

Otimização da Rota de Coleta de Lixo Através de um Algoritmo Heurístico do Problema do Caixeiro Viajante

Diego Filgueiras Balderrama

Instituto Metrópole Digital - IMD / Universidade Federal do Rio Grande do Norte - UFRN

Resumo

A coleta de lixo é um dos mais importantes serviços para a saúde das pessoas que vivem em determinada região. Dado isso, para minimizar o desperdício de recursos como combustível, o trajeto realizado pelo caminhão de lixo por um bairro deve ser o mais otimizado possível e o objetivo deste trabalho é utilizar um algoritmo heurístico para o Problema do Caixeiro Viajante para encontrar tal trajeto, pois assim como vários outros problemas do cotidiano, este também pode ser resolvido através de conceitos da Teoria de Grafos.

Palavras-chave: Teoria de Grafos, Caixeiro Viajante, Coleta de lixo

1. Introdução

Este documento tem como objetivo descrever o trabalho a ser realizado para a disciplina de Grafos do curso de Bacharelado em Tecnologia da Informação. A descrição está dividida na descrição do problema real, que será o da melhor rota para a coleta de lixo de uma comunidade, bem como definir a modelagem em grafos desse problema e apresentar uma breve conclusão a respeito.

2. Descrição do problema real

A coleta de lixo é um serviço essencial para qualquer comunidade e um dos mais importantes serviços para execução do saneamento básico eficiente de uma

*Autor correspondente

Email address: diego.filgueiras.084@ufrn.edu.br (Diego Filgueiras Balderrama)

determinada região, dado que atua na prevenção da proliferação de doenças para a população local.

Considerando esse fator, a rota realizada por um caminhão de coleta de lixo por uma comunidade visando o menor desperdício de recursos deve ser definida idealmente atendendo aos seguintes critérios: a passagem única por cada ponto de coleta da região; percorrimento da menor distância possível. A partir disso, consegue-se reduzir o combustível e o tempo gasto pelo caminhão o máximo possível. Ao final, o caminhão deve retornar ao ponto inicial do percurso.

3. Modelagem em grafos

Para esse problema, a modelagem é definida por um grafo $G = (V, M)$ onde $V = \{1, \dots, n\}$ é o conjunto de vértices e $M = \{1, \dots, m\}$ é o conjunto de arestas de G com custos C_{ij} associados a cada aresta conectando os vértices i e j . O problema consiste em localizar o menor Ciclo Hamiltoniano do grafo G e seu tamanho é calculado pelo somatório dos pesos das arestas que compõem o ciclo.

Nos conceitos do problema real, os pontos de coleta de lixo são representados pelos vértices do grafo e as ruas são representadas pelas arestas. Além disso, as arestas têm um peso atribuído a elas que representa o seu comprimento. O objetivo é visitar todos os pontos passando uma única vez por cada um deles e, ao final, retornar ao ponto de origem. Este percurso deve ser feito de forma a minimizar a distancia total percorrida.

O problema mais adequado para moldar a solução do problema apresentado é o do Caixeiro Viajante, pois o seu objetivo consiste basicamente em encontrar o menor Ciclo Hamiltoniano dentro do grafo, assim como foi definido o objetivo do presente projeto.

Para a execução deste trabalho, os grafos da base de dados admitidos para análise deverão seguir a restrição de serem completos, ou seja, quaisquer dois vértices distintos no grafo devem ser adjacentes.

4. Estado da Arte

4.1. Revisão Bibliográfica

Devido a sua relevância, o problema da melhor rota para a coleta de lixo para uma determinada região já foi abordado em alguns artigos e alguns deles serão citados a seguir.

O estudo de Moura, Fontes e Ribeiro (2001) [1] se propôs a definir a melhor rota para a coleta de lixo seletiva no campus da Universidade Federal de Viçosa (UFV) através de um Sistema de Informações Geográficas denominado *ArcView*. Ao final do estudo, foi encontrado um percurso que passava por todos os pontos de coleta da UFV e reduzia em aproximadamente 937 metros a distância total da rota que estava sendo realizada na época. Na figura 1, é possível visualizar uma captura de tela da solução que o *software* utilizado pelos autores calculou.

