

Análise de algoritmos de busca em Labirintos

- *Victor Andrietta Razoli Silva - Ciência da computação*
- *vitorandrietta@gmail.com*

Resumo

Os algoritmos de buscas são de suma importância para a diversas áreas da computação (redes, sistemas distribuídos etc), portanto é vital entender suas especificidades para saber em qual situação cada um se encaixa melhor.

Nesse documento foi feita análise de 2 buscas não informadas (DFS e BFS) e uma busca informada (A*) com duas heurísticas diferentes (distância euclidiana da chegada e distância euclidiana da chegada ao quadrado) em um labirinto bidimensional (60x60) com paredes (intransponíveis pelo agente) .

A solução foi discretizada de forma a utilizar um ambiente de labirinto que não possua configuração fixa em relação à disposição de paredes, visto que isso pode afetar diretamente o processo de busca, para maior generalidade da análise. O Agente definido pode se mover nas 4 posições adjacentes a sua localização atual (cima,baixo,direita e esquerda) , desde que não passe a borda do labirinto e que não tenha parede nas mesmas.

O resultado final aponta que o A* utilizando a distância euclidiana ao quadrado mostra-se melhor no caso médio , o que é mais notável ainda quando aumentamos a concentração de paredes do labirinto atual.

Trabalho Proposto

Modelagem do agente

O labirinto foi modelado (espaço de atuação do agente) como uma matriz 60x60 (lista de listas) que apresenta uma configuração de paredes , início e fim aleatórias, de modo a avaliar a performance do algoritmo no geral e não ter a falsa impressão com base um ambiente pontual.

A parede é definida como um valor numérico, assim como espaços não visitados e já visitados. A chegada e partidas são duas coordenadas bidimensionais (x,y)

O agente não sabe onde está a parede, a não ser quando ele está adjacente à ela, portanto o ambiente não é completamente conhecido pelo agente, apenas o ponto de chegada e o de partida.

O problema foi modelado de modo que o estado inicial é o ponto de partida e o estado final é a chegada. O labirinto é um parâmetro do objeto responsável por tratar o problema, o qual irá verificá-lo a cada tomada de decisão de ação, escolhendo ir em 1 das 4 direções possíveis.

Análise de dados

O levantamento dos dados para análise da diferença entre os métodos foi feita usando configurações aleatórias de labirinto com quantidade variada de paredes. Com os dados de 12 configurações diferentes e aleatórias foram confeccionadas tabelas para analisar os algoritmos em termos de número de posições visitadas, densidade de parede (**porcentagem de posições que são paredes**), tempo para chegar ao objetivo e distância até o objetivo.

Para explicitar melhor a relação entre as métricas utilizadas nas tabelas, foram feitos gráficos 3D (**numero de nós visitados, tempo para solução e densidade de parede**) utilizando 400 configurações diferentes de mapa , cada uma delas foi testada em cada algoritmo para confecção do gráfico

A partir desses dados e utilizando o modelo teórico estabelecido pela referência [\[b1\]](#) e [\[b2\]](#) conclusões abordadas nas seções [\[Resultados e discussão\]](#) e [\[Conclusão\]](#) foram feitas

Tabelas e Gráficos

Tabelas para comparação de desempenho, cada linha da tabela é uma configuração aleatória de labirinto (ponto de partida e de chegada também são aleatórios), linhas de mesmo número entre tabelas de algoritmos distintos possuem configurações de labirinto idênticas (para facilitar a comparação) .

Legenda geral para as tabelas: wall density refere-se à porcentagem do labirinto que está preenchida com parede.

BFS

nodes visited	wall density	time to reach goal (s)	euclidian distance to goal
2620.0	0.3	0.010890483856201172	50.28916384272063
2150.0	0.4	0.008867979049682617	40.162171256046406
1756.0	0.1	0.0075261592864990234	39.20459156782532
1376.0	0.1	0.00593256950378418	35.73513677041127
1028.0	0.4	0.004117727279663086	34.655446902326915
954.0	0.3	0.003962039947509766	32.55764119219941
2608.0	0.3	0.02644038200378418	23.08679276123039
952.0	0.1	0.004514217376708984	21.18962010041709
522.0	0.2	0.0020890235900878906	20.09975124224178
422.0	0.4	0.0016286373138427734	18.384776310850235
286.0	0.2	0.0011785030364990234	9.0
128.0	0.2	0.0004978179931640625	8.246211251235321

