

# Inteligência Artificial - Algoritmos de Busca: Redes Geográficas

**Nome:** Emerson Pereira Portela Filho

**Nome:** Gustav Shigueo Nicioka Asano

**Nome:** Pedro Henrique Raymundi

**Nome:** Wictor Dalbosco Silva

**NUSP:**11800625

**NUSP:**11212355

**NUSP:**11795634

**NUSP:**11871027

# 1. Introdução

Este relatório apresenta uma investigação sobre a aplicação de algoritmos de busca em Redes Geográficas, no contexto da implementação de um sistema baseado em inteligência artificial. O objetivo principal é explorar o desempenho de diferentes algoritmos de busca ao encontrar caminhos entre vértices em uma rede geográfica, bem como realizar comparações entre o algoritmo A\* e o algoritmo de Dijkstra. A linguagem escolhida pelo grupo para realizar a programação dos códigos foi Python.

Neste trabalho, serão abordados cinco algoritmos de busca: Busca em Profundidade, Busca em Largura, Algoritmo Best-First o Algoritmo A\* e também o de Dijkstra que servirá como comparação para o último experimento. O primeiro experimento consiste na geração de redes geográficas, onde serão selecionados vértices de origem e destino para aplicar todos os algoritmos de busca. Os resultados apresentados incluem o caminho final percorrido, destacando cada vértice que faz parte da solução, juntamente com a distância total percorrida.

No segundo experimento, serão geradas redes geográficas com diferentes configurações, e serão selecionados pares distintos de vértices (origem, destino) para aplicar os algoritmos de busca. Será reportada a distância média percorrida e o tempo médio gasto por cada algoritmo.

O terceiro experimento consiste em uma comparação entre o algoritmo A\* e o algoritmo de Dijkstra, utilizando as redes geradas no experimento anterior. Serão analisadas a eficiência e a precisão de ambos os algoritmos na busca por caminhos ótimos em diferentes configurações de redes geográficas.

Ao realizar esses experimentos, busca-se fornecer uma visão abrangente do desempenho dos algoritmos de busca em Redes Geográficas, contribuindo para o aprimoramento do conhecimento e da aplicação da inteligência artificial nesse campo. Os resultados obtidos poderão ser utilizados como base para a otimização de sistemas de navegação, logística e outras áreas que dependem da busca por caminhos eficientes em redes geográficas.

## 2. Implementação

### 2.1. Algoritmos de Busca

Para este projeto serão utilizados alguns algoritmos de busca que atuarão sobre uma rede geográfica, e a partir deles serão feitos alguns experimentos envolvendo os estudos sobre os caminhos que estes algoritmos encontram em diferentes tamanhos de grafos.

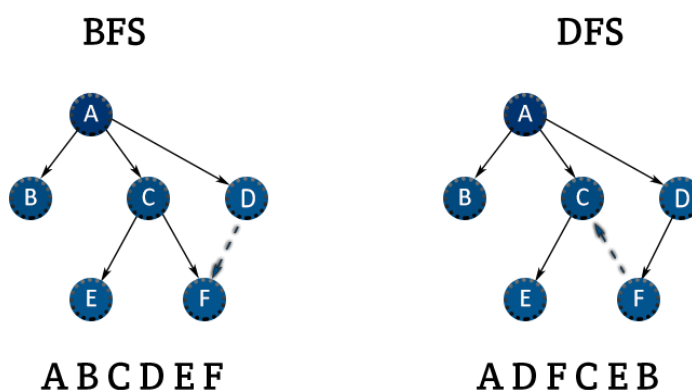
#### 2.1.1. Busca em profundidade (DFS)

A busca em profundidade é um algoritmo para percorrer ou pesquisar estruturas de dados de árvores ou grafos. O algoritmo começa no nó raiz (selecioneando algum nó arbitrário como o nó raiz no caso de um grafo) e explora o máximo possível ao longo de cada ramificação antes de retroceder.

Portanto, a ideia básica é começar pelo nó raiz ou por qualquer nó arbitrário, marcar o nó e mover-se para o nó não marcado adjacente, continuando esse ciclo até não haver mais nós adjacentes não marcados. Em seguida, retrocede e verifica outros nós não marcados e os percorre. Por fim, imprime os nós no caminho.

#### 2.1.2. Busca em largura (BFS)

A busca em largura, ao contrário da busca em profundidade, verifica todas as conexões do vértice antes de ir para o próximo nível. Para isso, é preciso guardar todos os vértices já visitados, resultando num consumo maior de memória, mas consegue chegar no destino mais rapidamente caso esteja num nível próximo da origem.