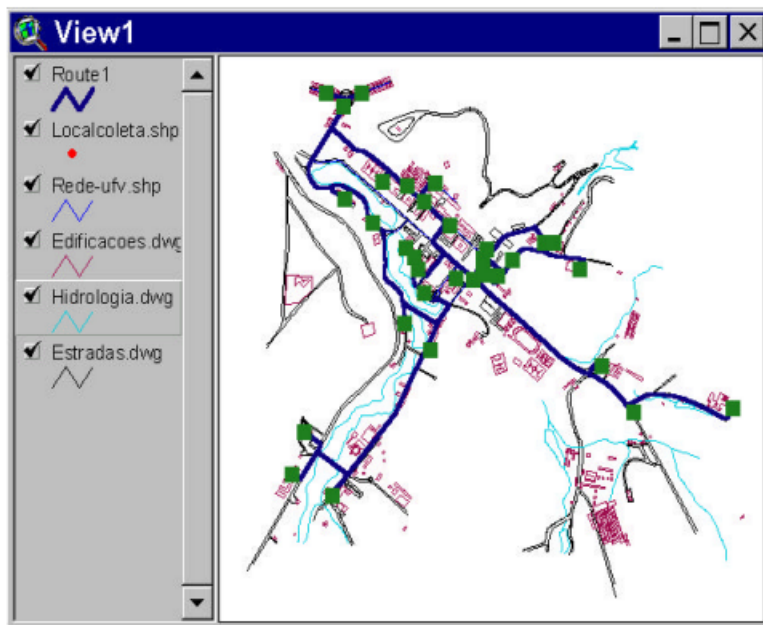


Figura 1: Melhor rota definida pelo *ArcView*. Fonte: Moura, Fontes e Ribeiro (2001).

O trabalho de Silva, Lins e Xavier (2020) [2] utilizou o algoritmo do Carteiro Chinês Direcionado para obter a rota ótima para os veículos coletores de lixo

para os bairros Engenho do Meio e Cordeiro, ambos localizados na cidade de Recife - PE. Os autores separaram a solução em três etapas: escrever o Modelo de Programação Linear que resolve o Problema do Carteiro Chinês Direcionado, resolver o modelo a partir do *software Soplex* e, por último, utilizar um algoritmo adaptado de Fleury para grafos direcionados a fim de sequenciar o percurso do caminhão. Como resultado, os autores conseguiram reduzir a distância percorrida em 12,7% no bairro Engenho do Meio e em 8,86% no bairro Cordeiro. Não há imagens do grafo resultante, porém a figura 2 exibe o resultado do *Soplex* para o Modelo de Programação Linear, que informa a distância total de cada percurso para os bairros estabelecidos, indicando que o resultado final conseguiu alcançar a solução ótima para os dois casos.

```

LP has 115 rows 200 columns and 400 nonzeros.
Equilibrium scaling LP (persistent)
Simplifier removed 0 rows, 0 columns, 0 nonzeros, 0 col bounds, 0 row bounds
Reduced LP has 115 rows 200 columns 400 nonzeros
Equilibrium scaling LP
type | time | iters | facts | shift | violation | obj value
L | 0.0 | 0 | 1 | 0.00e+00 | 5.20e+01 | 1.73830000e+04
L | 0.0 | 88 | 2 | 0.00e+00 | 0.00e+00 | 2.33170000e+04
--- unscaling internal solution
--- unsimplifying solution and basis
--- unscaling external solution
--- verifying computed solution
L | 0.0 | 0 | 1 | 0.00e+00 | 0.00e+00 | 2.33170000e+04
L | 0.0 | 0 | 1 | 0.00e+00 | 0.00e+00 | 2.33170000e+04
--- unscaling external solution
--- verifying computed solution

SoPlex status : problem is solved [optimal]
Solving time (sec) : 0.05
Iterations : 88
Objective value : 2.33170000e+04

