Universidade Federal Do Rio Grande do Sul
Instituto de Informática

Lucas Dinesh Weber Miranda

Trabalho 4 (Emparelhamentos)

Porto Alegre, 25 de maio de 2025

1. Introdução

Este relatório apresenta uma análise experimental do algoritmo de emparelhamento perfeito de peso mínimo em grafos bipartidos, utilizando o algoritmo Húngaro com busca de caminhos M-aumentantes mais curtos, implementado via transformação de Johnson e busca com Dijkstra.

O objetivo foi verificar na prática o comportamento do algoritmo, especialmente sua complexidade teórica O(n(m + n log n)), por meio da análise do tempo de execução em função do tamanho do grafo, assim como das métricas associadas a cada etapa do algoritmo.

---

2. Implementação

Toda a implementação do código foi realizada em C++, utilizando `vector` para a representação da matriz de custos e das variáveis auxiliares do algoritmo.

Adicionalmente, foi desenvolvido um conjunto de scripts em Python, utilizando as bibliotecas `pandas` e `matplotlib` para:

- **Automatizar a execução de testes em lote.**
- **Coletar métricas de desempenho e resultados.**
- **Gerar gráficos para análise experimental.**

---

3. Ambiente de Teste

Os testes foram realizados em uma máquina Dell G15, com:

- **8GB de memória RAM.**
- **1TB de armazenamento.**
- **Sistema operacional Linux Ubuntu 22.04 LTS (dual boot).**
- **IDE utilizada: Visual Studio Code (VSCode).**

Compilação realizada com `g++` e a flag `-O2` para otimização.

---

4. Metodologia

Para avaliar o desempenho do algoritmo, realizei diversos testes variando o tamanho do grafo bipartido, seguindo os arquivos disponibilizados pelo professor no repositório oficial:

https://github.com/mrpritt/epm_bipartido/tree/main/data

Os tamanhos de grafos testados foram:
$n = [10, 20, 50, 100, 200, 500]$.

As métricas avaliadas foram:

- **Tempo total de execução.**
- **Tempo médio por iteração do laço principal.**
- **Tempo médio por busca de caminho M-aumentante.**
- **Resultado do emparelhamento perfeito de peso mínimo.**

---

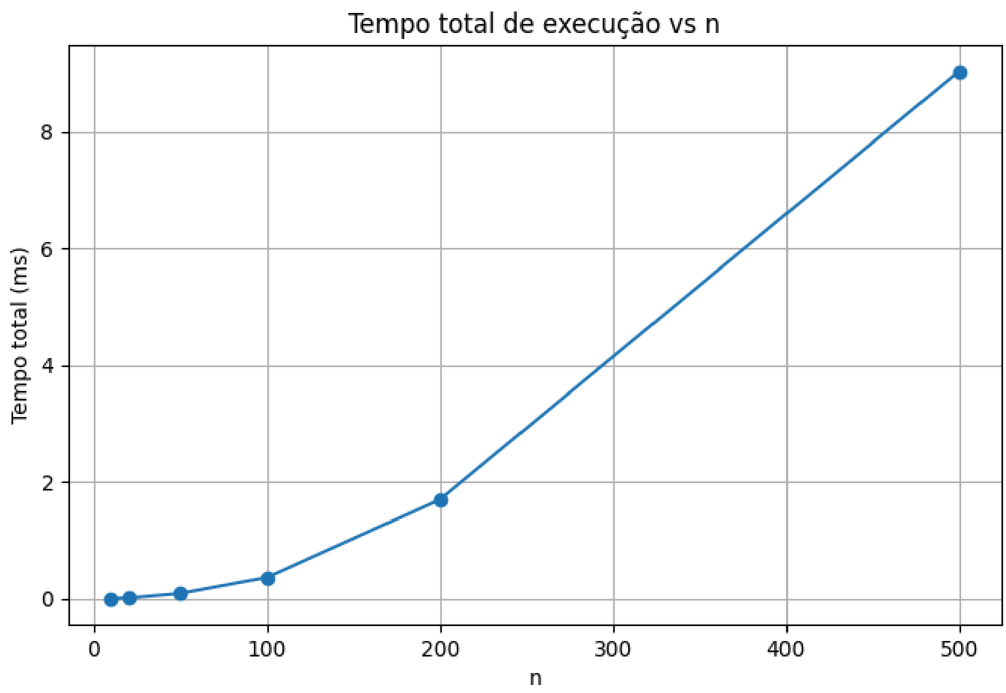
5. Análise Experimental e Desempenho Geral

