligados por um conjunto de arestas E (pares de nós), que podem ser orientadas ou não, e podem também ter pesos diferentes. Os vértices dos grafos utilizados em nossas análises são pontos cartesianos (coordenadas x e y), então podemos falar de fato da distância euclidiana entre vértices. Dito isso, existem vários tipos de grafos, dependendo de suas propriedades e de suas regras de geração, citaremos aqui o grafo KNN e o grafo Random, que serão explorados em nossas análises:

- Grafo KNN (K Nearest Neighbors) - cada vértice está ligado aos K vértices que estão mais próximos de si mesmo (vale ressaltar que é possível que um vértice tenha mais de K ligações pois ele pode ser um dos K vértices mais próximos de um vértice que não era um de seus K vértices mais próximos);
- Grafo Random - cada vértice está ligado a K outros vértices aleatórios (novamente é possível que um vértice tenha mais de K ligações, por motivo análogo ao do KNN).

Graficamente podemos visualizar os grafos como vários círculos, representando os vértices (cada um em sua respectiva posição no plano cartesiano no nosso caso), ligados entre si por segmentos de reta, representando as arestas.



Exemplo de representação visual de um grafo genérico

Nesse sentido, uma busca em um grafo consiste em encontrar um caminho (se existir) entre dois vértices do grafo, podendo apenas percorrer as arestas. Por exemplo, no grafo da figura acima, um caminho entre o vértice 4 e o vértice 1 seria o caminho $[4,3,6,1]$, e não existe caminho entre os vértices 5 e 3.

Definido o conceito de busca em grafos, podemos apresentar alguns métodos de realização dessa busca, que serão explorados em nossas análises:

- Busca em Profundidade - Procura um caminho estendendo um mesmo ramo do grafo até que não haja mais possibilidades nele (nesse caso é testado outra ramificação) - NÃO possui propriedades de melhor caminho;
- Busca em Largura por quantidade de nós (sem peso nas arestas) - Procura um caminho sempre estendendo os nós em um menor nível (distância, em número de nós), primeiro - Garante o caminho que percorre o MENOR NÚMERO DE NÓS;

- Busca em Largura por distância (com peso nas arestas) - Procura um caminho sempre estendendo os nós que percorreram uma menor distância euclidiana até o momento primeiro - Garante o caminho que percorre a MENOR DISTÂNCIA;
- Busca Heurística Best First - Procura um caminho sempre estendendo os nós que o algoritmo estima (a partir de uma heurística) serem mais próximos do destino primeiro, para assim tentar chegar ao destino mais rapidamente, reduzindo o tempo de execução. No nosso caso essa heurística é a distância euclidiana em linha reta do nó em questão ao nó final da busca - NÃO possui propriedades de melhor caminho para a nossa escolha de heurística;
- Busca Heurística com algoritmo A - É um tipo de Busca Best First, mas usa-se, como fator para escolha de qual nó estender primeiro, a soma: distância euclidiana percorrida até o momento + uma heurística pessimista que no nosso caso é 10x a distância euclidiana em linha reta do nó em questão ao nó final da busca - NÃO possui propriedades de melhor caminho (justamente por usar uma heurística pessimista), mas no geral acha um caminho bom;
- Busca Heurística com algoritmo A* - É um tipo de Busca Best First, mas usa-se, como fator para escolha de qual nó estender primeiro, a soma: distância euclidiana percorrida até o momento + uma heurística otimista que no nosso caso é a distância euclidiana em linha reta do nó em questão ao nó final da busca - Garante o caminho que percorre a MENOR DISTÂNCIA;

2 Objetivos e Breve Descrição dos Métodos

Este trabalho de implementação tem como objetivo testar o funcionamento dos diferentes algoritmos de busca aprendidos no curso de Inteligência Artificial. Para a análise, foram construídos grafos KNN e Random com diferentes parâmetros V (número de nós no grafo) e K (número mínimo de conexões por nó), para gerar cenários com diferentes tamanhos e complexidades. Foram então testados os vários algoritmos de busca para todos os cenários de grafos, e considerando o resultado médio de várias execuções para cada cenário, obtivemos o tempo de execução médio, a distância percorrida média (comprimento do caminho) e a quantidade média de nós no caminho, que utilizamos para comparar os métodos em cada cenário de V e K.

