

Uma aplicação do problema do carteiro chinês direcionado na coleta de lixo urbano

An application of the chinese wallet problem targeted in urban waste collection

DOI:10.34117/bjdv6n5-062

Recebimento dos originais: 20/04/2020

Aceitação para publicação: 05/05/2020

Andersson Alves da Silva

Doutorando em Ciência da Computação pela Universidade Federal de Pernambuco – UFPE

Instituição: Universidade Federal de Pernambuco – UFPE

Endereço: Av. Jornalista Anibal Fernandes, s/n, CEP: 50740-560. Bairro: Cidade Universitária, Recife–PE, Brasil.

E-mail: andersson.alves.silva@outlook.com

Sóstenes Luiz Soares Lins

Ph.D. em Combinatorics and Optimization pela University of Waterloo, Canadá

Instituição: Universidade Federal de Pernambuco – UFPE

Endereço: Av. Jornalista Anibal Fernandes, s/n, CEP: 50740-560. Bairro: Cidade Universitária, Recife–PE, Brasil.

E-mail: sostenes.lins@gmail.com

Amanda da Silva Xavier

Doutoranda em Ciência da Computação pela Universidade Federal de Pernambuco – UFPE

Instituição: Universidade Federal de Pernambuco – UFPE

Endereço: Av. Jornalista Anibal Fernandes, s/n, CEP: 50740-560. Bairro: Cidade Universitária, Recife–PE, Brasil.

E-mail: amanda.sdx28@gmail.com

RESUMO

Otimizar rotas para caminhões de lixo urbano contribuem para a redução de custos dos cofres públicos. Este trabalho objetiva apresentar um programa simples para resolução do Problema do Carteiro Chinês Direcionado atuando na determinação de rotas mínimas para veículos coletores de lixo e aplicado em dois bairros na cidade de Recife-PE. O programa recebe a matriz de distâncias do grafo e determina a rota ótima em 3 etapas: escrever o modelo matemático que resolve o PCCD, resolvê-lo por um solver e traçar a rota em um grafo Euleriano. O resultado do programa foi comparado com as rotas atuais: redução da distância total de 12,7% (Engenho do Meio) e 8,86% (Cordeiro). O *tour* para o caminhão de lixo obteve redução na distância total, embora algumas ruas precisassem ser percorridas mais de uma vez para alcançar certos segmentos de ruas, ainda assim garantiu melhor solução do que a rota atual.

Palavras-Chave: Problema do Carteiro Chinês, Grafos Eulerianos. Coleta de lixo urbano, Algoritmo de Fleury.

ABSTRACT

Optimizing routes for urban garbage trucks contribute to reducing the costs of public coffers. This work aims to present a simple program to solve the Chinese Postman Problem, acting in the determination of minimum routes for garbage collection vehicles and applied in two neighborhoods in the city of Recife-PE. The program receives the distance matrix of the graph and determines the optimal route in 3 steps: writing the mathematical model that solves the PCCD, solving it by a solver and plotting the route in an Eulerian graph. The result of the program was compared with the current routes: reduction of the total distance of 12.7% (Engenho do Meio) and 8.86% (Cordeiro). The tour for the garbage truck reduced the total distance, although some streets had to be traversed more than once to reach certain segments of streets, it still guaranteed a better solution than the current route.

Keywords: Chinese Postman Problem, Eulerian Graphs. Urban garbage collection, Fleury's algorithm.

1 INTRODUÇÃO

O aumento dos resíduos sólidos urbanos é um dos maiores problemas enfrentados nas grandes cidades, uma vez que está diretamente ligado ao crescimento populacional. Nesta perspectiva, a problemática da quantidade de resíduos sólidos gerados representa um dos grandes desafios para os gestores públicos (SOUZA et al., 2018), onde existe a necessidade da existência de políticas públicas competentes para a realização da coleta, do transporte e da destinação correta destes resíduos (JACINTO et al., 2014).

