

Universidade de São Paulo

Instituto De Ciências Matemáticas e de Computação

Trabalho 1 - Resolução de Problemas por meio de Buscas

Carlos Henrique Hannas de Carvalho	nº USP: 11965988
Carlos Nery Ribeiro	nº USP: 12547698
Erik Melges	nº USP: 12547399
Gabriel Ribeiro Rodrigues Dessotti	nº USP: 12547228
Gustavo Barbosa Sanchez	nº USP: 11802440
Pedro Antonio Bruno Grando	nº USP: 12547166
Pedro Manicardi Soares	nº USP: 12547621

Prof. Dra. Solange Oliveira Rezende

SCC0630 - Inteligência Artificial

Sumário

1	Introdução	1
2	Modelagem	1
2.1	Definição do Problema	1
2.2	Construção do Grafo	1
2.2.1	Atributos das Arestas	3
2.3	Estratégias de Busca	5
2.3.1	Busca Cega	5
2.3.2	Busca Informada	6
3	Implementação	6
3.1	Busca em Largura	6
3.2	Busca A*	6
4	Resultados	7
4.1	Análise das Buscas	7
4.2	Interface	8
5	Conclusão	11

Lista de Figuras

1	Rede oficial da Pássaro Marron.	1
2	Grafo construído.	2
3	Análise de tempo em BFS e A*.	7
4	Análise de distância em BFS e A*.	8
5	Análise de custo de combustível em BFS e A*.	8
6	Interface de busca.	9
7	Exemplificação de busca BFS.	9
8	Exemplificação de busca A*.	10
9	Exemplificação de busca BFS.	10
10	Exemplificação de busca A*.	11

1 Introdução

Um problema de busca baseia-se na tomada de ações, que devem ser seguidas, formando um caminho de um ponto inicial até uma meta final. Os problemas de busca possuem um modelo: estado inicial - onde o agente inicia a busca; espaço de estados - conjunto possível que pode-se percorrer; estado final - onde o agente procura chegar. Esse modelo é usualmente apresentado como um grafo e há alguns algoritmos para solucionar o problema da busca.

Os algoritmos de buscas podem ser definidos como: informado ou cego. As buscas informadas possuem um agente que pode estimar o custo do ponto inicial até a meta; por sua vez, a busca cega não há alguma estimativa entre estado inicial e final. No atual trabalho estuda-se os algoritmos de busca A* (informado) e busca em largura (cego).

Durante o processo do algoritmo informado A^* , gera-se um árvore de busca, que se expande até encontrar uma solução. Os sub-processos (caminhhos) da árvore possuem uma função de avaliação, que é o somatório de custos passados (do estado inicial até o estado atual) e heurísticas de custos futuros (estimativas do estado atual até o estado final). Analisa-se a função de avaliação de todos os caminhos em aberto, mantendo ativo aquele que possui o menor valor. Tendo uma heurística admissível, a estratégia adotada por A^* retorna o sub-processo ótimo.

A busca em largura não recebe algum tipo de informação sobre custos até a meta. A estratégia consiste em expandir todos os sucessores do nó atual, depois os sucessores desses nós e assim por diante. Isso garante que haverá uma possível solução, mesmo sem informação de custos - diferentemente do que acontece com a busca em profundidade, que pode não encontrar uma solução.

Optou-se por estudar as buscas A^* e em largura, pois sempre haverá, pelo menos, uma solução de caminho, a partir de um estado inicial. Isso possibilita uma comparação de caminhos e custos entre os dois algoritmos estudados, dado o problema da malha rodoviária da companhia Pássaro Marron, que será discutido em breve.

2 Modelagem

2.1 Definição do Problema

O problema estipulado é a otimização da malha rodoviária da empresa Pássaro Marron. A resolução visa encontrar qual a melhor forma de sair de uma cidade para outra, considerando que não haja uma rota direta entre elas, com os menores tempo e custo de combustível possível por viagem. Essa otimização traz benefícios econômicos para a empresa, pois há minimização do custo de combustível por quilômetro rodado, e para os passageiros, consequentemente, já que os preços de passagem são baseados em custos operacionais de viagem.

2.2 Construção do Grafo

A construção do grafo baseou-se no mapa da rede da empresa Pássaro Marron, ilustrada na Figura 1:



Figura 1: Rede oficial da Pássaro Marron.