Os resultados coletados encontram-se na tabela abaixo:

n	Tempo total (ms)	Tempo médio iteração (ms)	Tempo médio busca (ms)	Resultado
10	0.004	0.0004	0.0001	104
20	0.010	0.0005	0.00015	589

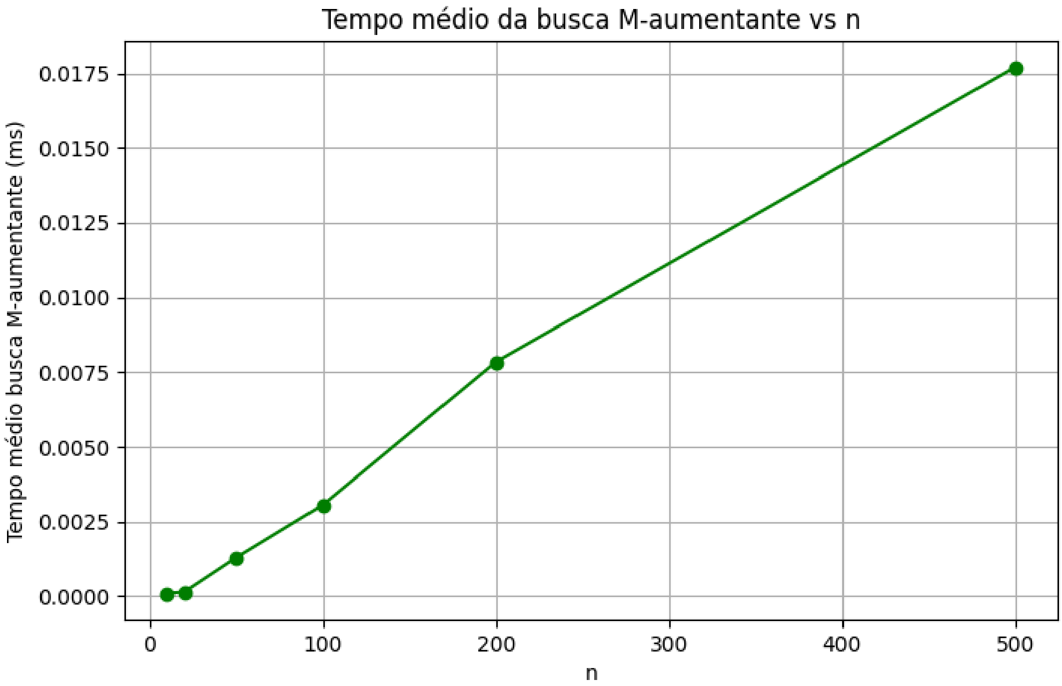
50	0.088	0.00176	0.0013	3402
100	0.361	0.00361	0.00306	16517
200	1.696	0.00848	0.007845	62884
500	9.040	0.01808	0.017706	410036

Gráfico 1 – Tempo total de execução vs n



O gráfico mostra que o tempo total de execução cresce rapidamente conforme o tamanho do grafo aumenta, sugerindo um comportamento compatível com a complexidade cúbica prevista pela teoria (O(n^3)), uma vez que em grafos bipartidos completos temos m = n^2.

Gráfico 2 – Tempo médio da busca M-aumentante vs n



O tempo médio por busca de caminho M-aumentante também cresce de forma aproximadamente linear com n, o que é esperado, dado que cada busca envolve analisar O(n) vértices e arcos, considerando o uso da transformação de Johnson com Dijkstra.

Discussão

- **A relação entre o tempo total e o tamanho do grafo evidencia a necessidade do uso de métodos eficientes para busca de caminhos aumentantes, como a abordagem com transformação de Johnson implementada.**

- **O tempo médio da busca M-aumentante se manteve próximo do tempo médio por iteração, demonstrando que a maior parte do custo está, de fato, na realização dessas buscas.**

---

6. Conclusão

Com base nos resultados obtidos, conclui-se que o algoritmo Húngaro implementado com busca de caminhos M-aumentantes via transformação de Johnson e Dijkstra apresentou desempenho compatível com a complexidade teórica O(n(m + n log n)), que, no caso de grafos bipartidos completos, resulta em O(n^3).

O uso de scripts para automação e análise gráfica foi fundamental para garantir a precisão e a clareza dos resultados experimentais.

O comportamento do tempo de execução reforça a importância de abordagens eficientes para problemas de emparelhamento em grafos grandes, especialmente na prática, onde instâncias de centenas ou milhares de vértices podem ser comuns.