

Grafos

**Davi Lopes Lemos¹,
Heduardo Witkoski Barcelos da Rocha¹**

¹Campus Bagé - Universidade Federal do Pampa (UNIPAMPA)
96.413-172 – Bagé – RS – Brazil

{davilemos, heduardorocha}.aluno@unipampa.edu.br

1. Escopo do experimento

Analisar <as formas de representação e grafos (LA e MA) e tipos de grafos (E e D) para o propósito de <avaliação> com respeito ao <tempo de execução de implementações em Java dos algoritmos de menor caminho de fonte única de Bellman-Ford e de Dijkstra> do ponto de vista do(a) <programador> no contexto de <estudos sobre Grafos no componente de Algoritmos e Classificação de Dados>.

2. Introdução

A análise de algoritmos desempenha um papel fundamental na computação, fornecendo subsídios para compreender a eficiência e a aplicabilidade de diferentes abordagens em diversos contextos. Conforme discutido por Cormen et al. [2024], a escolha de estruturas de dados, como as representações de grafos por lista ou matriz de adjacência, é determinante para o desempenho de algoritmos em cenários específicos, como grafos densos e esparsos. Este trabalho explora a implementação de algoritmos clássicos de grafos, aplicados a grafos representados por listas e matrizes de adjacência.

Os algoritmos implementados incluem os métodos de menor caminho de origem única, Bellman-Ford e Dijkstra, amplamente utilizados para calcular a menor distância entre um vértice inicial e os demais em grafos ponderados. O algoritmo Bellman-Ford utiliza uma abordagem iterativa para relaxar todas as arestas repetidamente, sendo adequado para grafos com pesos negativos. Já o algoritmo de Dijkstra emprega uma estratégia gulosa com o uso de filas de prioridade, garantindo eficiência em grafos com pesos não negativos.

Os experimentos realizados visam avaliar o impacto da escolha da representação de grafos (lista de adjacência ou matriz de adjacência) no desempenho temporal e na ocupação de memória de cada algoritmo. A análise considera diferentes tamanhos de grafos e densidades, desde cenários esparsos até densos, buscando identificar variações significativas nas métricas de execução. Adicionalmente, é realizada uma breve análise do consumo de memória das implementações, permitindo um entendimento mais abrangente das implicações práticas de cada abordagem.

O principal objetivo deste trabalho é validar ou refutar a hipótese de que as representações de grafos afetam significativamente o desempenho dos algoritmos, com foco

nos tempos médios de execução e memória utilizada. Para isso, grafos sintéticos foram gerados com combinações variadas de vértices e arestas, e os resultados serão apresentados e discutidos com base em métricas específicas, como a média e a distribuição dos tempos de execução.

3. Metodologia

A presente pesquisa experimental visa avaliar o desempenho dos algoritmos de menor caminho de origem única, Bellman-Ford e Dijkstra, analisando especificamente o impacto das diferentes representações de grafos (listas de adjacência e matrizes de adjacência) sobre o tempo de execução. Os algoritmos foram implementados em Java, utilizando os princípios da orientação a objetos, como o encapsulamento em classes e métodos, e estruturas de controle como `for` e `while` para iterar sobre os vértices e arestas. A medição precisa do tempo de execução foi realizada através do método `nanotime()` da classe `System`, o que permitiu capturar os tempos de início e término de cada execução com alta precisão, possibilitando o cálculo do tempo de execução de maneira detalhada e consistente.

Para os experimentos, foram gerados grafos sintéticos, variando em tamanho (número de vértices) e densidade (relação entre arestas e vértices), abrangendo desde grafos esparsos até densos. Cada configuração de grafo foi processada dez vezes para assegurar a confiabilidade e a repetibilidade dos resultados. As duas representações de grafos, listas de adjacência e matrizes de adjacência, foram analisadas separadamente, permitindo uma comparação rigorosa dos tempos de execução e do uso de memória em cada cenário. Através dessa abordagem, foi possível observar como as diferentes representações influenciam diretamente no desempenho dos algoritmos em termos de tempo e uso de recursos computacionais.

A análise do comportamento dos algoritmos também levou em consideração as características distintas de suas implementações. O Bellman-Ford adota uma abordagem iterativa, onde cada aresta é verificada repetidamente para relaxar os vértices, sendo ideal para grafos com pesos negativos, enquanto o Dijkstra utiliza uma estratégia baseada em uma fila de prioridade, onde os vértices com as menores distâncias acumuladas são selecionados, sendo restrito a grafos com pesos não negativos. A pesquisa investigou, portanto, de que maneira a escolha da representação de grafos impacta o desempenho dos algoritmos, tanto em termos de tempo de execução quanto de memória utilizada, considerando diferentes configurações de entrada e a diversidade das características dos grafos testados.

