Árvore Geradora Mínima Generalizada

Autor:
Pedro Lucas Botelho Freitas

Disciplina:
DCC059 - Teoria dos Grafos

JUIZ DE FORA
2025

Sumário

1 Descrição do Problema

O problema da **Árvore Geradora Mínima Generalizada** (AGMG) consiste em, dado um grafo $G = (V, E)$ cujos vértices estão particionados em k clusters, selecionar exatamente um vértice de cada cluster e construir uma árvore que conecte os vértices escolhidos de forma que o custo total (soma dos pesos das arestas utilizadas) seja mínimo.

1.1 Definição Formal

Seja $G = (V, E)$ um grafo e seja a partição de V dada por

$$V = V_1 \cup V_2 \cup \cdots \cup V_k, \quad \text{com } V_i \cap V_j = \varnothing \text{ para } i \neq j.$$

O objetivo é selecionar um vértice $v_i \in V_i$, para cada $i = 1, \ldots, k$, e determinar uma árvore geradora T sobre o conjunto $\{v_1, v_2, \ldots, v_k\}$, de forma que:

$$\text{custo}(T) = \sum_{(u,v)\in T} w(u, v)$$

seja mínimo.

1.2 Complexidade Computacional

Embora o problema clássico da árvore geradora mínima (resolvido por algoritmos de Prim ou Kruskal) seja solucionável em tempo polinomial, a restrição de escolher um vértice por cluster torna o problema NP-difícil. Isso implica que, para instâncias de grande escala, a obtenção de soluções exatas torna-se inviável, o que motiva a adoção de métodos heurísticos e metaheurísticos.

1.3 Desafios

Entre os principais desafios estão:

- **Explosão combinatória:** A escolha de um vértice para cada cluster implica um número exponencial de combinações possíveis.
- **Interdependência:** A seleção de um vértice afeta diretamente a qualidade da árvore geradora obtida.
- **Balanceamento:** Obter um equilíbrio entre a qualidade (custo total mínimo) e o tempo computacional, especialmente em grafos grandes e complexos.

1.4 Objetivo

O trabalho tem como objetivo implementar e comparar três abordagens heurísticas para resolver a AGMG:

1. **Algoritmo Guloso:** Seleciona iterativamente a aresta de menor custo que conecta um vértice de um cluster ainda não representado a um já incluído na solução.
2. **Algoritmo Randomizado:** Introduz aleatoriedade na escolha dos vértices e arestas, buscando escapar de mínimos locais e diversificar as soluções.
3. **Algoritmo Reativo:** Combina as estratégias gulosa e randomizada de forma adaptativa, ajustando os parâmetros durante a execução com base no desempenho.

2 Descrição das Instâncias

Foram utilizadas dez instâncias de grafos, cujas características são as seguintes

- **Instância 1:** 5000 vértices, 10.000 arestas, 1000 clusters, **não direcionado**.
- **Instância 2:** 6000 vértices, 15.000 arestas, 500 clusters, **direcionado**.
- **Instância 3:** 7000 vértices, 20.000 arestas, 3000 clusters, **não direcionado**.
- **Instância 4:** 8000 vértices, 25.000 arestas, 1500 clusters, **direcionado**.
- **Instância 5:** 9000 vértices, 30.000 arestas, 1800 clusters, **não direcionado**.
- **Instância 6:** 5500 vértices, 12.000 arestas, 1200 clusters, **não direcionado**.
- **Instância 7:** 6500 vértices, 5000 arestas, 800 clusters, **direcionado**.
- **Instância 8:** 7566 vértices, 22.222 arestas, 5 clusters, **não direcionado**.
- **Instância 9:** 8665 vértices, 27.654 arestas, 3100 clusters, **direcionado**.
- **Instância 10:** 9500 vértices, 35.000 arestas, 900 clusters, **não direcionado**.

3 Descrição dos Métodos Implementados

Foram desenvolvidas três abordagens para a resolução da AGMG, conforme descrito a seguir.

3.1 Algoritmo Guloso

Esta abordagem constrói a solução de forma iterativa. A cada iteração, seleciona-se a aresta de menor custo que conecta um vértice pertencente a um cluster ainda não representado a um vértice já incluído na árvore. Embora simples e com tempo de execução reduzido, o método pode ficar preso em mínimos locais.