3 Implementação

Nós implementamos, em python, um programa que possibilita a geração de grafos KNN e Random com diferentes parâmetros V e K, além de realizar qualquer tipo de busca e obter o caminho encontrado por meio de uma animação (para testar os métodos) e os dados a respeito do tempo de execução dessa busca e número de nós/comprimento total do caminho encontrado. Comentaremos brevemente sobre a lógica por trás da implementação das principais funções do programa, para mais detalhes sobre o código em si, ele foi enviado (com vários comentários explicando cada função) juntamente com esse relatório no arquivo main.py.

3.1 Funções de Geração de Grafo

Utilizamos uma classe No, capaz de guardar a posição (coordenadas x e y) do nó instanciado, além de uma lista de outros objetos da classe No, representando quais nós estão ligados a esse nó. Então, para gerar os grafos, primeiramente adicionamos V objetos No a uma lista, cada um com uma posição aleatória dentro de um quadrado $V \times V$; depois descobrimos quais nós estarão ligados a cada nó da lista, considerando a regra do tipo de grafo escolhido (para KNN serão os K nós mais próximos, e para Random serão K nós aleatórios); por fim adicionamos os objetos No que estão ligados a cada nó específico à sua lista de conexões (trabalhando também para que se um determinado No2 tiver o No1 na sua lista de conexões, o No1 deverá ter também o No2 na sua, mas sem repetir duas vezes o No2) e a função retorna essa lista de nós.

3.2 Funções de Busca

Para todos os algoritmos de busca que implementamos, seguimos a mesma lógica: guardamos uma lista dos nós para serem testados (inicialmente essa lista tem apenas o nó inicial) e dos caminhos respectivos para chegar a cada um desses nós, e em cada iteração realizada seguimos as mesmas etapas, descritas a seguir.

Primeiramente testamos se o primeiro valor da lista é o nó de destino, se for, a busca acabou e retornamos o caminho, caso contrário retiramos esse primeiro nó da lista e adicionamos os nós com os quais ele se conecta (mas apenas se esse nó de conexão não estava no caminho dele) ao final da lista, ou ao início, no caso da profundidade; depois ordenamos os nós da lista (já com os novos nós inseridos) de acordo com um critério que depende do método, na verdade, no geral essa é a única diferença entre os métodos (para profundidade e largura(nós) não há necessidade de fazer a ordenação, já para a largura(dist), best first, A e A* ordenamos em função dos respectivos fatores de escolha de qual nó estender primeiro, já descritos na Introdução); por fim, retiramos os nós duplicados da lista de testes, mantendo os primeiros que aparecem (após o processo de ordenação), pois os caminhos através de um mesmo nó não precisam ser testados mais de uma vez, e mantendo os primeiros estamos garantindo que o algoritmo testará o caminho que ele considera o "melhor" dentre os que passam por aquele nó; após isso a lista está pronta para a próxima iteração.

3.3 Funções para Utilização

Implementamos também uma função para facilitar o teste dos métodos, que possui um menu que permite o usuário escolher o tipo de grafo que deseja gerar, além dos parâmetros V e K, e também permite a escolha os vértices inicial e final da busca, e a realização dessa busca com qualquer um dos 6 algoritmos implementados, por fim, o usuário obtém a resposta como uma animação do caminho sendo realizado pelas arestas do grafo (animação funciona bem no site replit, mas às vezes não em outras IDEs).