Para verificar a hipótese sobre a eficiência das representações por Lista de Adjacência (LA) e Matriz de Adjacência (MA) em grafos densos e esparsos, foram realizados experimentos com 15 grafos gerados pelo programa GTgraph-random (GTgraph, 2024). Esses grafos variaram em número de vértices (100, 200, 500) e densidade (número de arestas), resultando em 15 combinações distintas. Os tempos de execução médios dos algoritmos Bellman-Ford e Dijkstra foram medidos para cada combinação, considerando ambas as representações, LA e MA.

Os dados coletados foram analisados por meio de scripts desenvolvidos em Python, utilizando bibliotecas como `pandas` e `matplotlib` para processar e visualizar os resultados. Para ilustrar a distribuição dos tempos de execução, foram utilizados boxplots, proporcionando uma visão detalhada sobre as variações e dispersões dos dados. Além

disso, gráficos complementares, foram utilizados para analisar mais profundamente a relação entre o tempo de execução e as características dos grafos, permitindo uma comparação detalhada entre as diferentes representações e densidades de grafos.

Os experimentos foram conduzidos em um ambiente de execução padronizado, com um sistema baseado no processador AMD Ryzen 7 5700U, 12 GB de memória DDR4 e o sistema operacional Ubuntu 24.04 LTS. Este ambiente foi escolhido para garantir a consistência dos resultados, minimizando as variações causadas por fatores externos e assegurando a precisão das medições durante os testes.

3.1. Experimento

"Na implementação de estruturas de dados para Grafos, as representações de um grafo G por Lista de Adjacência (LA) e por Matriz de Adjacência (MA) proporcionam desempenhos semelhantes (em relação às médias de tempo de execução) para processamento de grafos densos (D) ou esparsos (E), especificamente na execução de implementações dos algoritmos de menor caminho de fonte única de Bellman-Ford e de Dijkstra?"

Para investigar essa questão, foram desenvolvidas versões de ambos os algoritmos utilizando as duas representações, e os testes foram realizados em grafos gerados com diferentes características de densidade. A análise incluiu medições sistemáticas dos tempos de execução, buscando identificar cenários em que uma estrutura pudesse se destacar em termos de eficiência computacional. O experimento é conduzido em torno dessa pergunta, com o objetivo de compreender a relação entre a densidade dos grafos e a escolha da representação mais adequada.

4. Resultados

A partir da aplicação dos materiais e métodos descritos na Seção 3, viabilizou-se a geração de gráficos que permitiram uma análise detalhada dos tempos de execução para cada cenário. No que tange os gráficos de caixa (Figura 1), estes fornecem uma visão detalhada sobre a variabilidade e a distribuição dos tempos de execução dos algoritmos Bellman-Ford e Dijkstra, utilizando tanto a LA quanto a MA. Cada gráfico apresenta a dispersão dos dados, destacando a mediana, os quartis e os valores atípicos, permitindo uma comparação visual entre as representações de grafo analisadas. Observa-se que, para o algoritmo Bellman-Ford, a matriz de adjacência apresenta maior variabilidade nos tempos de execução, especialmente para grafos maiores, enquanto a lista de adjacência exibe uma distribuição mais concentrada e menor variabilidade, especialmente em grafos esparsos. Já para o algoritmo Dijkstra, embora a matriz de adjacência mostre uma menor dispersão, o tempo de execução tende a aumentar mais rapidamente conforme o número de vértices e arestas cresce, particularmente em grafos densos. Esses gráficos ajudam a ilustrar as diferenças de desempenho e a destacar como a escolha da representação do grafo pode impactar a eficiência dos algoritmos.