3.2 Algoritmo Randomizado

O método randomizado introduz um componente estocástico na seleção dos vértices e arestas. Ao escolher aleatoriamente entre as opções candidatas (por exemplo, dentre as arestas de menor custo), o algoritmo explora diferentes regiões do espaço de soluções. Essa abordagem pode resultar em soluções de custo maior, mas aumenta a diversidade e a chance de escapar de mínimos locais.

3.3 Algoritmo Reativo

O algoritmo reativo integra as estratégias gulosa e randomizada de forma adaptativa. Durante a execução, os parâmetros de escolha (por exemplo, a probabilidade de adotar a estratégia gulosa) são ajustados com base no desempenho observado. Assim, o método busca combinar a eficiência do guloso com a capacidade exploratória do randomizado, embora com um custo computacional adicional.

3.4 Estrutura e Modularidade do Código

O código, desenvolvido no arquivo `grafo.cpp`, foi organizado em módulos, cada um dedicado a uma das estratégias. Essa modularização facilita a manutenção e futuras extensões, permitindo que cada abordagem seja testada e aprimorada de forma isolada.

3.5 Gerenciamento de Memória

Foram utilizadas estruturas dinâmicas (como listas ligadas e vetores) para a representação dos grafos, garantindo a alocação e liberação eficientes de memória, especialmente em instâncias de grande porte.

3.6 Abordagem Orientada a Objetos e Clareza

A implementação segue os princípios da programação orientada a objetos, com classes bem definidas para representar vértices, arestas e clusters. Comentários detalhados (compatíveis com Doxygen) foram inseridos para facilitar a compreensão e a documentação do código.

4 Análise de Tempo de Execução entre Lista e Matriz

Para avaliar a eficiência das estruturas de dados, realizou-se uma comparação entre a representação por matriz de adjacência e por lista de adjacência, utilizando como exemplo a Instância 1. Os resultados obtidos foram:

4.1 Matriz de Adjacência

- **Guloso:** custo total = 7044, tempo de execução = 0,19 s.
- **Randomizado:** custo total = 50042, tempo de execução = 0,19 s.
- **Reativo:** custo total = 7044, tempo de execução = 3,66 s.

4.2 Lista de Adjacência

- **Guloso:** custo total = 7044, tempo de execução = 0,14 s.
- **Randomizado:** custo total = 51503, tempo de execução = 0,13 s.
- **Reativo:** custo total = 7044, tempo de execução = 2,22 s.

Figura 1: Comparação do custo total médio entre os algoritmos.