### 2.1.3. Algoritmo Best-First

Este algoritmo funciona como uma busca informada (também conhecido como Método Heurístico), ou seja, ela vai utilizar uma função de avaliação baseada em Fila de Prioridade que ajuda a decidir qual nó adjacente é mais promissor para realizar a exploração.

### 2.1.4. Algoritmo A\*

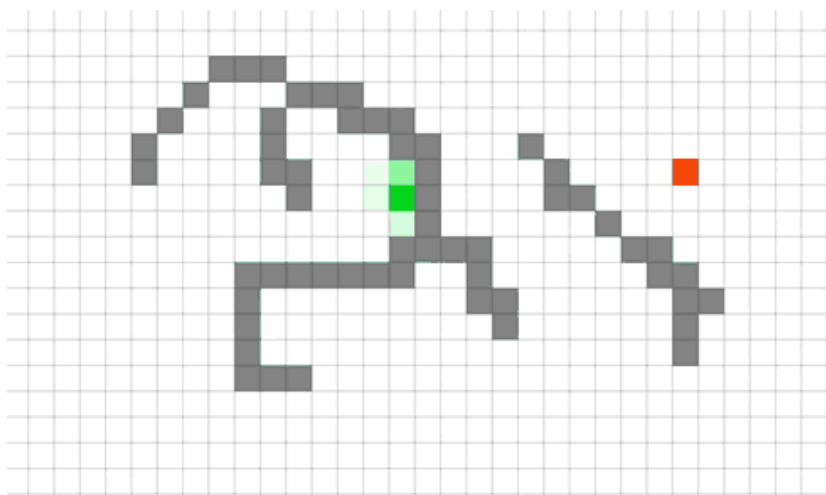
Este algoritmo é a combinação de aproximações heurísticas como do algoritmo Breadth First Search (Busca em Largura) e da formalidade do Algoritmo de Dijkstra.

Esse método usa os valores de duas funções para estimar o melhor caminho em relação ao custo, cujos valores são o G e o H.

- G: Valor do custo acumulado do caminho percorrido
- H: Valor restante do custo necessário até o destino

A soma dos dois valores acima estimam o caminho mais eficiente de um ponto inicial até o ponto final. O resultado dos valores acima é representado com a letra F e quanto menor for esse valor, menos custoso será o trajeto calculado.

A eficiência deste algoritmo é bastante relativa, tendo o seu desempenho diretamente influenciado pelo contexto do programa, com fatores como a qualidade dos dados de entrada e os métodos heurísticos adotados, que são usados para estimar o menor valor possível até o destino. Alguns dos mais usados são a distância de Manhattan e a distância euclidiana. (Para visualizar o gif entre no [docs](https://github.com/ysshah95/Astar-Algorithm-for-ACME-Robotics))



<https://github.com/ysshah95/Astar-Algorithm-for-ACME-Robotics>

## 2.2. Rede Geográfica

O grupo selecionou a Rede Geográfica para trabalhar, nela os nós são representados pontos de interesse que podem ser compreendidos como cidades, pontos turísticos ou outros locais geográficos. Já as arestas por sua vez são as distâncias entre tais pontos.

Algumas aplicações que utilizam as redes geográficas abrangem muitos escopos, desde problemas de logísticas, planejamentos urbanos, análise de redes sociais, até a realização de roteiros turísticos de viagens. Temos então que este tipo de rede é muito bom para representação de fluxos, como produtos, dados, mercadorias, e até pessoas.

Para a criação dos vértices foi feita uma função que recebe o tamanho  $n$  dos eixos  $x$  e  $y$ . A partir disso é usado uma função de números aleatórios que irá popular o espaço delimitado.

Enquanto as arestas, vão ser feitas de uma função que recebe os vértices gerados anteriormente e um fator  $\lambda$ . Ao gerar um número entre 0 e 1 e esse número for menor que  $P$  será gerado uma aresta, onde  $P$ , que utiliza a distância euclidiana entre os dois vértices ( $S_{ij}$ ), é dado por