DFS

nodes visited	wall density	time to reach goal (s)	euclidian distance to goal
2620.0	0.3	0.011008739471435547	50.28916384272063
2096.0	0.4	0.008179426193237305	40.162171256046406
2010.0	0.1	0.007126569747924805	39.20459156782532
3246.0	0.1	0.013571023941040039	35.73513677041127
624.0	0.4	0.0022051334381103516	34.655446902326915
1354.0	0.3	0.004588603973388672	32.55764119219941
2608.0	0.3	0.011014461517333984	23.08679276123039
653.0	0.1	0.001748800277709961	21.18962010041709
2406.0	0.2	0.007992267608642578	20.09975124224178
101.0	0.4	0.00031256675720214844	18.384776310850235
2194.0	0.2	0.006955146789550781	9.0
2324.0	0.2	0.007233619689941406	8.246211251235321

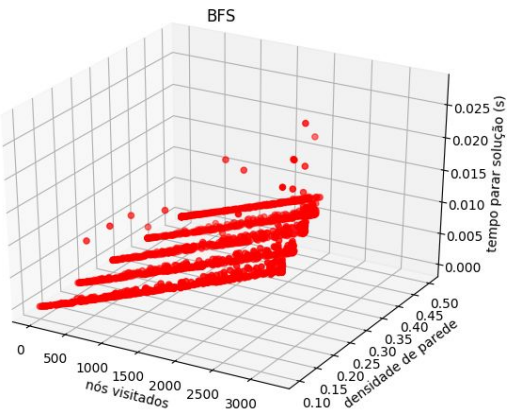
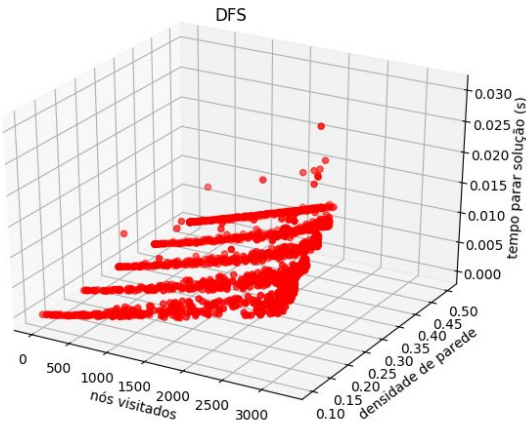
A* Heurística: Distância euclidiana

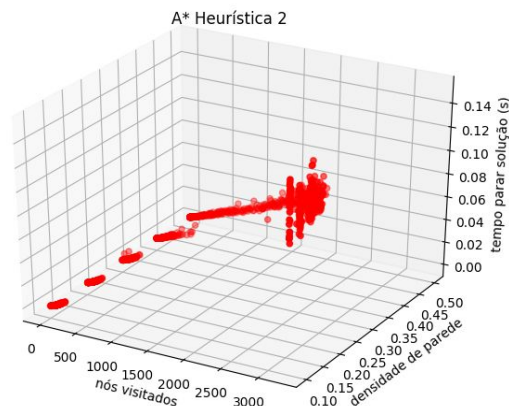
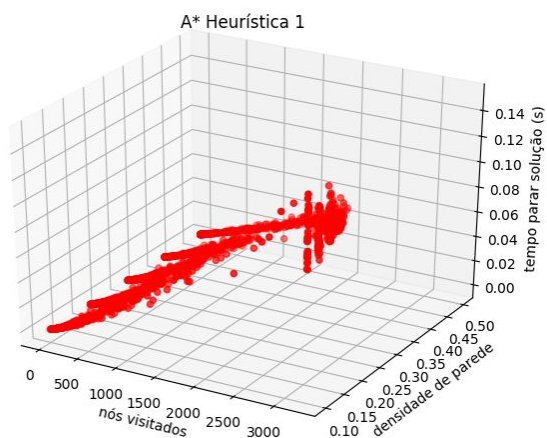
nodes visited	wall density	time to reach goal (s)	euclidian distance to goal
2620.0	0.3	0.07637834548950195	50.28916384272063
1032.0	0.4	0.027688026428222656	40.162171256046406
509.0	0.1	0.01714324951171875	39.20459156782532
426.0	0.1	0.010762929916381836	35.73513677041127
272.0	0.4	0.0053021907806396484	34.655446902326915
187.0	0.3	0.003699064254760742	32.55764119219941
2608.0	0.3	0.07042741775512695	23.08679276123039
251.0	0.1	0.006264209747314453	21.18962010041709
133.0	0.2	0.0028188228607177734	20.09975124224178
131.0	0.4	0.0025522708892822266	18.384776310850235
75.0	0.2	0.001394510269165039	9.0
46.0	0.2	0.0008225440979003906	8.246211251235321