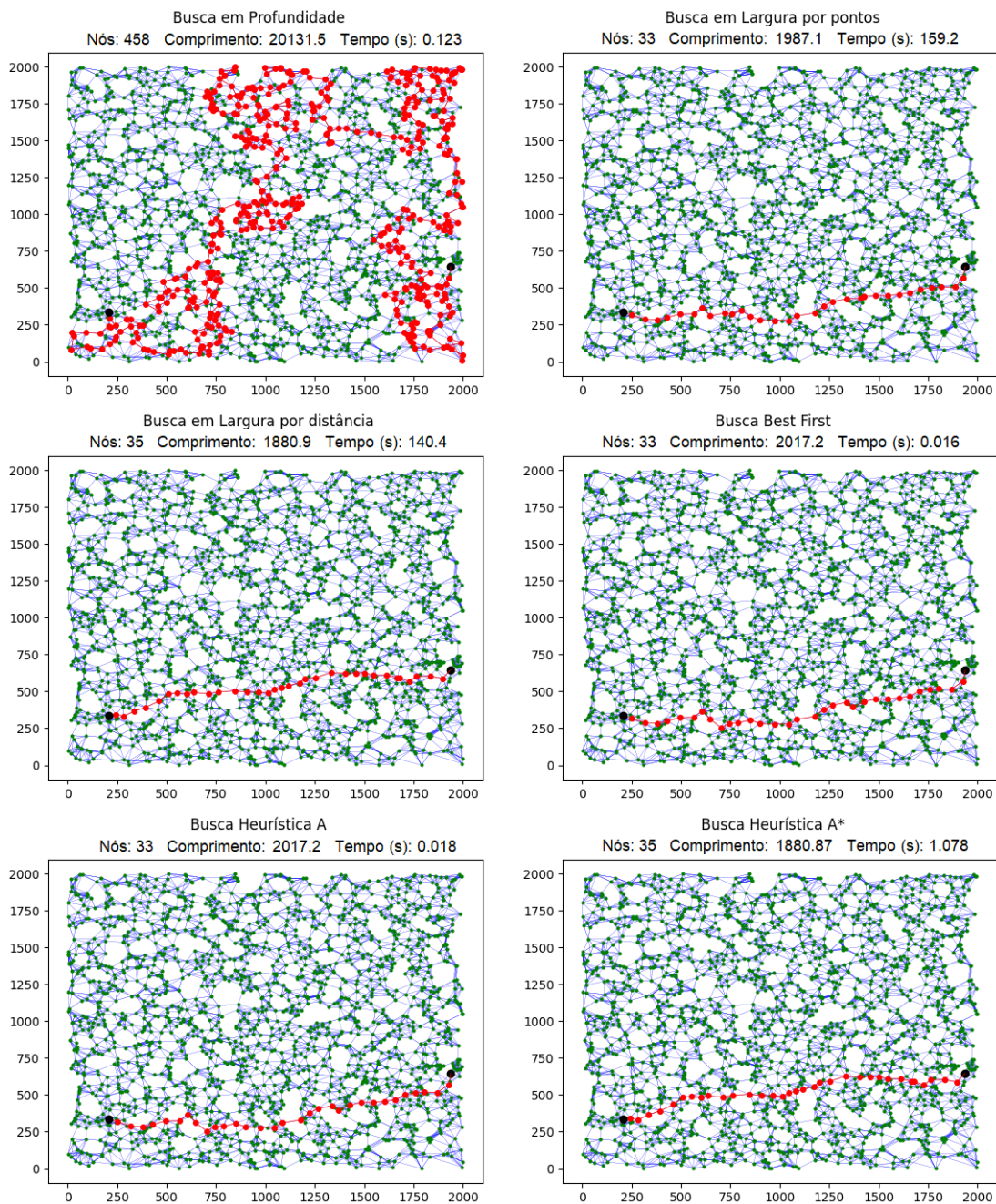
Outra função implementada foi uma função para comparar os métodos, ela gera N exemplos de grafos com o tipo, V e K especificados, escolhe nós aleatórios como inicial e final e então para cada grafo gerado ela realiza todos os tipos de busca, guardando em um arquivo os dados de tempo de execução, quantidade de nós no caminho e comprimento do caminho para cada execução de cada tipo de método, além do número de vezes que a busca falhou, por não haver um caminho (quando não há um caminho o algoritmo ignora o caso e gera outro grafo e outros nós inicial e final no lugar).

