

# Trabalho Prático Final

## GCC218 - Algoritmos em Grafos

## GCC262 - Grafos e suas Aplicações

### Universidade Federal de Lavras

Prof. Mayron César O. Moreira

Março de 2024

## 1 Introdução

Estudar problemas de logística é crucial para otimizar o fluxo de bens e serviços, resultando em maior eficiência e redução de custos para empresas e consumidores. A análise detalhada de processos logísticos permite identificar gargalos, melhorar o planejamento de rotas, gerenciar estoques de forma mais eficaz e implementar tecnologias que aprimoram a tomada de decisões.

A logística desempenha um papel fundamental na competitividade das empresas, influenciando diretamente a satisfação do cliente e a sustentabilidade ambiental. Ao compreender os desafios logísticos, é possível desenvolver soluções inovadoras que impulsionam o crescimento econômico e promovem um futuro mais eficiente e responsável.

## 2 Definição formal

O problema base pode ser definido em um grafo conexo  $G = (V, E)$ , onde  $V$  é o conjunto de nós e  $E$  o conjunto de arestas. Os nós representam interseções (ou esquinas) em uma região (urbana ou rural), enquanto as arestas são as vias de acesso (ruas, avenidas, etc). Um subconjunto  $E_R \subseteq E$  dessas arestas deve ser atendido. Seja  $n = |E_R|$  o número de serviços. Uma aresta  $(i, j) \in E$  pode ser percorrida qualquer número de vezes com um custo de  $c_{ij}$  cada vez, e uma demanda de  $q_{ij}$  está associada a qualquer aresta  $(i, j) \in E_R$ . **O problema visa encontrar um conjunto de viagens de veículos com custo mínimo**, tal que cada viagem comece e termine em um nó depósito  $v_0 \in V$ , cada aresta requerida seja atendida por uma única viagem, e a demanda total para qualquer veículo não exceda uma capacidade  $Q$ .

A variação estudada no trabalho prático redefine  $G$ , em particular, como um multigrafo conectado  $G = (V, E, A)$ , onde  $V$  é o conjunto de nós,  $E$  o conjunto

de arestas e  $A$  o conjunto de arcos (vias de mão única). Serviços são requeridos para um subconjunto de nós  $V_R \subseteq V$ , arestas  $E_R \subseteq E$  e arcos  $A_R \subseteq A$ , tal que  $n = |V_R| + |E_R| + |A_R|$ .

### 3 Etapas do trabalho

As etapas do trabalho prático são descritas a seguir.

#### 3.1 Etapa 1: Pré-processamento dos dados

Os objetivos da Etapa 1 consistem em:

- representar a modelagem do problema por meio de estruturas de dados em grafos;
- implementação da leitura dos dados;
- cálculo de estatísticas a respeito dos grafos.

Para o cálculo das estatísticas, o grupo deve implementar funções que retornem as seguintes informações.

1. Quantidade de vértices;
2. Quantidade de arestas;
3. Quantidade de arcos;
4. Quantidade de vértices requeridos;
5. Quantidade de arestas requeridas;
6. Quantidade de arcos requeridos;
7. Densidade do grafo (*order strength*);
8. Componentes conectados;
9. Grau mínimo dos vértices;
10. Grau máximo dos vértices;
11. Intermediação - A intermediação de um nó mede a frequência com que ele aparece nos caminhos mais curtos entre outros nós. *Não é necessário calcular outros caminhos mais curtos alternativos*;
12. Caminho médio;
13. Diâmetro.

**Importante:** muitas dessas métricas utilizam os resultados da matriz de caminhos mais curtos de múltiplas fontes. Assim, como um dos produtos da Etapa 1, é necessário desenvolver o algoritmo que gera tal matriz, assim como a matriz de predecessores.

### 3.2 Etapa 2: Solução Inicial

Na Fase 2, pretende-se desenvolver um algoritmo construtivo para o problema estudado na Fase 1. Por construtivo, entende-se como um algoritmo que inicia com uma solução vazia e ao final de suas iterações, constrói uma solução que atende a todas as restrições do problema. Logo, deseja-se que vocês implementem um algoritmo que retorne uma solução que:

- não ultrapasse a capacidade dos veículos em cada rota;
- cada serviço seja executado por exatamente 1 rota;
- caso uma rota passe mais de uma vez por um vértice, ou uma aresta, ou um arco requeridos, o valor de demanda do serviço e seu custo de serviço devem ser contados apenas 1 vez.