A realização da coleta dos RSU nas cidades é de responsabilidade dos órgãos públicos municipais (MORO et al., 2018) que, por muitas vezes, terceirizam esta atividade para empresas, que ainda partem do conhecimento profissional de seus funcionários para realizar a melhor rota (ou percurso) para a coleta destes resíduos nos bairros. Segundo Vecchi et al. (2015) a realização da coleta deve ser a mais bem planejada, por gerar entre 50% e 80% do custo de operação de limpeza pública. Por se tratar de uma atividade de grande importância na cadeia produtiva, a coleta de lixo é uma atividade logística importante dentro de qualquer cidade (GRAKOVA et al., 2018).

De acordo com a pesquisa da ABRELPE (2017), o Brasil em 2017 gerou um total de 78,4 milhões de toneladas de RSU, sendo coletados cerca de 71,6 milhões de toneladas de resíduos e 6,9 milhões de toneladas foram dispostos, possivelmente, para destinos impróprios. A geração total de resíduos do mesmo ano atingiu 214.868 toneladas diárias, o que representou um aumento de 0,48% em referência ao ano anterior, por um custo de R\$ 10,37 por habitante por mês, com todos os serviços de urbana. Para tanto, Beliën et al. (2012) ressaltam que a

coleta é a operação mais importante e cara do ciclo do descarte dos resíduos sólidos, em razão do trabalho intenso e utilização de caminhões no processo da coleta.

A coleta dos resíduos sólidos está relacionada aos problemas de otimização de rotas, uma vez que apresenta alto capital envolvido em termos de recursos financeiros, mão-de-obra e custos operacionais variáveis, que precisam ser minimizados. Assim, quanto melhor for a rota de coleta dos resíduos, maior será a redução nos custos, como também, dos efeitos ambientais gerados (emissão de gás carbônico do veículo coletor, alto consumo de combustível e acúmulo de lixo por não atender todas as ruas) (HANNAN et al., 2018).

A otimização de rotas é estudada pelos Problemas de Roteamento, que são separados em problemas de cobertura em arcos e em nós (WØHLK; LAPORTE, 2018). O principal objetivo dos problemas de cobertura em arcos está em determinar a rota de menor custo, para que todos os arcos de um grafo sejam percorridos ao menos uma vez, visto que o circuito pode ter restrições ou não (KONOWALENKO et al., 2012). Os problemas de roteamento em arcos, em especial o Problema do Carteiro Chinês (PCC) otimiza rotas a partir da cobertura de todos os arcos (segmentos de ruas), tornando a coleta de resíduos sólidos viável e minimizando os custos operacionais.

O PCC é tratado na literatura para resolver diversos problemas práticos, como: rota de carteiros (IRNICH, 2008); rotas para leituristas de ruas (USBERTI et al., 2008); leituristas de barragens (DOS SANTOS, 2016); inspeção e manutenção de linhas ferroviárias (YILMAZ et al., 2017); patrulhamento estratégico de seguranças (HOCHBAUM et al., 2014; SHAFABI; HAGHANI, 2015); serviços de inverno (remoção de neve, derramamento de sal em estradas, etc.) (LUKMAN et al., 2018); fiscalização de linhas de ônibus; serviços de transporte escolar, entre outros.

Diante do contexto, o presente trabalho apresenta um estudo sobre o PCC Direcionado (PCCD) com o objetivo de desenvolver um programa simples para resolução do PCCD para minimizar a rota do caminhão de lixo urbano. O programa foi implementado em linguagem de programação *Python 3.7*, aplicado em dois bairros na cidade de Recife-PE e comparado com as rotas atuais.

O trabalho se justifica pela abordagem de otimização para resolução de problemas práticos através de resolução de modelos de programação linear para aplicações práticas. Otimizar rotas para veículos coletores de lixo contribuem para a redução de custos atrelados ao consumo de combustível, emissão de gases poluentes, redução da jornada de trabalho para as coletas, desgaste de pneus e peças, etc. Os bairros utilizados na pesquisa seguem um padrão

empírico para sua rota e por muitas vezes, algumas ruas não são atendidas, isso leva a não eficiência da rota traçada, então a aplicação torna-se excepcional para alcance de melhorias.