```

```

LP has 332 rows 604 columns and 1208 nonzeros.
Equilibrium scaling LP (persistent)
Simplifier removed 0 rows, 0 columns, 0 nonzeros, 0 col bounds, 0 row bounds
Reduced LP has 332 rows 604 columns 1208 nonzeros
Equilibrium scaling LP
type | time | iters | facts | shift | violation | obj value
L | 0.0 | 0 | 1 | 0.00e+00 | 2.20e+02 | 5.25380000e+04
L | 0.0 | 199 | 1 | 0.00e+00 | 5.90e+01 | 7.61890000e+04
L | 0.0 | 354 | 3 | 0.00e+00 | 0.00e+00 | 8.96430000e+04
--- unscaling internal solution
--- unsimplifying solution and basis
--- unscaling external solution
--- verifying computed solution
L | 0.0 | 0 | 1 | 0.00e+00 | 0.00e+00 | 8.96430000e+04
L | 0.0 | 0 | 1 | 0.00e+00 | 0.00e+00 | 8.96430000e+04
--- unscaling external solution
--- verifying computed solution

SoPlex status : problem is solved [optimal]
Solving time (sec) : 0.07
Iterations : 354
Objective value : 8.96430000e+04

```

Figura 2: Captura de tela do *Soplex*. Fonte: Silva, Lins e Xavier (2020).

Voltando o foco para os algoritmos que se propõem a resolver o Problema do Caixeiro Viajante, temos, inicialmente, o seu algoritmo exato. Ele se baseia na comparação do custo do percurso de todos os Ciclos Hamiltonianos do grafo, o que já se torna inviável mesmo para grafos com um baixo número de vértices.

Visto que o algoritmo exato não é praticável para a grande maioria dos casos do Problema do Caixeiro Viajante, vários algoritmos heurísticos foram desenvolvidos e alguns deles serão mencionados a seguir.

O algoritmo Vizinho Mais Próximo, proposto por Bellmore e Nemhauser (1968) [3], é uma heurística para o Problema do Caixeiro Viajante que se baseia na escolha gulosa de percorrer os vértices de acordo com a aresta de menor peso a cada iteração e possui complexidade $O(n^2)$. Moro, Schroeder e Jesus (2013) [4] aplicaram o esse algoritmo para resolver o problema de otimização da rota de coleta de lixo na rede rural do município de Missal - PR. Os resultados foram significativamente promissores, dado que o percurso calculado possibilitou a redução da distância percorrida pela rota vigente em 19,7%.

Outro algoritmo heurístico para solucionar o Problema do Caixeiro Viajante é o de Cristofides (1976) [5], que possui uma linha de raciocínio mais complexa e robusta, criando primeiramente uma árvore geradora mínima T do grafo original G , depois fazendo o emparelhamento perfeito de peso mínimo M no subgrafo induzido por todos os vértices de grau ímpar de G , em seguida combinando as arestas de T e M em um novo grafo H , após isso gerando um Ciclo Euleriano em H e, por fim, a partir desse ciclo, formando um Circuito Hamiltoniano ignorando os vértices repetidos. A complexidade desse algoritmo é $O(n^3)$.

Ainda no contexto heurístico, temos o algoritmo de melhoria 2-opt, proposto por Croes (1958) [6], que funciona eliminando duas arestas não adjacentes do grafo original, reconectando-as por outras duas arestas de forma que ainda haja um único ciclo e verificando se houve redução no custo do percurso. Esse processamento é executado para todos os pares de arestas. Ao final, o ciclo escolhido será o de menor custo dentre todos os encontrados durante as iterações. Sua complexidade é $O(n^2)$.

4.2. Casos de Teste

Os casos de testes utilizados na solução do problema apresentado neste trabalho serão obtidos a partir da biblioteca TSPLIB, na qual é possível filtrar amostras de grafos que satisfaçam a condição de serem completos.

4.3. Links Relevantes

Esta seção apresenta alguns *links* de trabalhos relacionados ao problema real e à modelagem em grafos.