Os valores de referência das soluções, assim como o formato padrão para a escrita das soluções estão no Campus Virtual. Saliento que:

- o código deve ser autoral, e no fim do semestre, qualquer membro da dupla deve ser capaz de explicá-lo na entrevista final da disciplina.
- todas as instâncias disponibilizadas devem ser testadas.
- os grupos não precisam disponibilizar o link dos repositórios github, visto que isso já fora feito na Fase 1. **No entanto**, os grupos **devem disponibilizar, no Campus Virtual**, uma pasta com todas as soluções geradas pelo grupo. A pasta deve ter o nome do grupo, com a extensão .zip. Exemplo: “G1.zip”. Cada solução deve seguir o padrão de nomenclatura “sol-nome\_instancia.dat”. Exemplo: “sol-BHW1.dat”.

### 3.3 Etapa 3: Métodos de melhoria

Na Fase 3, pretende-se aprimorar o algoritmo construtivo da fase 2 através de um novo algoritmo construtivo ou um algoritmo de busca local. As mesmas restrições e instâncias testes serão mantidas nessa etapa. Os grupos poderão desenvolver sua própria solução ou utilizar algum código pronto obtido de publicações da literatura do problema ou por meio de um sistema de Inteligência Artificial. Os valores de referência das soluções, assim como o formato padrão para a escrita das soluções estão no Campus Virtual. Saliento que:

- independente da forma a qual o grupo obtenha o código, qualquer membro da dupla deve ser capaz de explicá-lo na entrevista final da disciplina.
- todas as instâncias disponibilizadas devem ser testadas.
- os grupos não precisam disponibilizar o link dos repositórios github, visto que isso já fora feito na Fase 1. **No entanto**, os grupos **devem disponibilizar, no Campus Virtual**, uma pasta com todas as soluções geradas pelo grupo. A pasta deve ter o nome do grupo, com a extensão .zip. Exemplo: “G1.zip”. Cada solução deve seguir o padrão de nomenclatura “sol-nome\_instancia.dat”. Exemplo: “sol-BHW1.dat”.

### 3.4 Etapa 4: Relatório e entrevista

Cada grupo deverá confeccionar um relatório de até 5 páginas (sem contar a capa), que contenha os seguintes itens:

- Introdução: motivação prático do estudo e sua descrição matemática formal;
- Objetivos: qual é o objetivo do trabalho prático, e os objetivos secundários. Note que existem objetivos em comum de todos os grupos. No entanto, dependendo da escolha dos algoritmos, os objetivos da Etapa 3 podem modificar.
- Metodologia: apresente o pseudocódigo dos algoritmos implementados nas etapas 2 e 3.
- Resultados: apresente gráficos que resumam os resultados e comparem os algoritmos das Etapas 2-3. Seja claro e conciso.
- Conclusões e outras perspectivas: indique as conclusões que o grupo extraiu do trabalho, e o que poderia ser aprimorado quanto à resolução do problema e até mesmo a concepção do problema.

## 4 Regras

1. O trabalho pode ser feito em dupla ou individual. Para que o tempo seja melhor aproveitado, incentivo fortemente que os alunos façam o trabalho em duplas.
2. **O código deverá ser autoral.**
3. A linguagem de programação permitida será Python ou C++.
4. O grupo pode utilizar estruturas de dados da *standard library*. No entanto, as funções relacionadas diretamente a grafos, advindas de *frameworks* como *networkx* ou *igraph* não serão permitidas no momento.
5. Qualquer tentativa de plágio será devidamente punida pelos docentes com medidas administrativas cabíveis.
6. Nas Fases 2-3, qualquer grupo que não siga as regras padronizadas poderá ter sua nota das Fases 2-3 decrescida em até 30% (3 pontos).

## 5 Entregáveis

A dupla deverá criar um repositório no GitHub, e nesse repositório, apresentar código fonte e README bem documentado. O link do repositório será inserido no Campus Virtual, **por apenas um membro da dupla**. Para as Fases 2-3, o grupo deverá apresentar as soluções encontradas, seguindo estritamente o

padrão estabelecido pelo docente. Na Fase 4, um relatório no formato PDF deverá ser confeccionado.

## **6 Instâncias teste**

As instâncias estão disponíveis no Campus Virtual.