Entretanto, algumas cidades não possuem quantidades relevantes de ônibus que fizessem sentido no escopo do presente trabalho. Nesse sentido, considerou-se as 33 cidades a serem estudadas:

Cidades
Aeroporto de Guarulhos
Aparecida
Bertioga
Caçapava
Cachoeira Paulista
Campinas
Campos do Jordão
Caragatatuba
Cruzeiro
Guaratinguetá
Ilhabela
Itajubá
Jacareí
Lorena
Mogi das Cruzes
Paraibuna
Paraisópolis
Pouso Alegre
Queluz
Roseira
Santa Branca
Santa Isabel
Santo André
Santo Antônio do Pinhal
Santos
São Bento do Sapucaí
São Bernardo do Campo
São Caetano do Sul
São José dos Campos
São Paulo
São Sebastião
Taubaté
Tremembé
Ubatuba

A partir disso, verificou-se todas as possíveis conexões entre elas. Construiu-se, a partir da Tabela 1, o grafo abaixo:

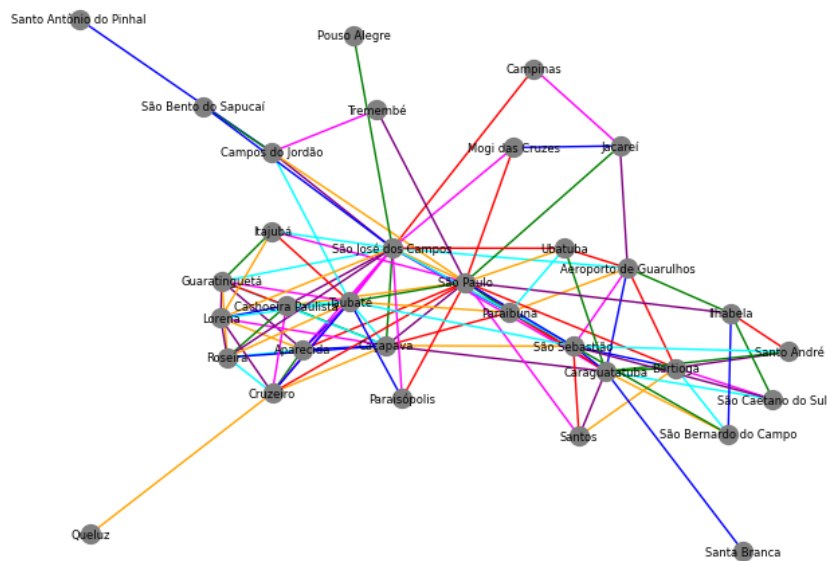


Figura 2: Grafo construído.

2.2.1 Atributos das Arestas

Conforme descrito na Seção 2.1, a resolução do problema inclui a otimização de tempo e gasto de combustível. Desse modo, para cada aresta considerou-se três atributos: distância percorrida, tempo de viagem e custo de combustível.

Distância percorrida: Para determinar a distância percorrida, utilizou-se o Google Maps calculando a rota entre as rodoviárias das cidades envolvidas.

Tempo de viagem: O tempo de viagem a ser considerado foi determinado a partir dos dados disponíveis no próprio site da empresa.

Custo de combustível: Para calcular uma estimativa de custo médio de combustível para cada rota, adota-se:

- Consumo energético médio: 0,511 L/km [3]
- Preço médio do Diesel: 5,98 reais/L [4]

Portanto, a estimativa de custo médio por quilômetro:

$$custo_{combustível} = 0,511 \frac{L}{km} \times 5,98 \frac{reais}{L} \approx 3,10 \frac{reais}{km}$$

Então, constrói-se a seguinte tabela, com os atributos de cada rota, ou seja, cada aresta do grafo da Figura 2:

Tabela 2: Rotas e custos de viagens

Rota	Distância Percorrida (km)	Tempo de Viagem (min)	Custo de Combustível (R\$)
GRU - Bertioga	99	155	306,90
GRU - Caraguatatuba	161	140	499,10
GRU - Ilhabela	189	200	585,90
GRU - Jacareí	68	70	210,80
GRU - Paraibuna	110	90	341,00
GRU - S.J. dos Campos	76	70	235,60
GRU - São Sebastião	184	205	570,40
GRU - Ubatuba	212	280	657,20
Aparecida - Caçapava	60	60	186,00
Aparecida - Cruzeiro	47	95	145,70
Aparecida - Guaratinguetá	7	15	21,70
Aparecida - Lorena	20	30	62,00
Aparecida - Roseira	12	15	37,20
Aparecida - S.J. dos Campos	85	85	263,50
Aparecida - São Paulo	172	180	533,20
Aparecida - Taubaté	44	35	136,40
Bertioga - Caraguatatuba	112	205	347,20
Bertioga - Santo André	99	140	306,90
Bertioga - Santos	77	100	238,70
Bertioga - S. B. do Campo	96	110	297,60
Bertioga - S. C. do Sul	107	170	331,70
Bertioga - São Paulo	116	165	359,60
Bertioga - São Sebastião	87	155	269,70
Caçapava - C. Paulista	92	145	285,20
Caçapava - Caraguatatuba	101	170	313,10
Caçapava - Cruzeiro	105	175	325,50
Caçapava - Guaratinguetá	66	85	204,60
Caçapava - Lorena	79	115	244,90
Caçapava - Paraibuna	50	90	155,00
Caçapava - Roseira	50	55	155,00
Caçapava - S. J. dos Campos	27	25	83,70
Caçapava - São Paulo	120	130	372,00
Caçapava - São Sebastião	123	230	381,30
Caçapava - Taubaté	21	30	65,10
C. Paulista - Cruzeiro	18	30	55,80

Continua na próxima página

Rota	Distância Percorrida (km)	Tempo de Viagem (min)	Custo de Combustível (R\$)
C. Paulista - Guaratinguetá	28	55	86,80
C. Paulista - Lorena	18	30	55,80
C. Paulista - Roseira	44	95	136,40
C. Paulista - S. J. dos Campos	117	130	362,70
C. Paulista - São Paulo	213	245	660,30
C. Paulista - Taubaté	75	70	232,50
Campinas - Jacareí	141	150	437,10
Campinas - S. J. dos Campos	150	180	465,00
C. do Jordão - S. A. do Pinhal	20	50	62,00
C. do Jordão - S. B. do Sapucaí	33	90	102,30
C. do Jordão - S. J. dos Campos	85	120	263,50
C. do Jordão - São Paulo	181	225	561,10
C. do Jordão - Taubaté	45	90	139,50
C. do Jordão - Tremembé	38	50	117,80
Caraguatatuba - Paraibuna	60	60	186,00
Caraguatatuba - Santa Branca	106	110	328,60
Caraguatatuba - Santo André	194	350	601,40
Caraguatatuba - Santos	248	315	768,80
Caraguatatuba - S. B. do Campo	201	320	623,10
Caraguatatuba - S. C. do Sul	186	380	576,60
Caraguatatuba - S. J. dos Campos	92	110	285,20
Caraguatatuba - São Paulo	180	160	558,00
Caraguatatuba - São Sebastião	26	60	80,60
Caraguatatuba - Ubatuba	52	90	161,20
Cruzeiro - Lorena	32	55	99,20
Cruzeiro - Queluz	25	35	77,50
Cruzeiro - Roseira	58	125	179,80
Cruzeiro - S. J. dos Campos	131	160	406,10
Cruzeiro - São Paulo	227	275	703,70
Cruzeiro - Taubaté	89	100	275,90
Guaratinguetá - Itajubá	86	155	266,60
Guaratinguetá - Lorena	12	30	37,20
Guaratinguetá - Roseira	18	25	55,80
Guaratinguetá - S. J. dos Campos	91	100	282,10
Guaratinguetá - Taubaté	50	50	155,00
Ilhabela - Santo André	223	290	691,30
Ilhabela - S. B. do Campo	231	260	716,10
Ilhabela - S. C. do Sul	215	320	666,50
Ilhabela - São Paulo	210	270	651,00
Itajubá - Lorena	71	120	220,10
Itajubá - S. J. dos Campos	174	230	539,40
Itajubá - São Paulo	261	325	809,10
Itajubá - Taubaté	132	170	409,20
Jacareí - Mogi das Cruzes	54	75	167,40
Jacareí - São Paulo	85	100	263,50
Lorena - Roseira	31	55	96,10
Lorena - S. J. dos Campos	104	130	322,40
Lorena - Taubaté	62	80	192,20
Mogi das Cruzes - S. J. dos Campos	63	105	195,30
Mogi das Cruzes - São Paulo	61	80	189,10
Paraibuna - S. J. dos Campos	35	50	108,50
Paraibuna - São Paulo	123	150	381,30
Paraibuna - São Sebastião	74	120	229,40
Paraibuna - Taubaté	67	100	207,70
Paraibuna - Ubatuba	105	160	325,50
Paraisópolis - S. J. dos Campos	100	150	310,00
Paraisópolis - São Paulo	217	245	672,20
Paraisópolis - Taubaté	81	90	251,10
Pouso Alegre - S. J. dos Campos	236	230	731,60
Roseira - S. J. dos Campos	73	75	226,30