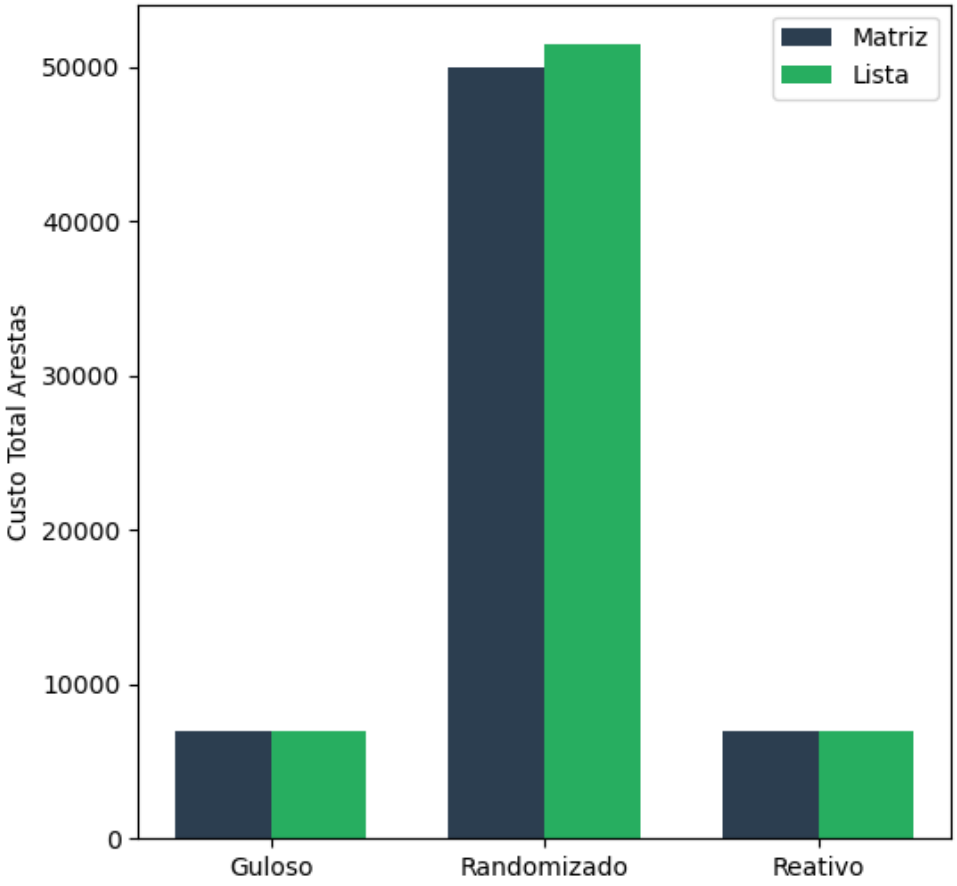
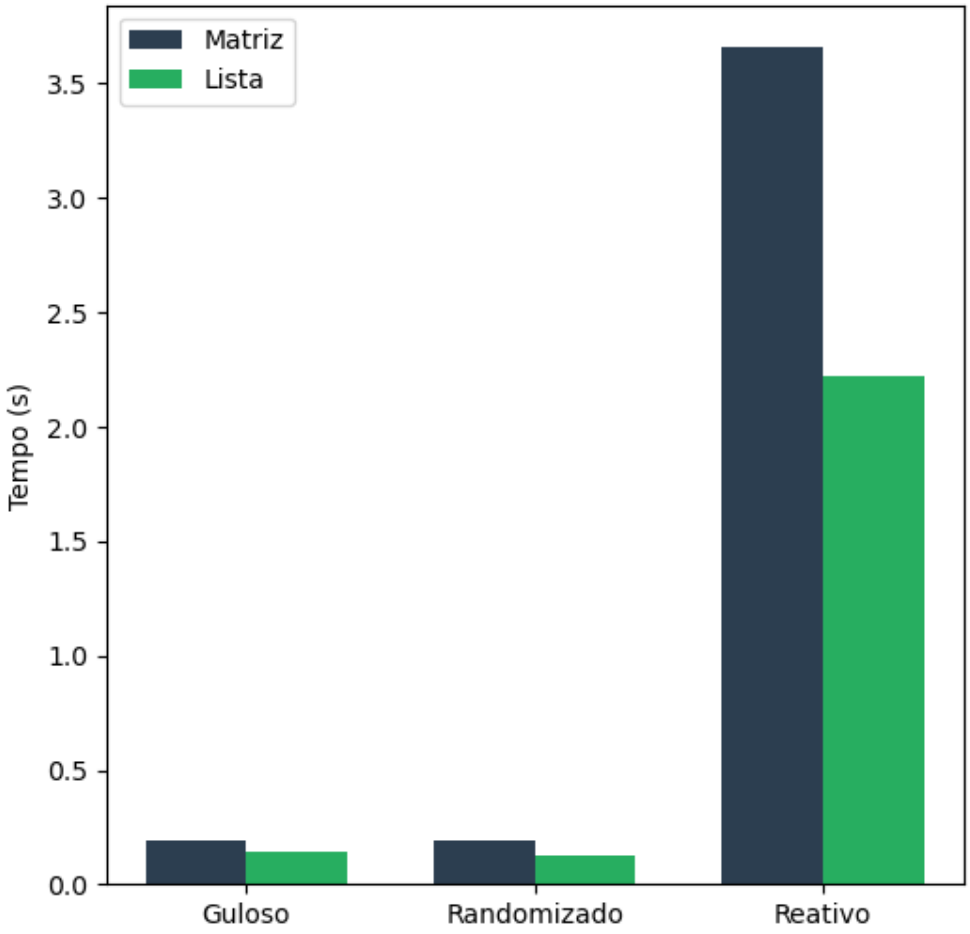


Figura 2: Comparação do tempo médio de execução entre os algoritmos.