A contribuição desse trabalho parte da elaboração de um programa que determina rotas ótimas, como suporte a coleta de lixo urbano, de forma mais prática e rápida, economizando tempo e dinheiro. O programa tem propósito de aplicar de forma simples a resolução do PCCD para redes urbanas através da escolha de bairros a serem roteados pelo caminhão de lixo, entrada de um grafo que represente o mapa, solucionar o problema e tratar uma rota para o veículo.

O trabalho está dividido em: seção 1 com a problemática dos resíduos sólidos, o objetivo e justificativa da pesquisa; seção 2 com o breve embasamento teórico sobre grafos, grafos Eulerianos e Problema do Carteiro Chinês com foco no caso direcionado; seção 3 apresenta a descrição das etapas para se alcançar os objetivos da modelagem; seção 4 aborda os resultados computacionais implementados com a aplicação do programa para dois bairros e as principais discussões e, por fim, a seção 5 com as considerações finais expostas com a aplicação do trabalho.

2 REFERENCIAL TEÓRICO

Um grafo pode ser representado por $G = (N, E \cup A)$, onde N é o conjunto de vértices (nós), E o conjunto de arestas e A o conjunto de arcos. Uma aresta é representada por um par não orientado $\{i, j\} \in E$, sendo i e j vértices do conjunto N que indicam uma ligação não direcionada entre eles. De forma análoga, um arco é formado por um par orientado $(i, j) \in A$, onde os vértices i e j pertencem ao conjunto N e indicam uma ligação direcionada do vértice i para j .

Um grafo dirigido é um grafo direcionado cujos arcos também são direcionados, um grafo não dirigido possui todas as arestas não direcionados, entretanto quando existem arcos e arestas no mesmo grafo, este é chamado de grafo misto (AHUJA et al., 1993).

Quando num grafo existir uma sequência de vértices distintos formando um percurso, então esta sequência é chamada de caminho. Para enfatizar onde o caminho começa e termina, uma sequência de vértices distintos $(i_1, i_2, \dots, i_{k-1}, i_k)$ representa um caminho de i_1 para i_k em G . Os nós i_1 e i_k são chamados de nó inicial (ou origem) e nó final (ou destino) de G , respectivamente (KELLER; TROTTER, 2015). Em um grafo dirigido, um caminho recebe o nome de caminho dirigido.

Quando o caminho $(i_1, i_2, \dots, i_{k-1}, i_k)$ apresentar o vértice inicial igual ao final ($i_1 = i_k$), independente da sequência dos arcos, recebe o nome de ciclo (BERTSEKAS, 1998). Já o circuito de um grafo é uma sequência de vértices $(i_1, i_2, \dots, i_{k-1}, i_k)$ onde $i_1 = i_k$ e i_i é adjacente a i_{i+1} (TUCKER, 2012), portanto é um caminho fechado. Um ciclo que percorre todos os arcos/arestas de um grafo exatamente uma vez, recebe o nome de ciclo Euleriano (ou de Euler). De forma análoga, se um grafo G apresentar um circuito em que cada um de seus arcos/arestas é percorrido apenas uma vez, este é um circuito Euleriano (BERTSEKAS, 1998).

A teoria dos grafos teve suas origens há quase três séculos em um problema na cidade de Königsberg na Prússia (atual cidade de Kaliningrad na Rússia). O rio Pregel que atravessa a cidade tem duas grandes ilhas, que foram conectadas a cidade por sete pontes. A população frequentemente se perguntava se era possível percorrer a cidade, de tal forma, que se pudesse atravessar as sete pontes uma única vez cada, o matemático Leonhard Euler, em 1736 solucionou este problema aplicando a teoria dos grafos (KELLER; TROTTER, 2015).