Continua na próxima página

Rota	Distância Percorrida (km)	Tempo de Viagem (min)	Custo de Combustível (R\$)
Roseira - Taubaté	32	25	99,20
Santo André - São Sebastião	213	315	660,30
Santos - São Paulo	72	100	223,20
Santos - São Sebastião	160	265	496,00
S. B. do Sapucaí - S. J. dos Campos	83	120	257,30
S. B. do Campo - São Sebastião	216	280	669,60
S. C. do Sul - São Sebastião	217	350	672,70
S. J. dos Campos - São Paulo	97	90	300,70
S. J. dos Campos - São Sebastião	109	160	337,90
S. J. dos Campos - Taubaté	43	60	133,30
S. J. dos Campos - Ubatuba	134	190	415,40
São Paulo - São Sebastião	198	250	613,80
São Paulo - Taubaté	133	150	412,30
São Paulo - Tremembé	157	210	486,70
São Paulo - Ubatuba	226	300	700,60
São Sebastião - Taubaté	147	230	455,70

Para a busca heurística A*, precisamos de uma função de avaliação, que estima o custo de um nó até o outro. Assim, usamos a distância estimada em linha reta entre as cidades, as quais são exibidas na tabela a seguir.

Tabela 3: Rotas e custos estimados de viagens

Rota	Distância Estimada (km)
GRU - Aparecida	147
GRU - Bertioga	60
GRU - Caçapava	91
GRU - Cachoeira Paulista	181
GRU - Campinas	83
GRU - Campos do Jordão	126
GRU - Caraguatatuba	114
GRU - Cruzeiro	189
GRU - Guaratinguetá	155
GRU - Ilhabela	124
GRU - Itajubá	157
GRU - Jacareí	60
...	...

Como há 33 cidades em estudo, resumiu-se as combinações de cidades. A tabela acima pode ser vista completa no arquivo .csv: [GitHub do Trabalho 1](#).

Percebe-se que se trata de uma heurística admissível, pois as distâncias estimadas são as distâncias em linha reta entre as duas cidades, que serão menores ou iguais às distâncias reais. Como os cálculos das funções de avaliação dos outros custos são proporcionais a essa distância, pode-se dizer que eles também têm heurísticas admissíveis.

2.3 Estratégias de Busca

2.3.1 Busca Cega

A **busca em largura** explora primeiro todos os nós vizinhos do nó inicial (cidade de partida) - antes de partir para os mais distantes e, enfim, chegar no nó final - a cidade de destino. No contexto da otimização da malha rodoviária, a busca em largura foi usada visando encontrar o melhor caminho entre duas cidades, mesmo que elas não tenham ligação direta, baseando-se nas medidas explicitadas anteriormente - distância, tempo e custo, em dinheiro, de combustível.

No escopo das buscas não informadas, a busca em largura foi escolhida devido a possuir algumas vantagens em relação às outras, dentre as quais podemos destacar a sua simplicidade e completude - pois explora todas as rotas possíveis -, bem como o fato do grafo ser adequado a ela.

2.3.2 Busca Informada

O **algoritmo A*** surge como uma alternativa mais eficiente que a estratégia explorada no tópico anterior. Ao contrário da busca em largura, o A* usa uma informação extra: a distância em linha reta entre as cidades - para ajudar na previsão total de custo em um caminho.

Esse algoritmo faz escolhas inteligentes de qual será o caminho mais promissor com base em uma função de avaliação, que combina o custo real atual do caminho com o custo previsto para o restante dele. O uso dessa heurística faz com que sejam evitadas as explorações de caminhos que possivelmente serão mais caros. Assim, caso a heurística seja admissível - no nosso caso, ela é, uma vez que a distância em linha reta sempre é menor ou igual que a distância real -, o algoritmo garante a solução ótima, sendo esse o grande motivo da escolha por esse algoritmo.

3 Implementação

3.1 Busca em Largura

O código da busca em largura foi estruturado para lidar com um arquivo .csv, cujas informações serão extraídas usando a biblioteca Pandas. O arquivo contém colunas com dados de custos de tempo, distância e combustível entre duas cidades, semelhante à tabela 2. Com auxílio da biblioteca NetworkX para construir o grafo direcionado, cujos nós são as cidades e as arestas têm os três valores de custo.

Após o recebimento dos nós de origem e de destino, a função de busca percorre uma fila começando pela origem, explorando os vizinhos até alcançar o destino. Essa função guarda um dicionário com os nós anteriores para cada nó visitado e, assim, quando o destino é encontrado, basta retornar esse dicionário, que terá o caminho até a solução.

