



# Trabalho de Pesquisa Operacional

Parte 02 - Trabalho de implementação computacional

Problema da rota do caminhão coletor de lixo

**Leonardo Azalim de Oliveira**  
**Raíh César Silva de Lima**

Sob orientação do Professor Marcos Passini

Departamento de Ciência da Computação  
Universidade Federal de Juiz de Fora  
Minas Gerais - Brasil  
ERE 2021.1

# 1 Introdução

O presente trabalho foi apresentado como parte das avaliações da disciplina de Pesquisa Operacional do Departamento de Computação da Universidade Federal de Juiz de Fora. O mesmo tem por objetivo prover uma solução otimizada para o problema da rota do caminhão coletor de lixo. Este problema é da classe de problemas de caminho de custo mínimo[7], podendo ser considerado uma variação do conhecido Problema do Carteiro Chinês (PCC), referenciado na literatura em inglês como *Chinese Postman Problem* (CPP) (pertencente à classe de *Arc Routing Problems*[1]) ou *Route Inspection Problem*[8].

Podemos considerar o problema de interesse como um problema derivado do PCC direcionado. Já que por conta da natureza das questões relacionadas à logística de rotas de veículos, as leis de trânsito vigentes e consequentemente o sentido de trânsito das vias devem ser respeitados.

Os dados utilizados foram gentilmente cedidos pelo Departamento de Limpeza Urbana (DEMLURB)[5], órgão pertencente à administração direta da Prefeitura de Juiz de Fora (PJF)[4].

## 2 Dados

$G$  = Grafo que representa todas as ruas e esquinas da área analisada

$i, j$  = Nós pertencentes à  $G$  que representam os cruzamentos entre as ruas da área analisada

$(i, j)$  = Tupla que representa os arcos de  $G$

$c_{i,j}$  = Custo de deslocamento pelo arco  $(i, j)$

## 3 Variáveis

$x_{i,j}$  = Número de vezes que o caminhão coletor passa pelo arco  $(i, j)$

## 4 Função Objetivo

z: Minimizar a distância percorrida pelo caminhão coletor

$$z(c, x) = \sum_{i,j=1}^{i,j} (c_{i,j} \times x_{i,j})$$

## 5 Restrições

1. Todas os arcos (ruas) devem ser cobertos:

$$x_{i,j} \geq 1, \forall (i, j) \in G$$

2. O caminhão não deve parar em um nó (cruzamento), ou seja, o número de entradas em um nó deve ser igual ao número de saídas do mesmo:

$$\sum_{j:(i,j)=1}^{i,j} x_{i,j} - \sum_{j:(j,i)=1}^{i,j} x_{j,i} = 0 \quad \begin{cases} \forall i \in N, \\ \forall (i, j) \in G \end{cases}$$

## 6 Modelo

Minimizar  $z = \sum_{(i,j)=1}^{i,j} c_{i,j} x_{i,j}$

Sujeito a:

$$x_{i,j} \geq 1$$

$$\sum_{j:(i,j)=1}^{i,j} x_{i,j} - \sum_{j:(j,i)=1}^{i,j} x_{j,i} = 0$$

## 7 Solução / Implementação

Após a etapa de modelagem matemática foi implementada uma solução computacional para o problema proposto. Foi utilizada uma estrutura de dados de grafo direcionado (apoiada por uma tabela de emparelhamento de nós). Cada aresta representa uma rua da área de interesse. A direção das arestas, representa o sentido de trânsito das ruas. E os nós representam os cruzamentos entre elas.

O programa e seu código fonte foram publicados na plataforma *Github*[2] através de um repositório público[6]. Escrito na linguagem C++, o programa realiza a leitura dos dados provenientes de um arquivo no formato CSV (*comma-separated values*) com o intuito de prepará-los para serem processados pelo *solver* do programa *IBM® ILOG CPLEX®*[3].

O *solver* então retorna uma das possíveis soluções e o programa, além de a exibir na saída padrão do sistema, também salva uma cópia no disco local.

### 7.1 Limitações

O programa possui algumas limitações por conta da forma como foi modelado. Dentre elas, podemos destacar as seguintes:

- A rota (caminho) é somente aberta, ou seja, o ponto de partida não é igual ao ponto final;
- Não foi feito o tratamento das conversões não permitidas, devido ao considerável aumento da complexidade e ao limite de escopo de trabalho proposto;
- O peso do caminhão e de sua eventual carga não interferem na escolha dos pontos da rota; e
- Todas as ruas foram consideradas como somente sendo de sentido único de trânsito, pois a rota 38 se encontra na região central da cidade onde a grande maioria das ruas é de sentido único;

## 8 Agradecimentos

Gostaríamos de agradecer aos profissionais Everton, Joana, Telma e Victória que integram o Setor de Engenharia do Centro Administrativo e ao Senhor Anderson do Setor de Coleta, ambos setores pertencentes ao DEMLURB[5] pelos dados cartográficos e informações que serviram de base para este trabalho. Também gostaríamos de agradecer ao Google pela disponibilização das ferramentas Google Earth e Google Maps que proporcionaram as aferições feitas à partir dos dados cartográficos. Por último, mas não menos importante, gostaríamos de agradecer à IBM que propiciou a licença de uso do componente computacional CPLEX®[3] o qual foi de suma importância para obtenção da solução do problema tratado.

## Referências bibliográficas

- [1] Ravindra Ahuja, Thomas Magnati, and James Orlin. *Network Flows: Theory, Algorithms and Applications*. New Jersey, USA: Prentice Hall, 1993.
- [2] GitHub. *Where the world builds software*. Acessado em 07 September 2021. 2021. URL: <https://github.com/about>.
- [3] IBM. *IBM CPLEX Optimizer*. Acessado em 07 September 2021. 2021. URL: <https://www.ibm.com/br-pt/analytics/cplex-optimizer>.
- [4] Prefeitura de Juiz de Fora. *Website Oficial*. Acessado em 02 September 2021. 2021. URL: <https://www.pjf.mg.gov.br/>.
- [5] Departamento de Limpeza Urbana - PJF. *Apresentação*. Acessado em 02 September 2021. 2021. URL: <https://demlurb.pjf.mg.gov.br/apresentacao.php>.
- [6] Leonardo Oliveira and Raih Lima. *PO-DEMLURB*. 2021. URL: <https://github.com/oliveiraleo/PO-DEMLURB>.
- [7] Wikipédia. *Problema do caminho mínimo*. Acessado em 08 September 2021. 2021. URL: [https://pt.wikipedia.org/wiki/Problema\\_do\\_caminho\\_m%C3%ADnimo](https://pt.wikipedia.org/wiki/Problema_do_caminho_m%C3%ADnimo).
- [8] Wikipédia. *Route inspection problem*. Acessado em 02 September 2021. 2021. URL: [https://en.wikipedia.org/wiki/Route\\_inspection\\_problem](https://en.wikipedia.org/wiki/Route_inspection_problem).