Euler observou que o problema das pontes de Königsberg poderia ser representado por um grafo não direcionado no qual cada aresta corresponde as sete pontes e os vértices exprimem cada uma das duas margens do rio das duas ilhas. Euler provou que a questão seria análoga a encontrar um caminho no grafo que envolvesse cada aresta exatamente uma vez, o que passou a ser chamado de caminho de Euler (caminho Euleriano).

Para o problema das sete pontes de Königsberg, visto que um caminho fechado requer que cada vértice tenha o mesmo número de arestas entrando (*indegree*) e arestas saindo (*outdegree*) deste, obviamente o caso não teve solução devido todos os vértices serem de grau ímpar (DROR, 2012). Porém, os grafos com a característica de apresentarem passeios fechados que percorrem todas as arestas exatamente uma vez recebem o nome de grafo unicursal ou Euleriano.

Como a atividade principal do problema das sete pontes de Königsberg era de percorrer certos arcos de uma rede de transporte, então este problema pode ser visto como a primeira aplicação de roteamento de arco (ASSAD; GOLDEN, 1995). Entre os Problemas de Roteamento de Arcos, o Problema do Carteiro Chinês está entre os mais famosos e mais abordados na literatura (SHAFABI; HAGHANI, 2015).

O Problema do Carteiro Chinês (PCC) é um problema de otimização que pertence à classe de problemas de roteamento de arcos, cujo objetivo é encontrar um *tour* em um grafo considerando a menor distância ou custo (DROR, 2012). Um *tour* é um circuito fechado em um grafo percorrendo todas as arestas de G , no mínimo uma vez. Em um grafo G , uma

sequência alternada de nós e arestas $(n_1, e_1, n_2, e_2, \dots, n_l, e_l, n_{l+1})$, de tal forma que $n_1 = n_{l+1}$, representa um *tour*. Um *tour* recebe o nome de *tour* de Euler, se, e somente se, este incluir cada aresta de G exatamente uma vez (ASSAD; GOLDEN, 1995). O *tour* do carteiro distingue-se do circuito Euleriano devido, no primeiro, ser permitida a repetição de arestas, quando necessário.

O PCC foi introduzido pela primeira vez por Meigu Guan, em um artigo chinês, onde Guan introduziu o problema de determinar a rota mais curta de um carteiro (SCHRIJVER, 2003). Guan foi um matemático chinês, que durante a Revolução Cultural Chinesa permaneceu algum tempo trabalhando como funcionário dos correios (EISELT et al., 1995). Pela contribuição com a introdução do tema e a publicação que motivou os problemas de otimização de rede, o problema ficou conhecido como o Problema do Carteiro Chinês, em sua homenagem (ASSAD; GOLDEN, 1995).

Conforme Assad e Golden (1995) o problema proposto por Guan envolvia adicionar um conjunto de arestas ao grafo para garantir que todos os nós do grafo tenham grau par. Ainda, segundos os autores, o problema proposto por Guan não produzia um algoritmo polinomial para o problema do carteiro. Assim, Edmonds (1965) entendeu que o problema de Guan em adicionar um conjunto de arestas de custo mínimo para obter um grafo com todos os nós de grau par poderia ser resolvido como um “problema de emparelhamento (*matching*)” de forma eficiente. Deste modo, Edmonds e Johnson (1973) desenvolveram algoritmos exatos para grafos não direcionados, direcionados e mistos, combinando dois algoritmos conhecidos de até então: o algoritmo de caminho mais curto e um algoritmo para o emparelhamento máximo.

Por definição, o PCC padrão é um problema no qual, dado um grafo $G = (N, E)$ cujas arestas (i, j) possuem um comprimento não negativo $c_{ij} > 0$, deseja-se identificar um caminho de comprimento mínimo que se inicie em algum nó, passe por todos os arcos de G pelo menos uma vez e retorne ao nó inicial (AHUJA et al., 1993).