4.3 Discussão

Observa-se que a utilização da lista de adjacência resulta em uma redução notável no tempo de execução, especialmente para o algoritmo reativo. Isso se deve à eficiência na manipulação de grafos esparsos, onde o acesso direto às arestas é mais otimizado em uma estrutura dinâmica do que em uma matriz de adjacência.

5 Análise de Resultado com Teste de Hipótese entre os Métodos

Nesta seção são apresentados os resultados experimentais obtidos para cada instância, o resumo estatístico e os testes de hipótese realizados sobre o custo total das arestas e o tempo

de execução.

5.1 Resultados por Instância

Instância 1

Tabela 1: Resultados para a Instância 1

Algoritmo	Custo Total Arestas	Tempo (s)
Guloso	7044	0,19
Randomizado	50042	0,19
Reativo	7044	3,66

Instância 2

Tabela 2: Resultados para a Instância 2

Algoritmo	Custo Total Arestas	Tempo (s)
Guloso	1274	0,28
Randomizado	25625	0,25
Reativo	1274	5,11

Instância 3

Tabela 3: Resultados para a Instância 3

Algoritmo	Custo Total Arestas	Tempo (s)
Guloso	33309	0,38
Randomizado	151881	0,36
Reativo	33309	7,14

Instância 4

Tabela 4: Resultados para a Instância 4

Algoritmo	Custo Total Arestas	Tempo (s)
Guloso	5970	0,49
Randomizado	74692	0,50
Reativo	5970	9,25

Instância 5

Tabela 5: Resultados para a Instância 5

Algoritmo	Custo Total Arestas	Tempo (s)
Guloso	8054	0,63
Randomizado	88887	0,62
Reativo	8054	11,79

Instância 6

Tabela 6: Resultados para a Instância 6

Algoritmo	Custo Total Arestas	Tempo (s)
Guloso	8730	0,24
Randomizado	61104	0,24
Reativo	8730	4,43

Instância 7

Tabela 7: Resultados para a Instância 7

Algoritmo	Custo Total Arestas	Tempo (s)
Guloso	8179	0,30
Randomizado	40695	0,31
Reativo	8179	5,94

Instância 8

Tabela 8: Resultados para a Instância 8

Algoritmo	Custo Total Arestas	Tempo (s)
Guloso	4	0,42
Randomizado	121	0,42
Reativo	4	8,02

Instância 9

Tabela 9: Resultados para a Instância 9

Algoritmo	Custo Total Arestas	Tempo (s)
Guloso	22001	0,57
Randomizado	157410	0,54
Reativo	22001	10,85

Instância 10

Tabela 10: Resultados para a Instância 10

Algoritmo	Custo Total Arestas	Tempo (s)
Guloso	1831	0,72
Randomizado	46015	0,67
Reativo	1831	13,26

5.2 Resumo Estatístico

Para uma visão global, calcularam-se a média (μ) e o desvio padrão (σ) dos resultados considerando as dez instâncias:

Tabela 11: Resumo Estatístico dos Resultados

Métrica	Guloso	Randomizado	Reativo
Custo Total Arestas	9639.6 ± 9809	69647.2 ± 48463	9639.6 ± 9809
Tempo (s)	0.422 ± 0.168	0.41 ± 0.158	7.945 ± 3.27

5.3 Testes de Hipótese

Para avaliar a significância dos resultados, foram realizados os seguintes testes:

- **Custo Total Arestas:**
  - Guloso vs. Randomizado: $p < 0,001$ (diferença significativa).
  - Guloso vs. Reativo: $p = 1,0$ (sem diferença significativa).
  - Randomizado vs. Reativo: $p < 0,001$ (diferença significativa).

- **Tempo de Execução:**
  - Guloso vs. Reativo: $p < 0,001$ (diferença significativa).
  - Randomizado vs. Reativo: $p < 0,001$ (diferença significativa).
  - Guloso vs. Randomizado: $p = 1,0$ (sem diferença significativa).