A* Heurística Distância Ao quadrado

nodes visited	wall density	time to reach goal (s)	euclidian distance to goal
2620.0	0.3	0.07941722869873047	50.28916384272063
207.0	0.4	0.004862785339355469	40.162171256046406
124.0	0.1	0.0028390884399414062	39.20459156782532
102.0	0.1	0.0022220611572265625	35.73513677041127
100.0	0.4	0.0019359588623046875	34.655446902326915
88.0	0.3	0.0017650127410888672	32.55764119219941
2608.0	0.3	0.07091426849365234	23.08679276123039
69.0	0.1	0.001332998275756836	21.18962010041709
58.0	0.2	0.0010488033294677734	20.09975124224178
49.0	0.4	0.0009534358978271484	18.384776310850235
49.0	0.2	0.0008883476257324219	9.0
29.0	0.2	0.0005271434783935547	8.246211251235321

Gráficos tridimensionais para análise da dependência entre as métricas adotadas (porcentagem de parede, nós visitados, tempo de execução)





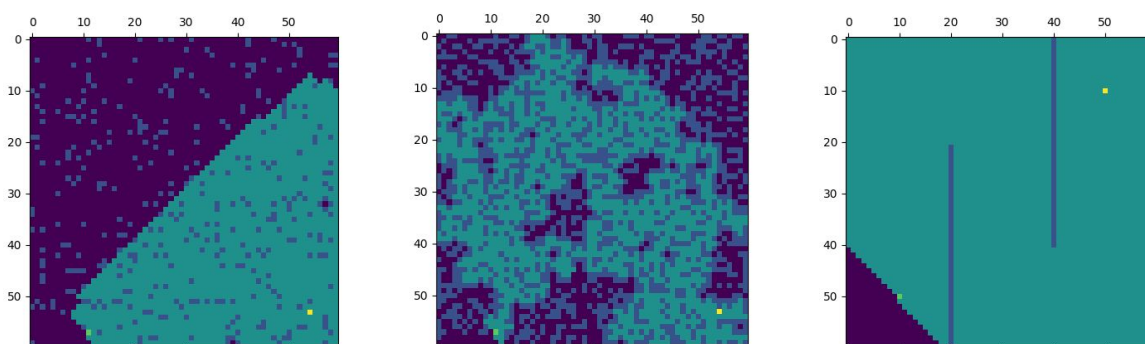
Imagens das buscas em alguns labirintos

Imagens Amostrais das 4 buscas , (da direita pra esquerda): busca feita em um labirinto aleatório densidade de parede de 10% (1), 50% (2) e busca feita no labirinto fornecido no enunciado (3).