O PCC pode ser aplicado em grafos não direcionados, direcionados e mistos. Para cada caso, existem requisitos necessários para garantir a unicursalidade de um grafo e, portanto, que seja determinado o *tour* do carteiro. Eiselt et al. (1995) apresentam essas condições:

- Grafo não direcionado: o grafo deve ser conexo e apresentar todos os vértices de grau par, ou seja, ter as condições propostas por Guan (1962);

- Grafo direcionado: o grafo precisa ser fortemente conexo e todos os nós ter grau par, ou seja, o número de arcos incidentes a um nó deve ser igual ao número de arcos saindo do mesmo nó (*indegree* igual ao *outdegree*);

- Grafo misto: grafo fortemente conexo e todos os vértices precisam ter grau par para as arestas direcionadas e não direcionadas, tornando o grafo balanceado.

O PCC em grafos não direcionadas e direcionadas possuem algoritmos e modelos matemáticos que apresentam solução exata em tempo polinomial, já para grafos mistos, o problema é considerado *NP-completo* (WANG; WEN, 2002). Como o foco do trabalho são os grafos direcionados para aplicação do PCC, então será abordado mais sobre esse caso.

O Problema do Carteiro Chinês Direcionado (PCCD) é representado por um grafo direcionado $G = (N, A)$ que possui arcos com comprimento não negativo c_{ij} para todo $(i, j) \in A$. Para sua resolução é necessária a garantia de que G seja fortemente conexo (EISELT et al., 1995; WANG; WEN, 2002). Outra condição necessária e suficiente para a existência de um *tour* de Euler para o PCCD é que o grafo deve ser simétrico, ou seja, o número de arcos que entram em um determinado nó deve ser igual ao número de arcos saindo desse nó (ASSAD; GOLDEN, 1995).

Portanto, quando as duas condições necessárias são satisfeitas, encontrar um *tour* de Euler pode ser facilmente resolvido por qualquer algoritmo que encontre um circuito Euleriano em um grafo não direcionado adaptado para grafos direcionados. Como é o caso do algoritmo de Fleury presente em Christofides (1975) ou o algoritmo simples proposto por Ahuja et al. (1993) para determinar o *tour* de Euler em grafos Eulerianos direcionados. Entretanto, estes algoritmos servem apenas para grafos Eulerianos.

Nos casos mais comuns do contexto prático, o grafo não é Euleriano, para tanto é necessário torná-lo Euleriano. Para isso, de acordo com Assad e Golden (1995), cópias adicionais de alguns arcos precisam ser incluídas para realizar o *tour*. Tal procedimento pode ser feito atribuindo a uma variável de decisão x_{ij} na qual representa o número de vezes que o arco (i, j) é percorrido em G . Para isso, pode-se tratar o caso como um problema de fluxo de custo mínimo, onde o resultado estabelece o menor custo total do *tour* do carteiro (AHUJA et al., 1993; ASSAD; GOLDEN, 1995). Wang e Wen (2002) utilizaram o modelo matemático abordado para resolução exata do PCCD por meio de programação linear.

$$\text{Minimize} \quad \sum_{(i,j) \in A} c_{ij} x_{ij} \quad (01)$$

Sujeito a:

$$\sum_{(i,j) \in A} x_{ij} - \sum_{(i,j) \in A} x_{ji} = 0, \quad \forall i \in N \quad (02)$$

$$x_{ij} \geq 1 \text{ e inteiro}, \quad \forall (i,j) \in A \quad (03)$$

No modelo matemático, a função objetivo (01) é minimizar o custo total do *tour* do carteiro, onde c_{ij} é a capacidade da aresta (i,j) que na prática pode representar custo, distância, etc. As restrições: (02) garantem a conservação de fluxo, cada nó tem seu *indegree* igual ao seu *outdegree* e; (03) permitem que cada arco (i,j) seja percorrido pelo menos uma vez em G e que a variável pertence ao conjunto de números inteiros. Uma vantagem deste modelo matemático é que sua matriz de restrições, conforme abordado por Wang e Wen (2002), é unimodular, essa propriedade garante que a resolução dada pela programação inteira seja a mesma solução da programação linear no caso de a variável inteira ser relaxada, tornando o método de solução mais rápido, especialmente para grafos de grande porte.