Essa função também salva imagens do primeiro grafo testado para os determinados V e K, sendo uma imagem para cada método, com o caminho encontrado pelo método indicado em vermelho e os nós inicial e final em preto.

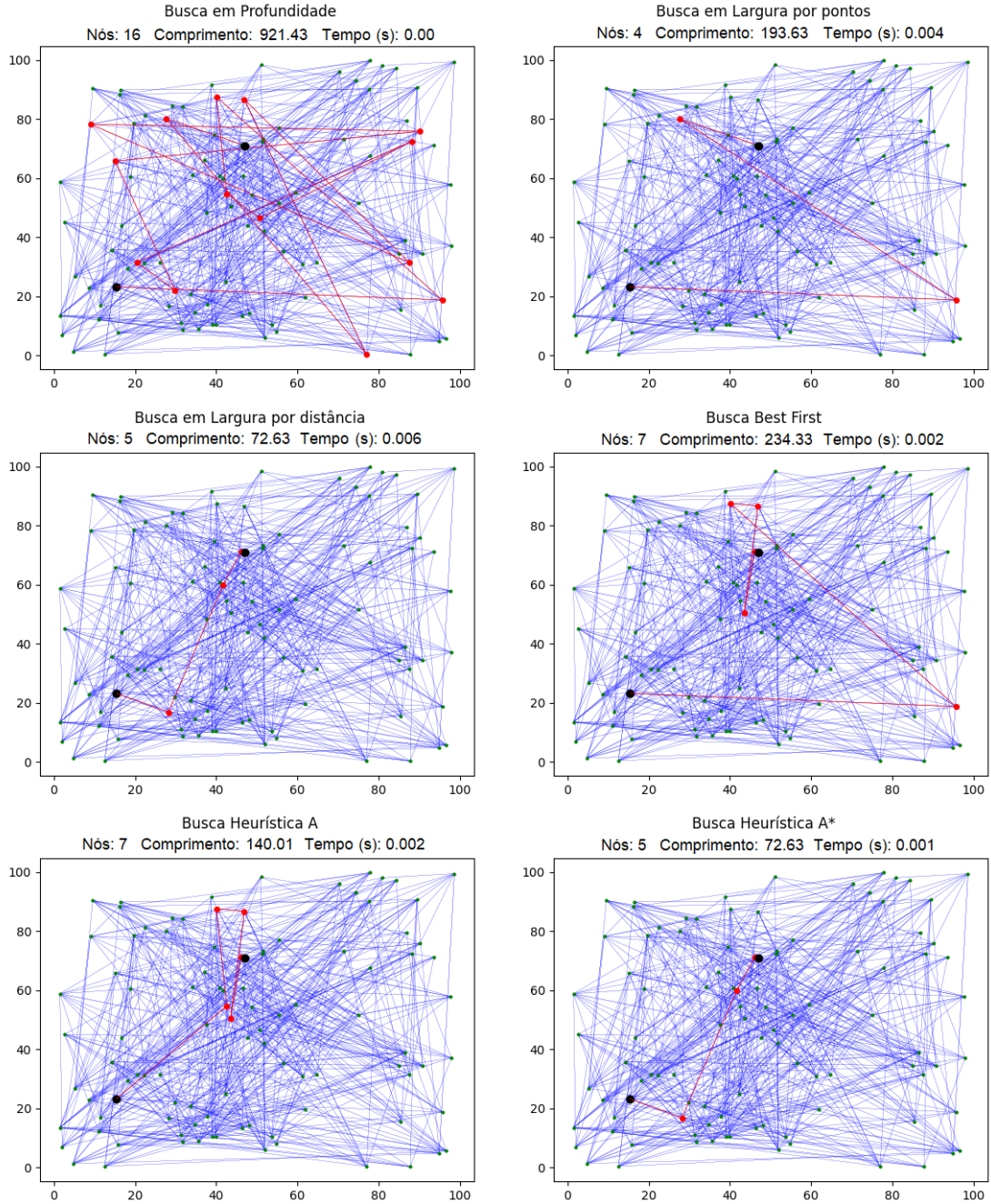
4 Execução

Utilizando as funções de utilização, já explicadas anteriormente, executamos o algoritmo com diversas combinações de V e K: [100, 500, 2000, 5000, 7000, 10000] e [3, 5, 7], respectivamente. Com isso, geramos vários exemplos e você pode conferir alguns desses resultados a seguir:

Grafos KNN - V=2000 e K=7



Grafos Random - V=100 e K=5



Outros exemplos estão disponíveis para visualização nas respectivas pastas dentro do .zip com o programa. A análise desses dados será realizada posteriormente.

5 Resultados e Discussão

5.1 Tratamento dos Dados

Para minimizar os efeitos de situações consideradas incomuns para um determinado método específico, decidimos executar a função de comparação com $N = 25$, formatando e salvando os dados em um arquivo de bloco de notas. Posteriormente, transferimos os

dados para um planilha do Google Sheets e fizemos os valores médios das incógnitas de interesse: quantidade de nós no caminho, comprimento do caminho percorrido e tempo de execução. Observação: A planilha citada anteriormente está acompanhando o relatório e pode ser consultada para mais detalhes.

Vale salientar que infelizmente devido ao tempo e a memória necessários para a sua execução, não foi possível a realização de muitos testes para os valores mais altos de V. Nesses casos, executamos apenas uma vez com o objetivo de realizar uma visualização, mas por serem específicos de uma situação, tais dados não foram levados em consideração durante a comparação.

Queremos observar também que os métodos de Largura necessitam de um tempo de execução e memória consideravelmente elevados para altos valores de V e por isso, não realizamos testes para esses métodos nesses casos.

Como última observação, ao utilizar $V = [500, 2000]$ no grafo Random, diminuimos o valor de N para 5. O motivo dessa mudança é o elevado tempo e memória necessários para a execução do método Best First, cujo problema especificamente no grafo Random será discutido mais adiante.

Por fim, colocamos esses valores médios em uma mesma tabela, separada para grafos KNN e Random, e geramos gráficos comparando os métodos de busca (pelos dados médios levantados) para $V = [100, 500, 2000]$ e $K = [3, 5, 7]$. Tanto as tabelas quanto os gráficos serão estudados mais adiante na discussão.