$$P = e^{-\lambda S_{ij}}$$

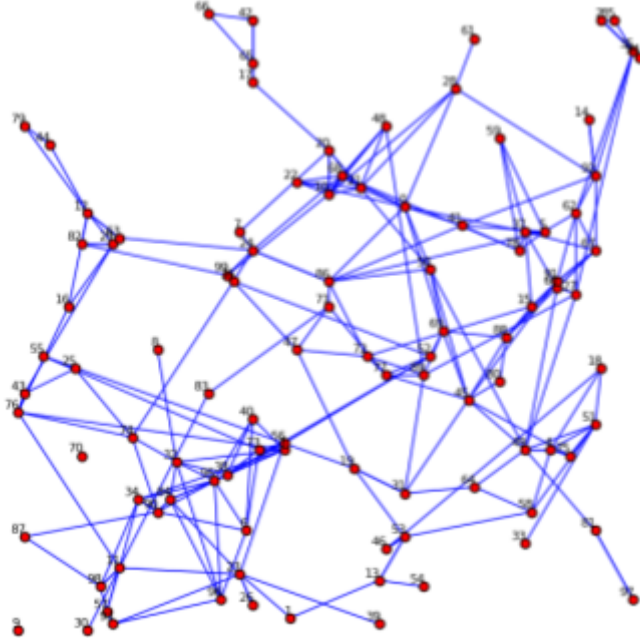
## 3. Resultados e discussões

### 3.1. Experimento 1

Neste experimento, para o tipo de rede selecionado pelo grupo, que foi Redes Geográficas, o objetivo era gerar redes e aplicar todos os algoritmos de busca a cada uma delas. Em cada execução, foi selecionado um vértice de origem e um vértice de destino. Será mostrado o caminho percorrido pelo algoritmo (incluindo todos os vértices que fazem parte do caminho solução) e a distância total percorrida.

No caso da visualização do experimento, geramos uma rede com um número reduzido de vértices, um total de 100 vértices. Essa configuração permitirá uma análise mais clara e compreensível dos resultados obtidos

Grafo Utilizado para o experimento 1



### 3.1.1. Busca em profundidade (DFS)

Caminho percorrido: [31, 19, 47, 36, 24, 7, 20, 10, 22, 0, 21, 28, 92, 7, 37, 5, 62]

Distância final percorrida:

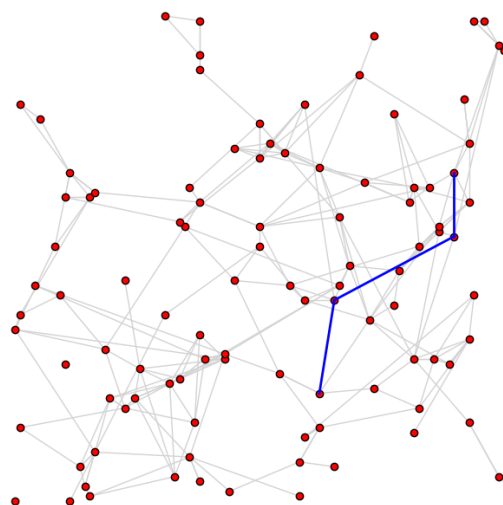
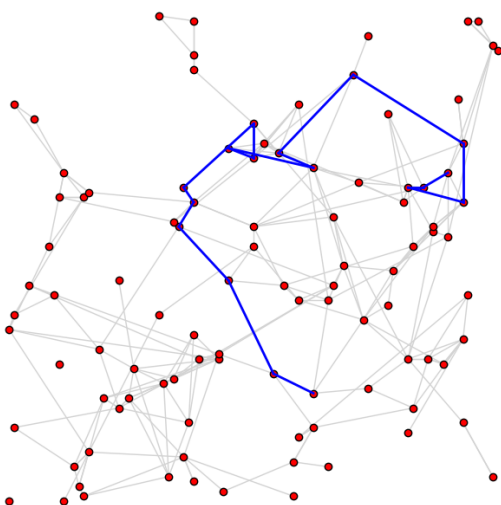
191.08202811947837

### 3.1.2. Busca em largura (BFS)

Caminho percorrido: [31, 89, 27, 62]

Distância final percorrida:

59.530072189583706

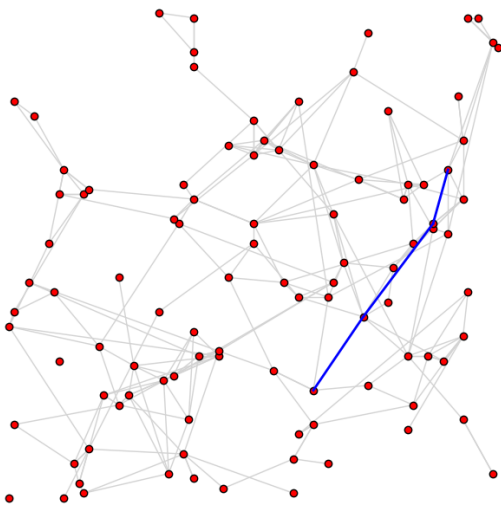


### 3.1.3. Algoritmo Best-First

Caminho percorrido: [31, 45, 91, 62]