O PCCD também pode ser formulado e resolvido em programação matemática como um problema clássico de transportes, como abordado por Beltrami e Bodin (1974). Seja qual a abordagem para determinar quais os arcos serão adicionados, após tal procedimento, pode-se utilizar os diversos algoritmos existentes na literatura para traçar o *tour* de Euler, como aqui abordado. Trabalhos de Baker (1983) e Lin e Zhao (1988) também apresentam outras propostas de algoritmos para resolução eficiente do PCCD.

Os algoritmos de construção de rotas para traçar o circuito Euleriano são bem tratados na literatura, sendo um dos mais conhecidos o algoritmo de Fleury para grafos Eulerianos (KAUFFMAN, 1967). O algoritmo de Fleury é considerado um dos mais fáceis de serem implementados, embora só possa ser aplicado em grafos Eulerianos ou semi-eulerianos (quando no grafo existe apenas dois nós de grau ímpar, forçando o caminho a iniciar em um desses nós e finalizar no outro) (SAOUB, 2017), então a priori, já deve-se saber se o grafo possui esta característica antes de traçar a rota. O algoritmo segue o seguinte procedimento:

Passo 1. Construir uma lista vazia T que será armazenada o *tour* do carteiro.

Passo 2. Escolher um nó arbitrário e adicioná-lo na lista T .

Passo 3. Escolher e marcar uma aresta incidente ao nó escolhido, desde que o grafo que resulte da escolha permaneça conexo.

Passo 4. Rotular a aresta marcada no grafo e adicionar o nó a aresta no final da lista T .

Passo 5. Se todas as arestas do grafo original tiverem sido rotuladas, então retorne a lista T que conterà o circuito Euleriano. Se não, tome um nó adjacente ao nó escolhido anteriormente e volte ao **passo 3**.

Segundo Moro et al. (2018) o PCC pode ser facilmente resolvido pelo algoritmo de Fleury, no qual é fácil de ser implementado em linguagem de computador, a partir de uma matriz de adjacências.

3 PROCEDIMENTO METODOLÓGICO

A pesquisa segue uma modelagem na qual determinará a rota para a coleta de lixo urbano para dois bairros da cidade de Recife, capital do estado de Pernambuco. Inicialmente, foi feito um breve estudo da literatura sobre os principais temas envolvidos: grafos, roteamento de arcos e Problema do Carteiro Chinês (PCC).

A presente pesquisa é de natureza aplicada e tem seus objetivos como exploratória, sua abordagem é quantitativa e seu procedimento trata-se de uma modelagem. A modelagem utiliza-se de técnicas matemáticas para descrever o funcionamento de um sistema produtivo ou parte deste. A modelagem, segundo Ganga (2012), se defini como um modelo de uma representação de um sistema real, modelo este essencial ao processo da tomada de decisão, uma vez que foram obtidos dados reais como variáveis e restrições para processar as informações de um problema fornecendo uma solução otimizada ou viável para o cenário em estudo.

Para resolução do PCCD, foi desenvolvido um programa simples, que receberá a matriz de distâncias do grafo e determinará imediatamente o percurso ótimo do caminhão de lixo. Este procedimento será resolvido através de 3 passos, como mostra o Quadro 1. Para cada passo é necessário introduzir *inputs* (dados de entrada), descrição do procedimento e *outputs* (dados de saída ou resultado).

Quadro 1 – Procedimento em três passos para determinar o *tour* do caminhão

<i>Procedimento do programa para determinação de rotas</i>
Passo 1. Entrada: Matriz das distâncias dos segmentos de ruas (comprimento dos arcos). Processo: Escrever o Modelo de Programação Linear (MPL) para o PCCD em um arquivo de texto. Saída: Arquivo com extensão (.lp) para entrada no <i>solver</i> .