5.4 Visualização dos Resultados

As Figuras 3 e 4 ilustram, respectivamente, a comparação do custo total médio e do tempo médio de execução entre os algoritmos.

Figura 3: Comparação do custo total médio entre os algoritmos.

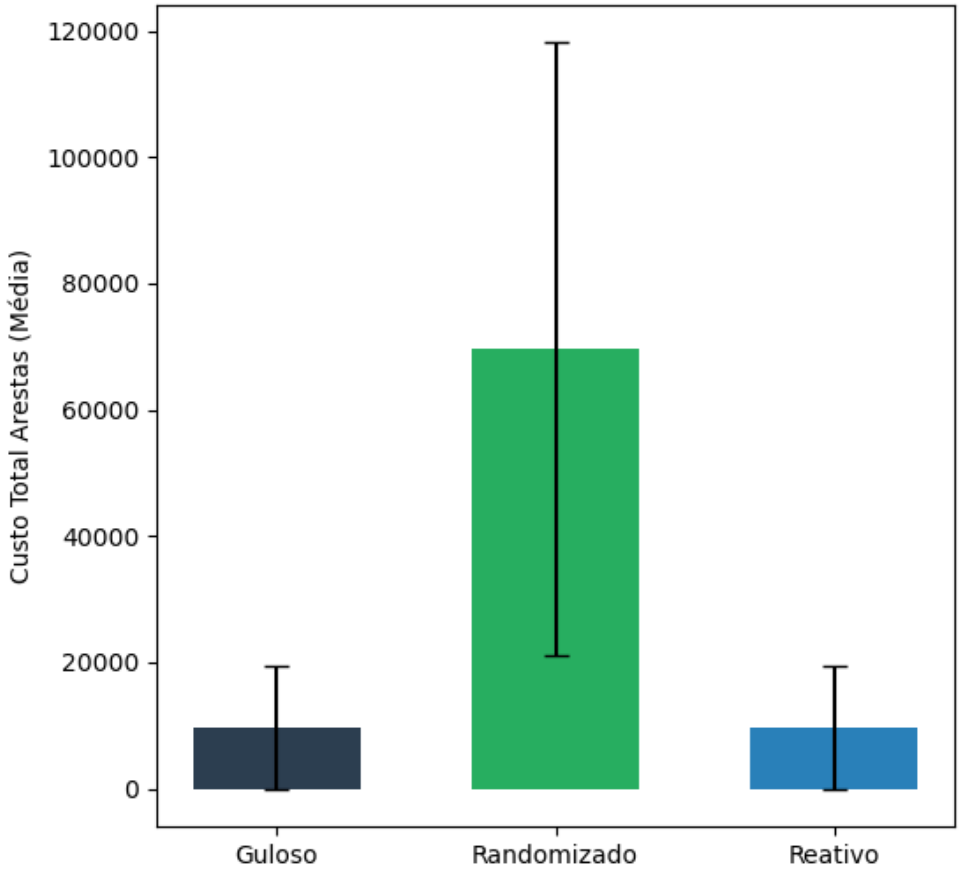
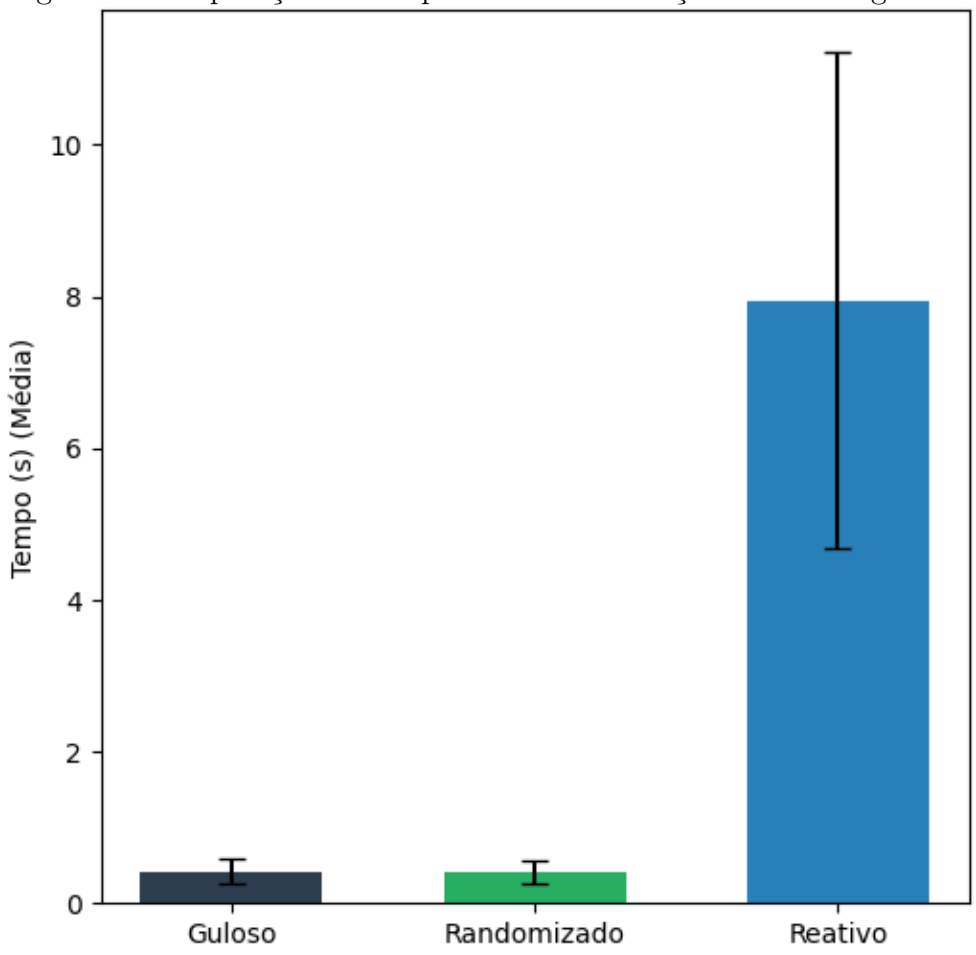