- Biblioteca de amostras de grafos que podem ser utilizados para resolução de Problemas do Caixeiro Viajante: ***TSPLIB***
- Dissertação de Mestrado realizada na UFRJ em que se desenvolveu o algoritmo heurístico MARCOLIX para o Problema do Caixeiro Viajante: **O Problema do Planejamento e Percorso de Veículos na Coleta de Lixo Urbano Domciliar**
- Trabalho de Conclusão de Curso para avaliar e roteirizar a coleta de lixo de uma cidade de pequeno porte, utilizando um Sistema de Informação Geográfica: **Roteirização através do SIG para Coleta de Lixo Doméstico: Um Estudo de Caso da Cidade de Silvânia - GO**

5. Descrição da Abordagem Algorítmica

5.1. Pseudocódigo

Tendo em vista a inviabilidade da execução do algoritmo exato do Problema do Caixeiro Viajante para entradas grandes, o presente trabalho propõe-se a resolver o problema da otimização da rota na coleta de lixo de um bairro utilizando a heurística do Vizinho Mais Próximo seguido do algoritmo de melhoria 2-opt. Abaixo, encontra-se o pseudocódigo do trecho mais crucial do projeto, escrito com base nos pseudocódigos disponíveis em [7] e [8]:

1. Escolher um vértice arbitrário como vértice atual.

2. Descobrir a aresta de menor peso que seja conectada ao vértice atual e a um vértice não visitado V .
3. Tornar V o vértice atual.
4. Marcar V como visitado.
5. Se todos os vértices do grafo estiverem visitados, ir para o passo 7.
6. Senão, ir para o passo 2.
7. Armazenar a ordem de visitação dos vértices.
8. Remover duas arestas distintas não consecutivas do grafo e armazenar o grafo resultante.
9. Construir todas as soluções viáveis contendo o grafo resultante.
10. Armazenar a melhor das soluções dentre as encontradas.
11. Se ainda houver duas arestas não selecionadas, ir para o passo 8.
12. Senão, encerrar o algoritmo.

O código disponível neste *link* foi consultado para melhor compreensão da implementação do algoritmo do Vizinho Mais Próximo durante a elaboração do projeto.

5.2. Análise de Complexidade

A começar pela leitura de dados, a complexidade dessa operação é $O(n)$, visto que é feito sobre os dados de cada vértice do grafo.

Após a leitura, é criada a matriz de distância, e por mais que seja apenas a diagonal inferior, sem redundância, sua complexidade ainda é $O(n^2)$.

Em seguida, para encontrar uma solução para o grafo dado, é executado o algoritmo do Vizinho Mais Próximo, que como já dito anteriormente, tem complexidade $O(n^2)$. Depois, é gerada a lista de arestas do percurso calculado, com complexidade $O(n)$.

Logo após, é aplicado um algoritmo de melhoria sobre o resultado obtido pelo Vizinho Mais Próximo, o 2-opt. Sua complexidade é $O(n^2)$, como já foi dito. Também é gerada uma lista de arestas desse novo percurso, com complexidade $O(n)$.

Ao final, somando as complexidades, $O(n) + O(n^2) + O(n^2) + O(n) + O(n^2) + O(n)$, obtemos a complexidade total de $O(n^2)$.

5.3. Experimentos Computacionais

5.3.1. Base de Dados

Os experimentos computacionais foram realizados sobre arquivos disponibilizados pela *TSPLib*, escolhendo alguns arquivos disponíveis neste *link*. Foram selecionados apenas arquivos que continham o valor `EUC_2D` definido na propriedade `EDGE_WEIGHT_FORMAT`. Dessa forma, após ler as coordenadas dos vértices, era possível calcular a distância entre dois pontos $A(x_a, y_a)$ e $B(x_b, y_b)$ através da distância euclidiana $d = \sqrt{(x_b - x_a)^2 + (y_b - y_a)^2}$.

5.3.2. Descrição Técnica da Plataforma de Execução

Os experimentos foram realizados no *Google Colaboratory*, uma plataforma *online*, em que é possível executar códigos em Python em um servidor da *Google*, ou seja, não são utilizados recursos da máquina local.