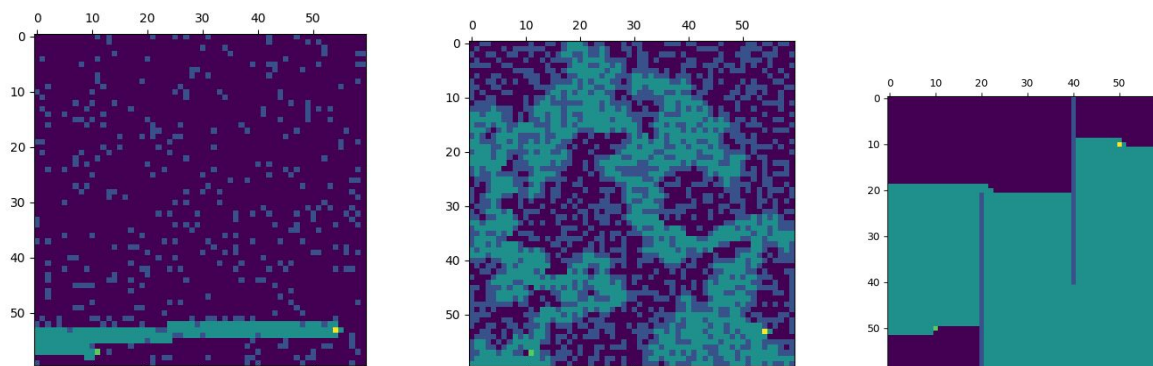
Legenda

Verde = chegada, amarelo=partida, azul mais escuro=posições não visitadas, azul escuro = paredes, verde escuro=posições visitadas

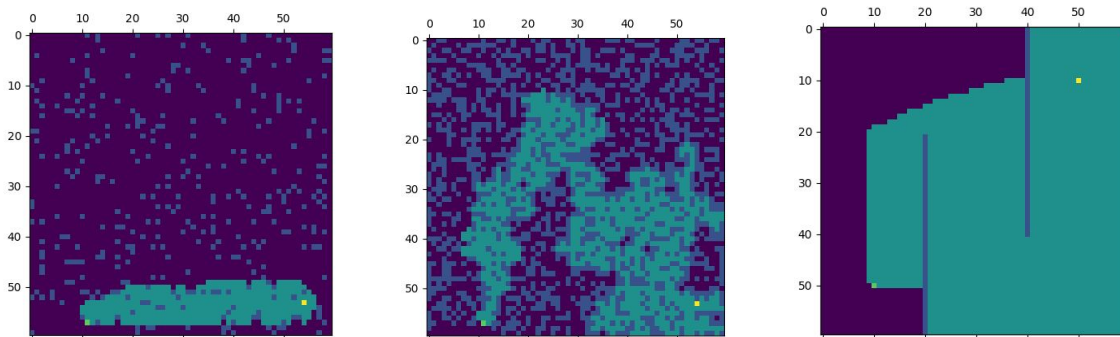
BFS



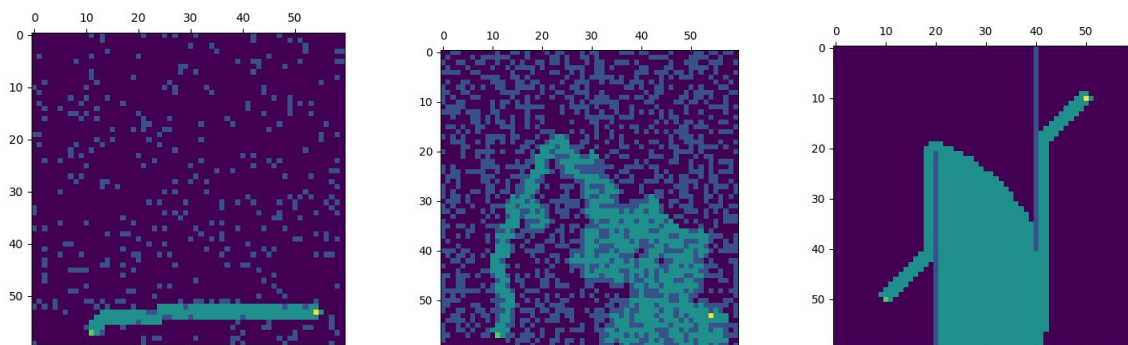
DFS



A* Heurística de distância euclidiana



A* Heurística de distância ao quadrado



Materiais e métodos

Os testes foram conduzidos com configurações aleatórias de labirinto, com o intuito de mostrar que o caso médio do algoritmo pode ser representado melhor em um ambiente aleatório com um grande número de elementos do espaço amostral.

No caso aleatório a porcentagem de paredes no labirinto foi variada de 10% a 50% assim como o ponto de partida e de chegada, em 400 iterações distintas, com a intenção de explicitar como essa variação afeta o desempenho dos algoritmos através dos gráficos 3D da seção [\[Tabelas e gráficos\]](#).

Tabelas foram extraídas que permitem efetuar a comparação entre os algoritmos, a enfatizar as duas métricas adotadas pelo A* em uma mesma configuração, através da análise de nós visitados e de tempo, ambos diretamente proporcionais entre si e à densidade de parede.