Figura 4: Comparação do tempo médio de execução entre os algoritmos.



5.5 Discussão

Observa-se que:

- Os algoritmos **Guloso** e **Reativo** obtiveram, para todas as instâncias, o mesmo custo total (por exemplo, 7044 na Instância 1 e 1274 na Instância 2), demonstrando que o método reativo, apesar de sua complexidade, alcança soluções equivalentes em termos de custo.
- O algoritmo **Randomizado** apresentou custos significativamente superiores, evidenciando a variabilidade introduzida pela aleatoriedade.
- Em termos de tempo de execução, os métodos guloso e randomizado tiveram desempenho muito semelhante (média em torno de 0,41 s), enquanto o reativo demonstrou

tempos elevados (média de aproximadamente 7,95 s), o que pode comprometer sua utilização em cenários que exijam respostas rápidas.

6 Conclusões

6.1 Desempenho dos Algoritmos

- **Custo Total:** Os algoritmos **Guloso** e **Reativo** apresentaram desempenho equivalente, enquanto o **Randomizado** obteve soluções de custo significativamente maiores.
- **Tempo de Execução:** Embora o método reativo garanta a mesma qualidade de solução que o guloso, seu tempo de execução é muito superior, o que o torna menos adequado para aplicações em tempo real.

6.2 Contribuições do Trabalho

Este estudo contribui para a compreensão das vantagens e limitações das abordagens heurísticas aplicadas à AGMG, destacando:

- A importância da escolha da estrutura de dados (lista de adjacência versus matriz de adjacência) para a eficiência computacional.
- A viabilidade de métodos híbridos (reativo) que combinam a eficiência do algoritmo guloso com a diversidade exploratória do randomizado.

6.3 Considerações Finais

Os resultados demonstram que, para o problema da Árvore Geradora Mínima Generalizada, a escolha do algoritmo deve considerar o equilíbrio entre a qualidade da solução e o tempo de execução. Embora o algoritmo reativo ofereça robustez ao combinar estratégias, seu alto custo computacional sugere que, em cenários onde a rapidez é essencial, o método guloso pode ser a alternativa mais apropriada.