As especificações técnicas do *hardware* e *software* da máquina que executou os experimentos são:

- Sistema Operacional: Linux 0065c2b217fe 5.4.188+ #1 SMP Sun Apr 24 10:03:06 PDT 2022 x86_64 x86_64 x86_64 GNU/Linux
- Processador: Intel(R) Xeon(R) CPU @ 2.20GHz
- Memória RAM: 13.29 GB
- Armazenamento: 107.72 GB

5.4. Resultados e Discussões

Foram executados, ao todo, doze casos de testes (grafos), com o número de vértices variando entre 16 e 18512 a depender do grafo. Porém, nos atentaremos aos três maiores casos utilizados.

O terceiro maior, com 2103 vértices, levou aproximadamente 7.45 segundos para calcular o resultado (tempo apenas da execução do VMP e 2-opt).

A distância do percurso obtido pelo programa foi de aproximadamente 87055 metros e, segundo a TSPLib, o resultado ótimo para esse grafo está entre 79952 e 80450.

O segundo maior caso tinha 2319 vértices e demorou aproximadamente 8.59 segundos para encontrar a solução (mais uma vez, considerando apenas o VMP e 2-opt). o resultado obtido pelo programa acusou uma distância cerca de 277258 metros, enquanto a TSLib indica a solução ótima com apenas 234256 metros.

Por fim, o maior caso de teste com 18512 vértices, demorou um total aproximado de 938.81 segundos, o equivalente a cerca de 15 minutos. O resultado calculado trouxe um percurso de 795767 metros e a solução ótima, de acordo com a TSPLib está entre 644650 e 645488.

Como era esperado, o resultado ótimo não foi alcançado nos casos de teste acima pois a abordagem utilizada trata-se de uma heurística. Todavia, o tempo necessário para se chegar um resultado é bem menor do que usando o algoritmo exato, dado que o problema de otimização da rota do Caixeiro Viajante é um problema NP-difícil, portanto a diferença entre os resultados obtidos e os esperados é baixa o suficiente para que o resultado seja aceito como resposta no âmbito deste trabalho.

6. Conclusão

Por fim, é possível concluir que existem problemas do cotidiano mais relacionados a Teoria de Grafos do que se pode imaginar e, quando aplicados corretamente, os conceitos e propriedades podem trazer um impacto positivo na questão de economia de recursos, como é o caso deste trabalho.

Também foi possível avaliar a viabilidade e custo-benefício de utilizar a solução heurística do VMP junto ao 2-opt em vez do algoritmo exato para o problema do Caixeiro Viajante

Referências

- [1] M. C. MOURA, A. A. FONTES, C. A. A. S. RIBEIRO, Determinação da melhor rota para coleta seletiva de lixo no campus da universidade federal de viçosa utilizando dos sistemas de informações geográficas, Anais X (2001) 1119–1125.
- [2] A. A. Silva, S. L. S. Lins, A. S. Xavier, O problema do carteiro chinês aplicado na otimização de rotas usadas na coleta de lixo reciclável: um estudo de caso, LI Simpósio Brasileiro de Pesquisa Operacional.
- [3] M. Bellmore, G. L. Nemhauser, The traveling salesman problem: A survey, Operations Research 16 (3) (1968) 538–558.
URL <http://www.jstor.org/stable/168581>
- [4] M. F. MORO, W. Schroeder, G. C. JESUS, Otimização da rota Ótima de coleta seletiva de resíduos na Área rural do município de missal - paraná, utilizando heurísticas de solução do problema do caixeiro viajante, XXXIII Encontro Nacional de Engenharia de Produção.
- [5] N. Christofides, Worst-case analysis of a new heuristic for the travelling salesman problem, Technical Report 388, Graduate School of Industrial Administration, Carnegie Mellon University (1976).
- [6] G. A. Croes, A method for solving traveling-salesman problems, Operations Research 6 (6) (1958) 791–812.
URL <https://EconPapers.repec.org/RePEc:inm:oropre:v:6:y:1958:i:6:p:791-812>
- [7] Wikipedia, Algoritmo do vizinho mais próximo.
URL https://pt.wikipedia.org/wiki/Algoritmo_do_vizinho_mais_pr%C3%B3ximo
- [8] V. E. Wilhelm, Problema do caixeiro viajante.
URL https://docs.ufpr.br/~volmir/P0_II_12_TSP.pdf