Através das análises assintóticas efetuadas pelo cormen foi possível extrair um modelo de comparação entre os algoritmos com um pouco mais rigor usando os dados levantados.

Resultado e Discussão

Análise pelos gráficos 3D seção

Podemos perceber de forma geral pelo gradiente que a densidade de parede é diretamente proporcional ao tempo de execução, mesmo em um mesmo número de nós visitados.

É perceptível a quantidade de flutuações do **DFS** em relação ao **BFS**, mesmo a complexidade de ambos sendo $O(|V| + |E|)$ essa diferença se dá por conta do critério de profundidade do DFS, em um ambiente aleatório em 400 labirintos distintos tornou-se notável que BFS, em média, possui comportamento similar ao DFS, sendo que o DFS possui maior spikes de granularidade sendo ou pior que o BFS (**penúltima linha de ambas as tabelas**) ou melhor que o BFS (**linha 5 de ambas tabelas**) em situações específicas (maior ou menor proximidade do ponto final e inicial).

As heurísticas do A^* de distância euclidiana em relação ao ponto, mostraram-se extremamente mais efetivas em relação às buscas não informadas **BFS** e **DFS**. pois nesse caso a heurística se baseia em cada vez mais diminuir a distância ao destino. Assim, a heurística 2 foi obtida a partir do aperfeiçoamento da heurística 1.

A heurística 2 do A^* (distância euclidiana ao quadrado) performa melhor que a 1, porque ela tende a “punir” mais o algoritmo quando ele se distancia da chegada, então ao encontrar uma parede, este tende a andar ao seu redor até encontrar uma maneira de superá-la. Portanto em casos com densidade de parede maior, $A^* 2$, em média, se comporta melhor, entretanto ao atingir uma certa densidade de parede a performance de ambos começa a se tornar similar. Isso acontece porque se existem poucas formas de contornar o caminho, ambos irão percorrer praticamente todos as posições que separam esse contorno.

Análise pelas tabelas

Um fator interessante é apontado pelas tabelas é que em média a distanciação do ponto de chegada e de partida afeta mais um ambiente aleatório com uma maior densidade de paredes no caso do $A^* 1$ (distância euclidiana) do que no $A^* 2$ (distância euclidiana ao quadrado), assim como o gráfico 3D também apontou

É perceptível pela linha **10** das tabelas de BFS e DFS que o DFS tende a performar melhor com uma densidade de parede maior que restringe o seu aprofundamento em direções evitando que talvez ele se distancie cada vez mais da chegada.

Conclusão

Considerando configurações aleatórias A heurística 2 do A^* (distância euclidiana ao quadrado) mostrou-se melhor em relação às demais, ainda mais em casos com maior número de paredes, por priorizar mais a redução da distância ao destino.

As análises foram em um grande espaço amostral, entretanto a complexidade seria melhor analisada com a utilização do fator de branch médio e o número médio de profundidade de cada algoritmo, cálculo que não pôde ser realizado facilmente com modificações no arquivo de search.py da biblioteca do **Aima**.

O estudo representa o comportamento do caso médio dos 4 algoritmos, entretanto ao invés de utilizar paredes aleatórias, uma boa métrica poderia ser testar a performance dos métodos em ambientes com paredes que fecham regiões inteiras e com poucos espaços válidos para passagem. Nesse caso estima-se que não haveria tanta diferença de performance entre as duas formas implementações com heurísticas diferentes do A^* que foram apresentadas.

Se o posicionamento de todas as paredes fosse considerado como uma informação conhecida a priori pelo agente, uma forma de melhorar drasticamente o desempenho do algoritmo seria levar em conta uma terceira heurística: ao encontrar uma parede, definir a nova chegada intermediária como o buraco da parede que reduz a distância entre o goal definitivo.

A análise gráfica de 400 execuções diferentes em labirintos aleatórios foi o suficiente para mostrar que o comportamento assintótico dos algoritmos condiz com os dados coletados (como analisado anteriormente no caso do BFS e DFS) e observar as diferenças de pontos de spikes (máximo e mínimos locais) entre eles.

Bibliografia Artificial Intelligence A Modern approach [b1] Stuart and Peter; - Algorithms [b2] - Thomas H. Cormen