Distância final percorrida:

53.03035807072322



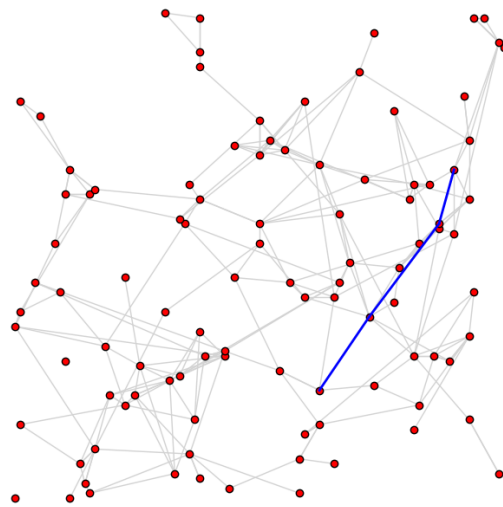
### 3.1.4. Algoritmo A\*

Caminho percorrido: [31, 45, 91, 62]

Distância final percorrida:

53.03035807072322

(nesse caso, pelo tamanho do grafo, o resultado se igualou ao Best-First)



## 3.2. Experimento 2

Para o tipo de rede selecionado pelo grupo, serão geradas redes com as seguintes configurações:

Rede Geográfica:

- a. ( $n=2000$ ,  $\lambda=0,01$ )
- b. ( $n=2000$ ,  $\lambda=0,02$ )
- c. ( $n=2000$ ,  $\lambda=0,03$ )

O valor de " $n$ " refere-se ao número total de vértices (ou nós) presentes na rede. No caso das configurações dadas, a rede terá 2000 vértices.

O valor de " $\lambda$ " representa a densidade da rede, ou seja, a probabilidade de existir uma aresta (conexão) entre dois vértices. Quanto maior o valor de  $\lambda$ , maior a probabilidade de haver conexões entre os vértices.

Será necessário selecionar, pelo menos, 10 pares distintos de vértices (origem e destino) e aplicar os algoritmos de busca. O objetivo é relatar a distância média percorrida e o tempo médio gasto por cada algoritmo. Essas métricas fornecerão informações sobre a eficiência e a qualidade dos algoritmos em relação à rede específica utilizada.

Para cada valor de  $\lambda$ , foi utilizado os mesmos 10 pares de vértices para ver como a mudança deste parâmetro altera a performance do algoritmo.

- $\lambda=0,01$

	DFS	BFS	Best-First	A*
Distância	170340	1238	1236	1266
Tempo	0.0031	0.250	0.0012	0.00033

- $\lambda=0,02$

	DFS	BFS	Best-First	A*
Distância	87583	1587	1589	1549
Tempo	0.0022	0.030	0.0010	0.00025

- $\lambda=0,03$

	DFS	BFS	Best-First	A*
Distância	38278	1555	1671	1614
Tempo	0.0023	0.019	0.0014	0.00040

A partir dos dados obtidos, temos que o DFS é aquele que percorre a maior distância de todos os algoritmos, o que é o esperado visto o funcionamento explicado anteriormente, e a distância diminui quanto maior  $\lambda$ .

Não houve grandes mudanças no tempo conforme a mudança de  $\lambda$ , exceto no BFS onde houve um ganho considerável de tempo ao aumentar o parâmetro.

Os algoritmos BFS, Best-First e A\* chegam no destino em distâncias similares, mas o A\* se demonstrou com a melhor performance, conseguindo o menor tempo.