5.2 Interferência dos Valores de V e K no Grafo KNN

Tabela da comparação dos métodos de busca no grafo KNN

Método	V	K	Médias			Método	V	K	Médias		
			Nº de Nós	Comprimento	Tempo(s)				Nº de Nós	Comprimento	Tempo(s)
Profundidade	100	3	33,12	258,00916	0,00028	Best First	100	3	14,00	123,58949	0,00172
		5	39,40	362,12630	0,00056			5	8,12	90,03403	0,00079
		7	42,20	415,43388	0,00076			7	5,52	69,68717	0,00060
	500	3	116,12	2010,70657	0,00866		500	3	28,12	580,97207	0,05934
		5	155,60	3062,37635	0,01636			5	14,48	372,98726	0,00212
		7	161,16	3494,69667	0,01305			7	12,00	342,61707	0,00339
	2000	3	375,68	13126,48497	0,07322		2000	3	54,56	2317,07334	3,67461
		5	513,24	19872,41904	0,21476			5	29,32	1555,57476	0,00937
		7	561,40	24135,43025	0,36325			7	23,00	1398,24998	0,00962
	5000	3	725,12	40160,52380	0,53679		5000	3	52,16	3395,31827	0,94721
		5	1136,84	69601,79037	2,00355			5	28,16	2318,76055	0,01233
		7	1584,44	106823,34151	5,47344			7	21,84	2095,73906	0,01118
Largura (nós)	100	3	13,36	119,60925	0,00331	A	100	3	13,56	119,22141	0,00534
		5	7,92	88,69719	0,00613			5	8,08	88,85245	0,00107
		7	5,52	70,79793	0,00431			7	5,52	69,68717	0,00080
	500	3	24,76	505,79890	0,48432		500	3	27,56	540,26129	0,63393
		5	14,00	367,90508	0,40753			5	14,52	370,12265	0,00281
		7	11,56	347,69769	0,57947			7	12,04	340,61003	0,00324
	2000	3	47,36	2064,23358	85,12750		2000	3	53,12	2229,57833	11,33757
		5	28,04	1547,41481	91,86467			5	28,96	1532,16847	0,00894
		7	21,68	1398,46305	53,95623			7	22,92	1385,37620	0,01110
Largura (Distância)	100	3	13,56	116,89575	0,00502	A*	5000	3	50,24	3258,09832	3,06573
		5	8,36	84,19940	0,01744			5	27,80	2277,33190	0,03153
		7	5,76	66,73766	0,01477			7	21,88	2076,09718	0,01241
	500	3	25,60	484,22121	0,37134		100	3	13,56	117,03468	0,00411
		5	14,64	348,75861	0,57278			5	8,28	84,42916	0,00451
		7	12,12	320,30474	0,93260			7	5,72	66,76473	0,00096
	2000	3	48,76	1976,58358	42,25700		500	3	25,60	484,53946	0,11187
		5	30,04	1446,79457	82,19476			5	14,64	348,96014	0,02016
		7	23,72	1300,51418	61,91124			7	12,04	320,60937	0,03089
		3	48,76	1976,61156	4,62097		2000	3	48,76	1976,61156	4,62097
		5	30,04	1446,90035	0,64052			5	30,04	1446,90035	0,64052
		7	23,76	1300,62575	0,26224			7	23,76	1300,62575	0,26224
	5000	3	46,52	2937,31149	1,85921		5000	3	46,52	2937,31149	1,85921
		5	28,04	2094,21198	0,30364			5	28,04	2094,21198	0,30364
		7	22,64	1955,08841	0,23968			7	22,64	1955,08841	0,23968

Ao analisarmos a tabela e levando em consideração o mesmo método, conseguimos notar certas relações de proporcionalidade entre os valores de V e K com nossas incógnitas de interesse. Iremos analisá-las logo em seguida:

Ao manter o V constante:

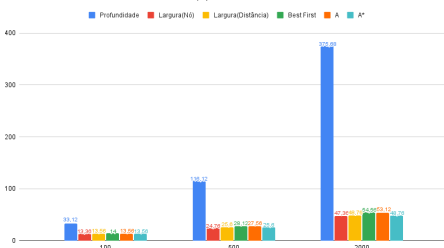
- O número de nós presentes no caminho diminui conforme aumentamos o K . Isso acontece porque ao aumentamos o K , aumentamos a quantidade de ligações, e com isso, as possibilidades existentes de seguir um caminho mais linear e com menos nós antes de chegar no destino. Repare que o método de profundidade é uma exceção a essa regra (para ele o número de nós aumenta), visto que seu comportamento é aleatório e pode se distanciar da resposta com mais facilidade graças a esse aumento, antes mesmo de encontrá-la;
- O comprimento do caminho segue a mesma lógica do números de nós, o que faz sentido, pois nos grafos KNN há uma forte dependência entre o comprimento do caminho e o número de nós nele;
- O tempo de execução para os métodos heurísticos (e para a profundidade) varia analogamente ao número de nós e comprimento, mas para os métodos em largura, ele varia de forma inconclusiva, pois ao aumentar K , por um lado aumentamos as possibilidades de seguir um caminho mais linear e chegar no destino mais rapidamente, mas por outro aumentamos o tempo levado para testar cada nível, pois o número de conexões por nó aumenta.