Para efeito de comparação com o algoritmo da busca informada, essa função também armazena o número de nós visitados.

Além disso, temos uma função do caminho final, a qual percorre o dicionário e encontra o caminho desenhado até a solução. Se esse caminho existe, ele é retornado, em conjunto com o número de nós visitados e o custo total - que é a soma dos pesos das arestas do grafo.

O código completo pode ser conferido: [GitHub do Trabalho 1](#).

3.2 Busca A*

Assim como no tópico anterior, foi usada a biblioteca NetworkX para operar o grafo, que foi construído com os nós sendo as cidades e as arestas entre eles sendo os três custos: tempo, distância e custo.

Em seguida, nesse grafo, após a definição de um nó início e um nó destino, foi aplicada a busca por A*. Para implementação dela, usou-se uma fila de prioridades em uma heap, em que os nós com menor avaliação tinham a maior prioridade - uma vez que gerariam os melhores caminho -, e uma outra fila para armazenar nós já expandidos a fim de se evitar o retorno neles.

Assim, o algoritmo implementado basicamente abre um nó, avalia cada um dos nós seguintes dele se ainda não foram expandidos, colocando-os na heap. Na próxima iteração do loop principal, é feito um pop no nó mais promissor e faz-se o mesmo processo até que seja encontrado o nó de destino.

A função de estimativa de custo é dada pela soma do custo real acumulado até o nó atual com o custo estimado do nó atual até o nó de destino - a heurística do nó -, que é o custo em linha reta entre as cidades extraídos de um arquivo csv definido. Dependendo do tipo de variável que está procurando otimizar (tempo, distância ou custo), fazemos as seguintes operações com as distâncias vindas do csv:

- Tempo: a distância é dividida por uma velocidade média de 100 km/h e obtemos um tempo em h;
- Distância: usamos a medida em km vinda do arquivo .csv;
- Custo: multiplica pelo custo estimado de R\$ 3/km e obtemos um custo em R\$.

Feitas essas operações, concluímos que a estimativa da heurística sempre subestima o custo real, uma vez que usamos a distância em linha reta (distância de Manhattan) nas operações, tornando-se uma heurística admissível nos três casos, fornecendo sempre uma solução ótima.

O código completo pode ser encontrado no [GitHub do Trabalho 1](#).

4 Resultados

4.1 Análise das Buscas

Uma vez concluídas as implementações dos dois algoritmos de busca utilizados, foram feitos testes para analisar a eficiência deles.

No gráfico abaixo, para o parâmetro de custo tempo, foi analisada a frequência de nós visitados em função do número de nós visitados para cada um dos dois algoritmos.

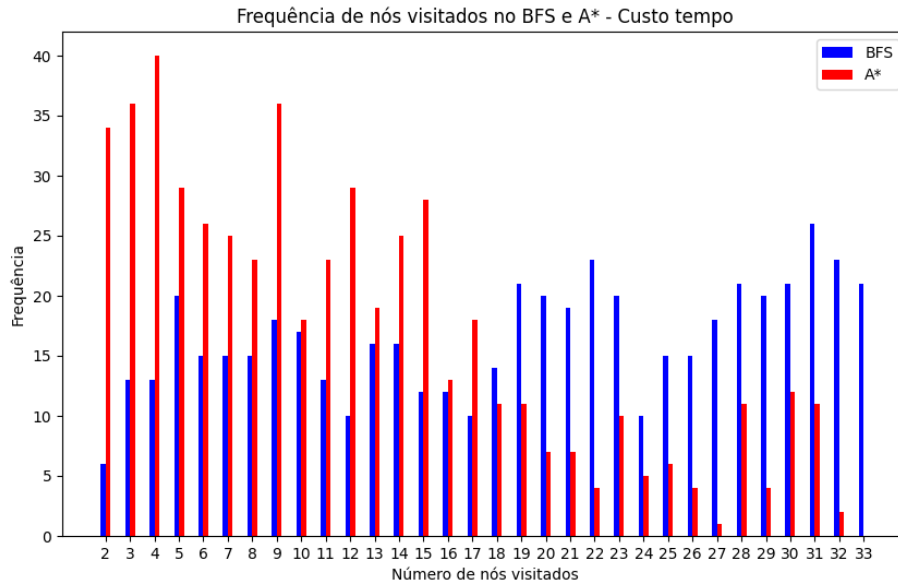


Figura 3: Análise de tempo em BFS e A*.