### 3.3. Experimento 3

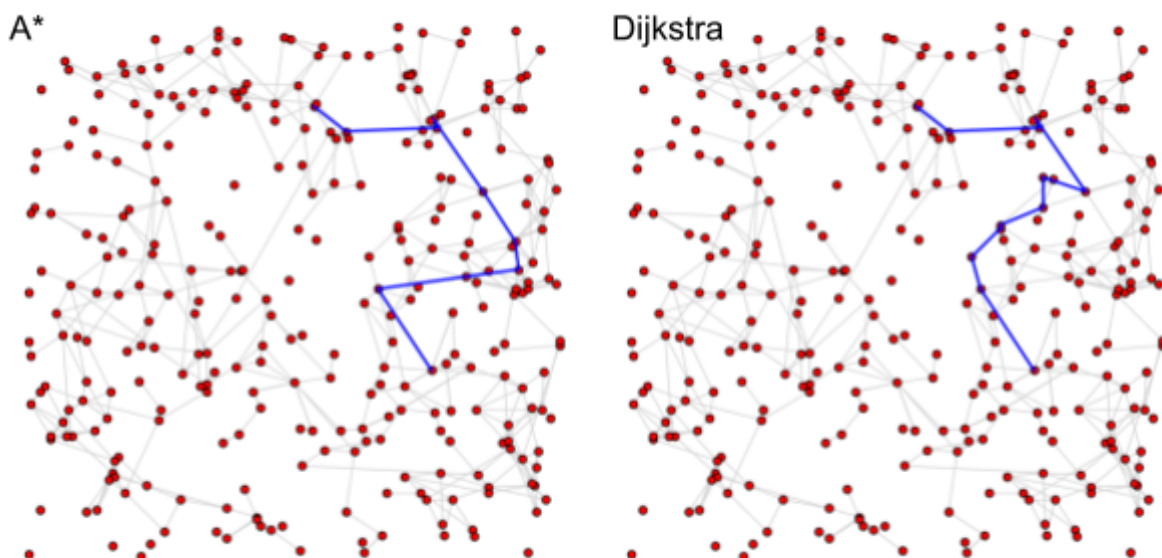


O último experimento consiste em realizar uma comparação entre dois algoritmos de busca: o algoritmo A\* e o algoritmo de Dijkstra. Essa comparação será feita utilizando as redes geradas no experimento anterior (Rede Geográfica com diferentes configurações).

O objetivo é avaliar e comparar o desempenho desses dois algoritmos na resolução de problemas de busca em cada uma das redes. Serão analisados fatores como a eficiência, a qualidade das soluções encontradas e o tempo de execução.

Essa comparação permitirá entender as diferenças entre o algoritmo A\* e o algoritmo de Dijkstra em termos de desempenho e capacidade de encontrar soluções ótimas. Os resultados obtidos serão utilizados para determinar qual algoritmo é mais adequado para a rede geográfica com as diferentes configurações utilizadas no experimento.

	Algoritmo A*	Algoritmo de Dijkstra
Tempo	0.00021049999986644252	0.003590169999824866
Caminho	37, 64, 243, 20, 213, 180, 110, 251, 163	37, 64, 243, 20, 213, 167, 165, 160, 46, 194, 251, 163
Distância	312.96805701124686	295.40194112161214



A partir dos experimentos diferença fundamental entre o algoritmo A\* e o algoritmo de Dijkstra está na forma como eles exploram o espaço de busca e tomam decisões para encontrar o caminho mais curto entre dois pontos em um grafo. As principais diferenças que podem ser descritas entre eles são:

- Heurística:
  - O algoritmo A\* utilizará a Distância Euclidiana como heurística para orientar a busca em direção ao destino desejado. A distância euclidiana fornece uma estimativa do custo restante para alcançar o objetivo a partir de cada nó. Essa heurística permite que o algoritmo A\* faça escolhas informadas, priorizando nós que estão mais próximos ao destino em termos de distância euclidiana.
  - Já o algoritmo de Dijkstra escolhe o nó com o menor custo atual a partir do ponto de partida. Ele expande gradualmente para os nós vizinhos, atualizando os custos acumulados ao longo do caminho. O critério de escolha do nó no algoritmo de Dijkstra é puramente baseado no custo acumulado até o momento.

## 4. Bibliografia

BOARDINFINITY. Difference Between BFS and DFS. Disponível em: <https://www.boardinfinity.com/blog/difference-between-bfs-and-dfs/>  
Acesso em: 20 maio 2023.

IEEE TRANSACTIONS ON GEOSCIENCE AND REMOTE SENSING. Disponível em: <https://ieeexplore.ieee.org/xpl/RecentIssue.jsp?punumber=36>.  
Acesso em: 20 maio 2023

"Geographic Information Science and Systems" por Paul A. Longley, Michael F. Goodchild, David J. Maguire, e David W. Rhind.

SHAH, Y. Astar-Algorithm-for-ACME-Robotics. Disponível em: <https://github.com/ysshah95/Astar-Algorithm-for-ACME-Robotics>.  
Acesso em: 20 maio 2023.

OPENGENUS IQ. Best First Search. Em: OpenGenus IQ. Disponível em: <https://iq.opengenus.org/best-first-search/>. Acesso em: 20 maio 2023.

WIKIPEDIA. Dijkstra's algorithm. In: Wikipedia: The Free Encyclopedia. Disponível em: [https://en.wikipedia.org/wiki/Dijkstra%27s\\_algorithm](https://en.wikipedia.org/wiki/Dijkstra%27s_algorithm). Acesso em: 20 maio 2023.