Ao manter o K constante:

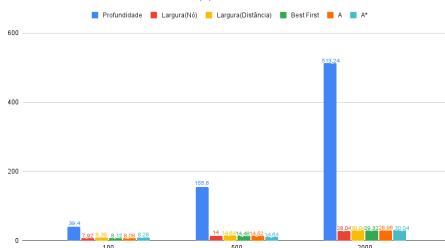
- O número de nós presente no caminho é diretamente proporcional à variação de V , uma vez que ao aumentarmos o valor de V , aumentaremos a distância euclidiana entre os pontos alvos, adicionando novos pontos entres eles;
- O comprimento do caminho percorrido também é diretamente proporcional à variação de V pelo mesmo motivo do número de nós;
- Teoricamente, o tempo de execução deveria variar também da mesma maneira: aumentar conforme aumentamos o valor de V , no entanto, para os métodos heurísticos, vemos uma irregularidade no caso $V=2000$ $K=3$, provavelmente graças a um caso específico muito difícil para os métodos, em que as estimativas não foram eficazes.

5.3 Comparação entre os Métodos no Grafo KNN

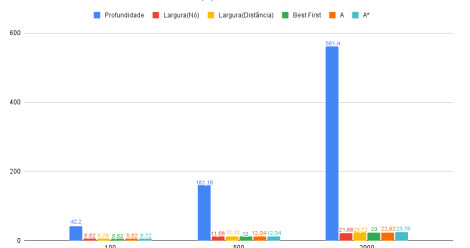
Quantidade média de nós no caminho de cada Método em função de V , para $K=3$



Quantidade média de nós no caminho de cada Método em função de V , para $K=5$



Quantidade média de nós no caminho de cada Método em função de V , para $K=7$





Da análise imediata dos gráficos acima, vemos mais claramente algumas coisas que já podíamos constatar pela tabela mostrada anteriormente:

- A quantidade de nós no caminho obtido pela busca em profundidade é muito maior que a dos caminhos obtidos por outros métodos, pois ele não possui qualquer propriedade de melhor caminho (geralmente é bem aleatório);
- Logicamente o mesmo comportamento da quantidade de nós se observa também no comprimento dos caminhos, pois para grafos KNN essas duas variáveis estão fortemente relacionadas;
- Os tempos de execução dos métodos em largura são muito maiores do que o dos outros métodos, pois exigem o teste de muitos nós.

Em uma análise mais minuciosa dos gráficos em conjunto com a tabela, podemos constatar também que:

- Como esperado, a quantidade média de nós no caminho encontrado pela busca em largura(nós) é sempre a menor em comparação com outros métodos, independente de V e de K, pois ele garante o caminho percorrendo o menor número de nós;
- Similarmente, o comprimento médio de caminho encontrado pela busca em largura(dist) é sempre igual ao encontrado pela busca A*, e são os menores quando comparados com outros métodos, independente de V e de K, pois eles garantem o caminho percorrendo a menor distância;
- Os métodos heurísticos levam um tempo consideravelmente menor para completar sua execução para casos com V grande (para V pequeno isso não é muito expressivo), pois o mecanismo de estimativa permite que eles testem muito menos nós para completar a busca.