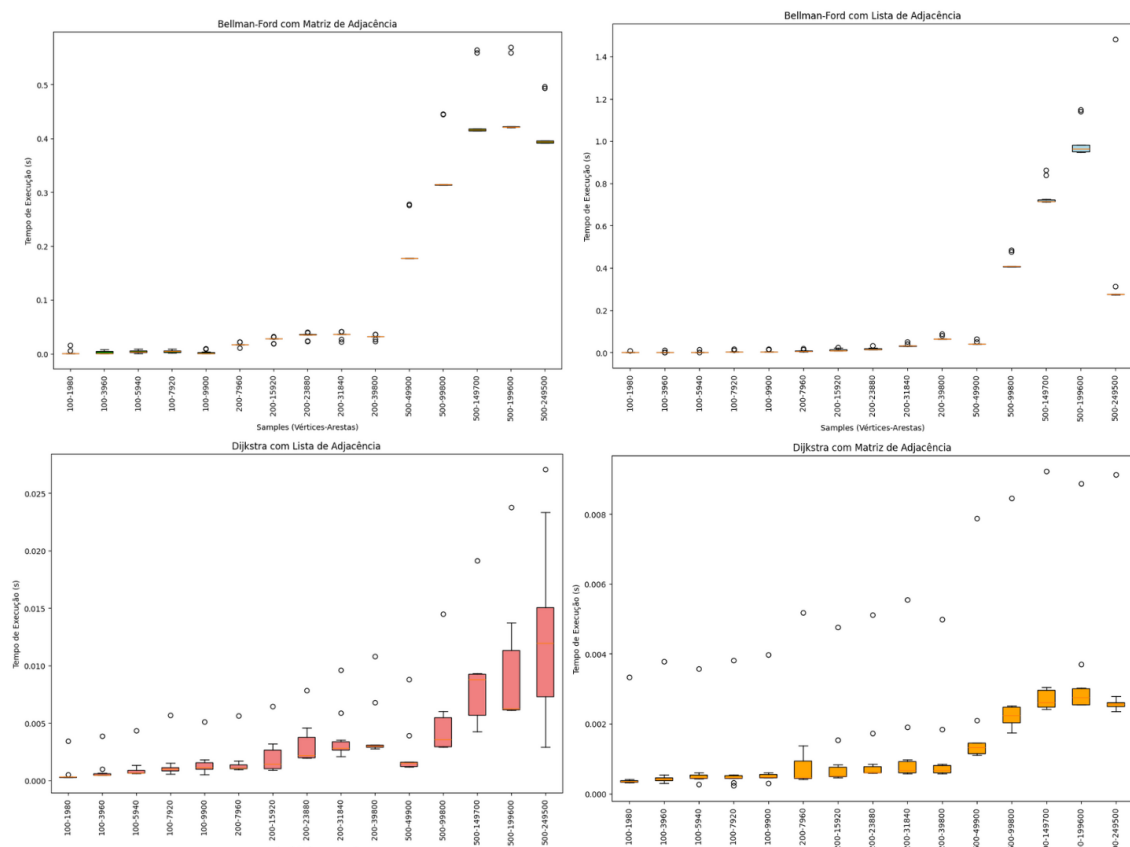


Figura 1. Gráficos de caixa

Adicionalmente, nos gráficos de tempos de execução médios para 100, 200 e 500 vértices (Figura 2), observa-se o desempenho dos algoritmos Bellman-Ford e Dijkstra utilizando listas de adjacência e matrizes de adjacência. Com relação à Hipótese H_0' , que propõe que os tempos médios de execução para os algoritmos Bellman-Ford e Dijkstra são semelhantes entre as representações por LA e MA, tanto para grafos esparsos quanto para grafos densos, esta só se prova válida no algoritmo de Dijkstra, enquanto o algoritmo de Bellman-Ford não confirma esta hipótese, visto que seus dados indicam que existem diferenças significativas entre os tempos médios de execução das representações LA e MA, tanto para grafos densos e esparsos. Desta forma, o resultados para Bellman-Ford confirmam a Hipótese Alternativa (HA').

Em relação à Hipótese H_0'' , que sugere que a densidade do grafo (esparso x denso) não afeta significativamente o desempenho médio dos algoritmos Bellman-Ford e Dijkstra, independentemente da representação LA ou MA, os resultados indicam que a densidade do grafo afeta significativamente o desempenho dos algoritmos. Para o Bellman-Ford, a LA é mais eficiente para grafos esparsos, enquanto a MA mostra maior eficiência para grafos densos. Para o Dijkstra, a MA tem desempenho significativamente melhor em grafos densos, enquanto a LA apresenta desempenho superior em grafos esparsos. Esses achados confirmam a Hipótese Alternativa (HA''), sugerindo que a densidade do grafo influencia significativamente o desempenho dos algoritmos, tanto em LA quanto em MA.