Por exemplo, foram visitados apenas 2 nós em 6 pares de cidades na busca em largura e em 34 no A*, enquanto foram visitados 3 nós em 13 pares da busca em largura e em 36 no A*. Analisando o gráfico como um todo, percebe-se que a tendência é que a busca em largura tenha poucos pares de cidades que contam com poucos nós visitados, enquanto o A* tenha poucos pares de cidades que contam com muitos nós visitados para encontrar a solução.

A seguir, nos dois próximos gráficos, foi realizada a mesma análise mudando o parâmetro de custo para, respectivamente, distância e combustível.

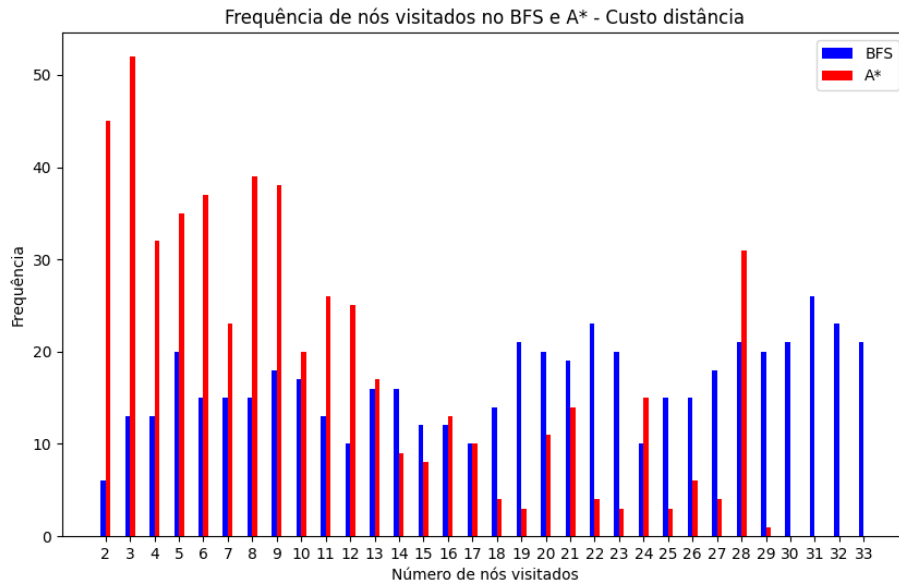


Figura 4: Análise de distância em BFS e A*.

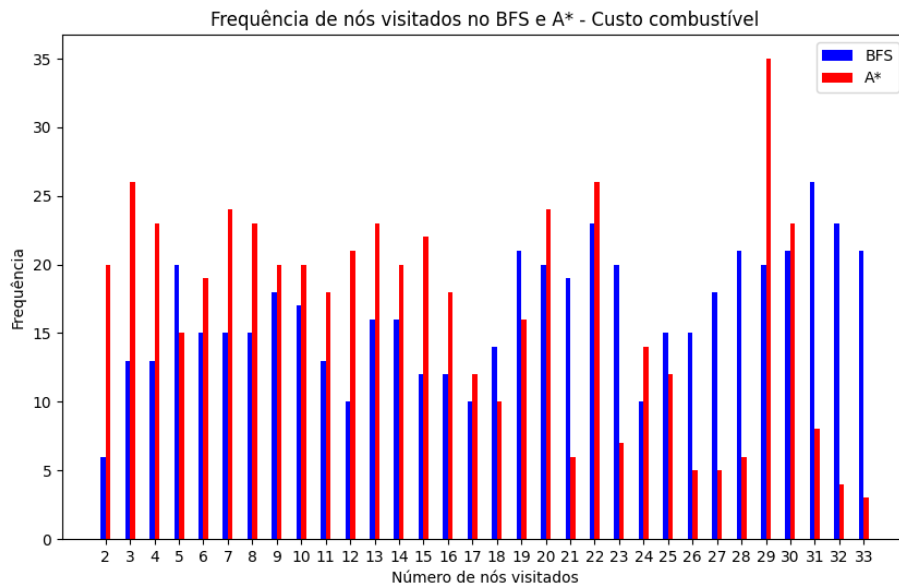


Figura 5: Análise de custo de combustível em BFS e A*.

Visualizando os dois novos gráficos expostos acima, torna-se evidente que a tendência é muito similar: poucas vezes a busca em largura vai ser eficiente de visitar menos nós, sendo o A* o algoritmo que tem mais frequência nas menores quantidades de nós visitados.

