

TEMA DO TRABALHO PRÁTICO 2

Objeto: Estender seu Tipo Abstrato de Dados - TAD ou classe desenvolvidos no Trabalho 1 de forma a assegurar que represente grafos conforme sejam definidas as instâncias do problema. Para este trabalho, o TAD ou a Classe devem apresentar os seguintes algoritmos para o problema:

- a) Algoritmo guloso
- b) Algoritmo guloso randomizado adaptativo;
- c) Algoritmo guloso randomizado adaptativo reativo;

PROBLEMA ALVO:

O **Minimum Gap Graph Partitioning Problem (MGGPP)** é um problema de particionamento de grafos ponderados por vértices, que possui as seguintes características principais:

Definição do Problema:

- **Grafo Não Direcionado:** O problema é definido em um grafo não direcionado $G=(V,E)$, onde V é o conjunto de vértices e E o conjunto de arestas.
- **Pesos nos Vértices:** Cada vértice $v \in V$ tem um peso associado w_v , que é um valor numérico. Este peso pode, por exemplo, representar algum atributo do vértice, como altura, demanda ou custo.
- **Partição em Subgrafos:** O objetivo do MGGPP é particionar o grafo em um número pré-definido p de subgrafos conectados, ou seja, cada subgrafo resultante deve ser conexo e não pode ser um conjunto vazio de vértices. Todos os subgrafos devem ter, no mínimo, dois vértices.
- **Minimização da Diferença de Pesos (Gap):** A característica principal do problema é minimizar a diferença, chamada de **gap**, entre o maior e o menor peso dentro de cada subgrafo. Para um subgrafo V_r , o gap γ_{V_r} é dado pela diferença entre o maior peso M_{V_r} e o menor peso m_{V_r} dentro do subgrafo. O objetivo é minimizar a soma dos gaps de todos os subgrafos, ou seja, minimizar $\sum_{r=1}^p \gamma_{V_r}$.

Regras e Restrições:

1. **Número pré-definido de subgrafos:** O número de subgrafos p é fixo e definido a priori, com a restrição de que $1 < p < n$, onde n é o número total de vértices do grafo.
2. **Subgrafos Conectados:** Todos os subgrafos devem ser conectados, ou seja, deve haver um caminho entre qualquer par de vértices dentro do mesmo subgrafo.
3. **Vértices Desconectados:** O problema considera que os subgrafos resultantes não podem ser desconexos ou compostos por apenas um vértice.
4. **Minimização do Gap Total:** O objetivo é minimizar a soma dos gaps em todos os subgrafos da partição, visando obter subgrafos cujas distribuições de pesos sejam o mais homogêneas possível.

Artigo de referência:

Artigo: <https://www.sciencedirect.com/science/article/abs/pii/S0305054821000939>

Instâncias disponíveis em: <https://homes.di.unimi.it/cordone/research/research.html>

- **SEMINÁRIO :: 10%**

- Envio dos slides até dia 23/setembro às 23:59 via classroom.
- Slides obrigatórios: Capa, organização da apresentação, descrição do problema, 1 slide para cada algoritmo implementado, experimentos computacionais (ambiente de testes, desenvolvimento, instâncias etc.), resultados e conclusões.
- Incluir link para git onde código deve estar disponível.
- Apresentação de 8 a 10 minutos.
- O seminário deve incluir: breve apresentação do problema; algoritmos propostos; experimentos computacionais; análise dos resultados; e conclusão.
- Todos os participantes do grupo devem participar do seminário.

- **RELATÓRIO :: 15%**

- Envio dos slides 26/setembro até 23:59 via classroom.
- Seguir modelo disponibilizado no classroom.
- Relatório de no máximo 10 páginas.

Observações:

- a) Incluir no git um arquivo txt com uma descrição mínima de como o programa deve ser executado em linha de comando (Linux), de forma a especificar qual o algoritmo será executado (guloso, guloso randomizado, ou guloso randomizado reativo), quais os parâmetros (número de iterações, semente de randomização, etc) e qual o arquivo de saída.
- b) O experimento a ser realizado nesta fase já foi detalhado em sala. Cabe a cada grupo avaliar o tempo que demanda o experimento de forma a assegurar-se de que o prazo para a entrega do relatório, do código e os arquivos de entrada e saída seja feita até a data estabelecida.
- c) Se código que não compila, então nota igual a zero.

IMPORTANTE

- O relatório deve informar as decisões de projeto e de implementação (fundamentadas nas dificuldades enfrentadas) referentes a apenas a parte 2 do trabalho de implementação, além de exemplos das saídas do programa para as funcionalidades solicitadas. O relatório não deve ultrapassar 10 páginas, não deve ter listagem de código fonte e todo pseudocódigo inserido deve ter as linhas numeradas;
- O trabalho é em grupo e deve ser feito sempre segundo as orientações. Os membros devem levar à professora as decisões de projeto para serem discutidas.
- Instâncias: Como são muitas instâncias, considerar as instâncias indicadas na planilha disponibilizada.

Perguntas Frequentes

1. Preciso fazer o trabalho 2 com o mesmo grupo do trabalho 1?

R - Não, mas as mudanças de composição devem ser informadas imediatamente após a postagem desta especificação no Classroom.

2. O projeto poderá ter mais de um arquivo fonte (c, cc, cpp e h)?

R - Pode (e, para boa organização do código, deve). A organização do projeto é livre.

3. O que será levado em conta na *avaliação individual* dos membros do grupo?

Ao longo do desenvolvimento do trabalho serão observados (entre outros) os seguintes aspectos.

- a. Envolvimento nas decisões de projeto.
- b. Participação na apresentação do seminário;
- c. Feedback dos colegas na autoavaliação do grupo.

Obs: Note que a nota do trabalho não será, necessariamente, a mesma para todos os membros do grupo.