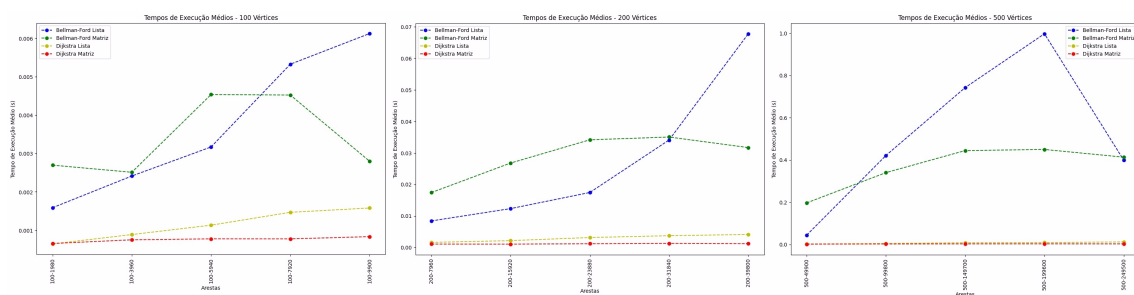


Figura 2. Gráficos com média de tempo de execução

Os resultados experimentais também indicam que o algoritmo Dijkstra é mais eficiente que o Bellman-Ford para a maioria dos casos, especialmente em grafos densos. Além disso, a LA é mais eficiente para grafos esparsos, enquanto a MA é mais eficiente para grafos densos. Esses achados corroboram as Hipóteses Alternativas (HA' e HA''), demonstrando que a forma de representação do grafo e o tipo de grafo (denso ou esparsos) afetam significativamente o desempenho dos algoritmos de menor caminho de fonte única. A análise confirma a teoria de Cormen et al. (2024), destacando a importância de escolher a representação de grafo adequada para otimizar o desempenho dos algoritmos.

5. Conclusão

Os resultados obtidos neste estudo destacam a relevância da escolha da representação de grafos no desempenho dos algoritmos de menor caminho de origem única, Bellman-Ford e Dijkstra, considerando diferentes configurações de densidade e tamanho dos grafos. Durante os experimentos, observou-se que as representações por Lista de Adjacência (LA) e Matriz de Adjacência (MA) apresentam comportamentos distintos dependendo do algoritmo utilizado e das características do grafo.

Para o algoritmo Bellman-Ford, a Lista de Adjacência mostrou-se mais eficiente em grafos esparsos, apresentando tempos de execução menores e menor variabilidade nos dados. Em contrapartida, a Matriz de Adjacência demonstrou maior eficiência em grafos densos, apesar de uma variabilidade mais acentuada nos tempos de execução. Esses achados indicam que o desempenho do algoritmo Bellman-Ford é fortemente impactado pela escolha da representação de grafo, refutando a hipótese inicial de que ambas as representações teriam desempenhos semelhantes em diferentes tipos de grafos.

O algoritmo Dijkstra, por sua vez, apresentou um comportamento mais consistente, com a Matriz de Adjacência se destacando em grafos densos e a Lista de Adjacência mostrando melhor desempenho em grafos esparsos. Além disso, os tempos de execução do Dijkstra foram, de maneira geral, inferiores aos do Bellman-Ford, confirmando sua maior eficiência em grafos com pesos não negativos. Esses resultados reforçam a importância de alinhar a escolha da representação de grafos às características específicas do problema, como densidade e tamanho, para otimizar o desempenho do algoritmo utilizado.

A análise também revelou que a densidade do grafo afeta significativamente o desempenho dos algoritmos, independentemente da representação adotada. Para grafos

esparsos, a Lista de Adjacência foi consistentemente mais eficiente, enquanto, em grafos densos, a Matriz de Adjacência apresentou melhor desempenho, especialmente no algoritmo de Dijkstra. Esses resultados corroboram a teoria de Cormen et al. (2024), que aponta a influência crucial da estrutura de dados no comportamento de algoritmos de grafos.

Em síntese, o estudo demonstrou que a escolha da representação de grafos deve ser cuidadosamente considerada em função das características do grafo e do algoritmo empregado. Enquanto a Lista de Adjacência se destaca em cenários de grafos esparsos, a Matriz de Adjacência é mais adequada para grafos densos, sendo que essas escolhas impactam diretamente a eficiência computacional, tanto em termos de tempo de execução quanto de uso de memória. Esses achados não apenas validam a hipótese de que a densidade e a representação do grafo influenciam significativamente o desempenho dos algoritmos, mas também fornecem subsídios valiosos para a tomada de decisões em aplicações práticas envolvendo grafos.

Referências

Cormen, T. H., Leiserson, C. E., Rivest, R. L., and Stein, C. (2024). *Algoritmos: Teoria e Prática*. GEN LTC, Rio de Janeiro, 4 edition.