5.4 Comparação com o Grafo Random

Tabela da comparação dos métodos de busca no grafo Random

Método	V	K	Médias			Método	V	K	Médias		
			Nº de Nós	Comprimento	Tempo(s)				Nº de Nós	Comprimento	Tempo(s)
Profundidade	100	3	42.92	2147.37383	0.00088	Best First	100	3	5.60	163.65570	0.01328
		5	52.20	2752.59627	0.00180			5	4.48	129.77880	0.00244
		7	51.72	2658.86157	0.00216			7	3.72	115.59653	0.00116
	500	3	174.20	46008.81427	0.04367		500	3	9.00	1115.03911	1.58439
		5	198.80	50924.19681	0.07362			5	9.20	1230.60427	33.09958
		7	138.60	34565.59967	0.06065			7	7.80	704.89334	1.39243
	2000	3	856.60	695011.63805	2.87562		2000	3	23.20	12155.52918	81.61189
		5	775.20	809342.83722	5.16398			5	14.40	6390.36472	0.18531
		7	756.40	796233.89841	6.80614			7	8.00	3827.22280	2.03794
Largura (nós)	100	3	3.60	150.94579	0.00128	A	100	3	4.72	127.91568	0.00523
		5	3.32	108.19864	0.00251			5	4.40	112.54253	0.00351
		7	3.12	134.01563	0.00295			7	3.72	103.67760	0.00141
	500	3	4.40	890.97834	0.07997		500	3	8.20	1020.38381	0.07161
		5	3.80	717.25172	0.11448			5	6.00	710.03837	1.00595
		7	3.80	672.67209	0.30142			7	6.20	602.32191	0.05646
	2000	3	5.00	3878.87458	9.53351		2000	3	10.20	5459.23729	3.14629
		5	5.20	3896.58707	28.75970			5	8.20	3591.07086	1.59009
		7	4.40	3429.24629	23.40691			7	8.60	3303.57541	1.61408
Largura (Distância)	100	3	4.00	100.40960	0.00403	A*	100	3	4.00	100.40960	0.00116
		5	3.72	85.63553	0.00736			5	3.72	85.63553	0.00143
		7	3.92	79.92499	0.01269			7	3.88	80.18754	0.00252
	500	3	5.00	649.88358	0.08797		500	3	5.00	649.88358	0.00600
		5	4.80	545.21109	0.44067			5	4.80	545.21109	0.04068
		7	4.00	389.94841	0.18153			7	4.00	389.94841	0.00679
	2000	3	6.00	3377.30920	11.24414		2000	3	6.00	3377.30920	1.78159
		5	6.00	2700.07603	24.56804			5	6.00	2700.07603	3.32183
		7	6.00	2120.10264	23.17991			7	6.00	2120.10264	0.82345

Podemos notar algumas diferenças em comparação com as buscas nos Grafos KNN:

- O numero de nós presentes nos caminhos dos Grafos Random é muito menor do que o dos Grafos KNN. Isso acontece pois, graças à aleatoriedade nas ligações dos Grafos Random, pode existir um ponto conectado diretamente ao destino independentemente da distância entre eles;
- Mesmo com o numero de nós menor, o comprimento do caminho geralmente é maior do que nos Grafos KNN. Isso acontece pois os grafos Random não possuem a relação entre comprimento e número de nós no caminho que os KNN possuem;
- Métodos Heurísticos não são tão bons para trabalhar nos Grafos Random, embora no geral ainda sejam melhores que os largura. Como exemplo, citaremos o método Best First que possui o caso com maior tempo de execução. Isso acontece porque nem sempre o ponto mais próximo do destino está necessariamente conectado com ele, então as estimativas (heurísticas) não são sempre muito eficazes;
- Ao ser sorteado pontos alvos aleatórios nos Grafos Random, a probabilidade de não existir um caminho é bem menor se comparado com Grafos KNN (nas execuções para os grafos Random, encontramos caminho em todas, enquanto para os grafos KNN houveram 3 casos com K pequeno em que não encontramos).

6 Conclusão

Dessa forma, concluímos a análise do desempenho (medido através das variáveis: tempo de execução, quantidade de nós no caminho encontrado e comprimento do caminho encontrado) dos métodos de busca para diferentes cenários: variamos o tamanho dos grafos, controlando os parâmetros V e K; variamos o método de busca utilizado; e variamos também o tipo de grafo gerado.