4.2 Interface

Para melhor facilidade de um usuário em procurar um caminho, foi criada uma interface gráfico, mostrada na imagem a seguir.

Otimizador de Rotas da Pássaro Marron

Qual tipo de busca deseja usar?

☒ BFS - Busca em Largura

☐ A*

Qual tipo de custo você deseja otimizar?

Tempo

Origem:

Campinas

Destino

São José dos Campos

Buscar

Figura 6: Interface de busca.

Nessa interface, escolhemos uma das duas buscas implementadas e, em seguida, o tipo de custo a ser otimizado e as cidades de origem e destino. Depois disso, basta apertar o botão 'Buscar' para gerar os resultados. Vamos mostrar, agora, exemplos do funcionamento dos algoritmos com essa interface.

Em primeiro lugar, vamos propor uma viagem de Lorena a Campinas otimizando o tempo numa busca em largura.

Qual tipo de busca deseja usar?

☒ BFS - Busca em Largura

☐ A*

Qual tipo de custo você deseja otimizar?

Tempo

Origem:

Lorena

Destino

Campinas

Buscar

Resultados:

Caminho final: ['Lorena', 'São José dos Campos', 'Campinas']

Custo: 310.0 min

Número de nós visitados: 17

Figura 7: Exemplificação de busca BFS.

De acordo com a imagem acima, obtemos que o caminho para fazer essa viagem é sair de Lorena para São José dos Campos para depois seguir para Campinas, tendo um custo de 310 minutos. Para essa busca, 17 nós foram visitados. Alterando para o algoritmo A*, temos o seguinte resultado.

Qual tipo de busca deseja usar?

☐ BFS - Busca em Largura

☒ A*

Qual tipo de custo você deseja otimizar?

Tempo

Origem:

Lorena

Destino

Campinas

Buscar

Resultados:

Caminho final: ['Lorena', 'Aparecida', 'São José dos Campos', 'Campinas']

Custo: 295.0 min

Número de nós visitados: 11

Figura 8: Exemplificação de busca A*.

Com isso, percebe-se que o caminho foi diferente: Lorena para Aparecida, Aparecida para São José dos Campos e, por fim, São José dos Campos para Campinas, tendo um custo de tempo menor de 295 minutos. Além disso, menos nós foram visitados: apenas 11 em contraponto aos 17 da busca em largura.

Seguindo para outro exemplo, vamos propor uma outra viagem: agora, de Campos do Jordão para Ubatuba, otimizando pela distância.

Qual tipo de busca deseja usar?

☒ BFS - Busca em Largura

☐ A*

Qual tipo de custo você deseja otimizar?

Distância

Origem:

Campos do Jordão

Destino

Ubatuba

Buscar

Resultados:

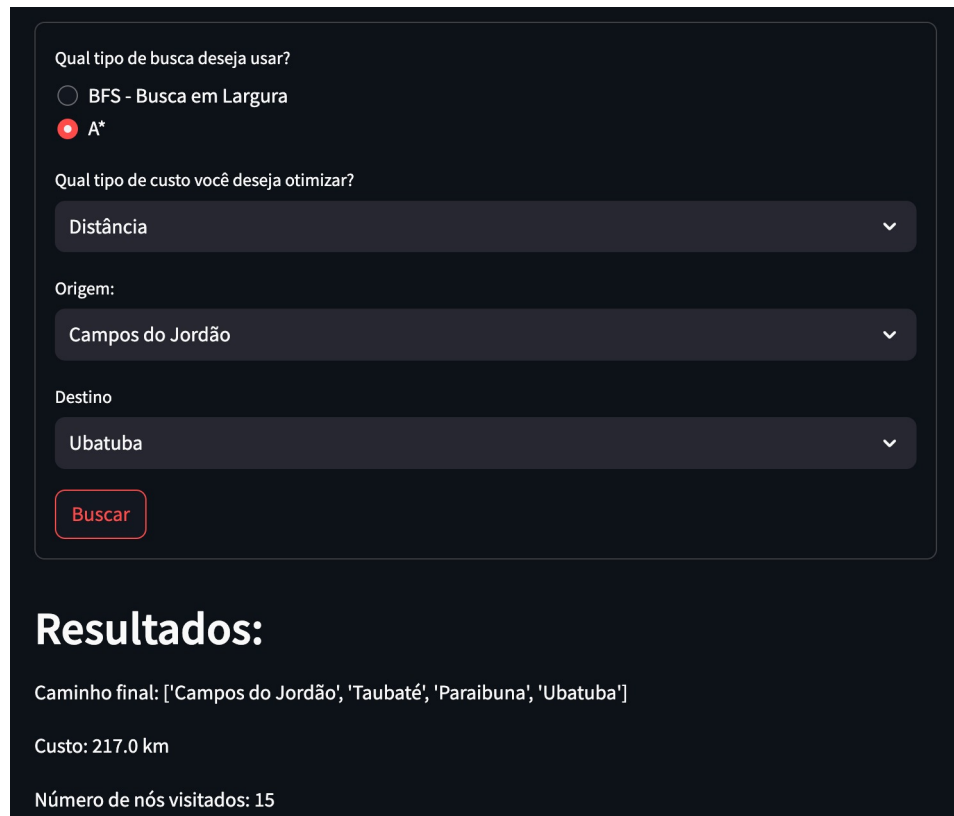
Caminho final: ['Campos do Jordão', 'São José dos Campos', 'Ubatuba']

Custo: 219.0 km

Número de nós visitados: 24

Figura 9: Exemplificação de busca BFS.

Com a busca em largura, encontrou-se o caminho de Campos do Jordão a Ubatuba passando apenas por São José dos Campos no meio do caminho. A distância total ficou de 219 km e o algoritmo visitou 24 nós.



The image shows a dark-themed web interface for a search algorithm. At the top, it asks 'Qual tipo de busca deseja usar?' (Which type of search do you want to use?) with two radio buttons: 'BFS - Busca em Largura' (unselected) and 'A*' (selected). Below this, it asks 'Qual tipo de custo você deseja otimizar?' (Which type of cost do you want to optimize?) with a dropdown menu set to 'Distância' (Distance). The 'Origem:' (Origin) field is set to 'Campos do Jordão' and the 'Destino' (Destination) field is set to 'Ubatuba'. A red 'Buscar' (Search) button is located below the destination field. The results section, titled 'Resultados:', shows the final path: 'Caminho final: ['Campos do Jordão', 'Taubaté', 'Paraibuna', 'Ubatuba']', the cost: 'Custo: 217.0 km', and the number of visited nodes: 'Número de nós visitados: 15'.

Figura 10: Exemplificação de busca A*.

Agora, com o A*, nota-se que o caminho mudou. Agora, a viagem começa em Campos do Jordão, parte para Taubaté seguindo para Paraibuna e, de lá, termina em Ubatuba, tendo a distância reduzida para 217 km. Além disso, o número de nós visitados foi de 15.

Assim, nota-se que a busca em largura normalmente sai atrás da busca informada com A* nos quesitos de eficiência - uma vez que a busca em largura precisa mais nós na maior parte das vezes - e de otimalidade - já que o A* consegue garantir uma solução ótima.

5 Conclusão

Procurou-se otimizar a malha rodoviária da empresa Pássaro Marron, quando não há viagens diretas entre duas cidades, adotando três critérios: distância, tempo e custo de combustível por viagem. O intuito dessa otimização é trazer benefícios econômicos para a companhia, minimizando o custo operacional de viagem e consequentemente, para os passageiros. Dessa forma, procurando minimizar os custos e estudar problemas de buscas, implementou-se os algoritmos A* e busca em largura.

A análise entre os dois algoritmos indica que a busca em largura tende a ser eficiente (menor número de nós visitados) quando o espaço amostral é relativamente pequeno. Se há um espaço de busca maior, a busca em largura, embora simples e completa, tende a visitar mais nós em comparação com o A*. No contexto da otimização de rotas, os resultados mostram que a busca informada tende a encontrar caminhos mais eficientes em termos de tempo, distância ou custo de combustível, dependendo do critério escolhido.

Em resumo, a busca em largura é viável em situações com poucos espaços de estados. O algoritmo A*, quando há uma heurística admissível, destaca-se em problemas mais complexos, ainda oferecendo soluções de otimalidade.

Referências

- [1] S.Russel; P. Norvig. "Inteligência Artificial - Uma Abordagem Moderna". 4ª ed., 2022.
- [2] Rezende, Solange. "Estratégias de Busca Informada A* - slide de aula". 2024.

- [3] Universidade Federal do Rio de Janeiro. *Relatório Final - Rendimento Energético*. Disponível em: http://www.ltc.coppe.ufrj.br/src/uploads/2020/05/rel_final_rend_ener_03_03_16.pdf.
- [4] Petrobras. *Preços dos Combustíveis - Diesel*. Disponível em: <https://precos.petrobras.com.br/web/precos-dos-combustiveis/w/diesel/sp>.