Passo 2.

Entrada: Arquivo com extensão (.lp) para entrada no *solver*.

Processo: Resolver o MPL, com o objetivo de determinar o número de vezes que cada segmento de rua deverá ser percorrido para minimizar a distância total do trajeto do caminhão de lixo.

Saída: Quantidade de vezes que cada segmento de rua é percorrido e a distância mínima total.

Passo 3.

Entrada: Número de vezes que cada segmento de rua é percorrido e a distância mínima.

Processo: Sequenciar o percurso do caminhão de lixo percorrendo todas as ruas, por meio do algoritmo de Fleury modificado.

Saída: *Tour* para o caminhão.

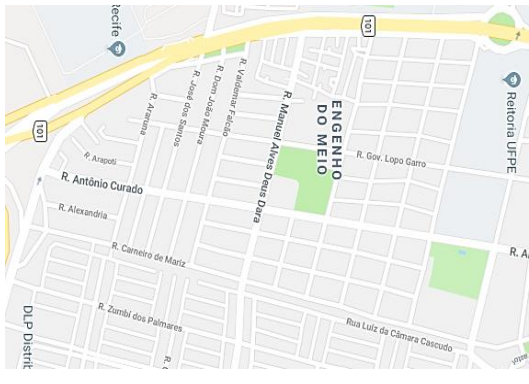

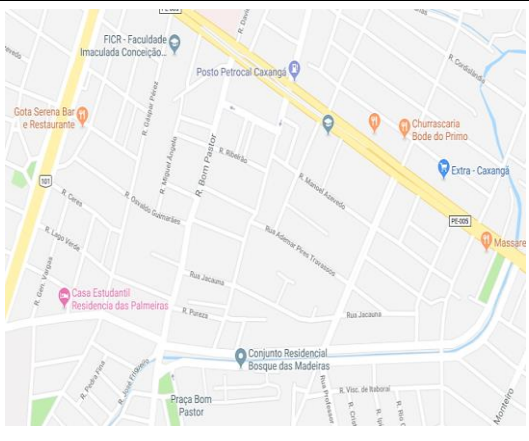

Fonte: Elaborado pelos autores (2020)

Para aplicação e validação dos dados do programa, será utilizado emprego prático para dois bairros da cidade de Recife. A escolha dos bairros foi, principalmente, por serem de pequeno porte e por apresentarem a grande maioria das ruas com uma única direção, sendo de interesse para o PCCD. As ruas com mão dupla (ruas que podem ser atravessadas nos dois sentidos) foram acrescentadas no grafo, arcos indo e voltando, ou seja, o caminhão deverá percorrer tais segmentos de ruas nos dois sentidos, isso se justifica pelo fato de serem ruas muito largas, o que inviabiliza a coleta em um único sentido. Os bairros escolhidos foram: Engenho do Meio e Cordeiro.

As informações sobre as rotas de coleta de lixo urbano nos bairros foram fornecidas pela Secretaria de Infraestrutura e Serviços Urbanos de Recife. A representação dos bairros em forma de grafo será coletada por dados geográficos. Os dados geográficos serão coletados a partir do uso do *software Open Street Maps*.

O *Open Street Maps (OSM)* é um *software* cooperativo *online* que apresenta uma visualização em formato de mapa de todo o globo terrestre. A partir dele, se torna possível selecionar locais e cadastrar pontos no mapa do *OSM* para transformar os segmentos de ruas em arcos e as esquinas (cruzamentos de segmentos de ruas) em nós e gerar uma matriz de distâncias para representá-lo em forma de grafo. A saída do programa é um arquivo .xlsx, formato de planilha do *Microsoft Office Excel*. Na Figura 1 é apresentado a visualização dos dois bairros no mapa e a sua respectiva representação do grafo no *OSM*.

Figura 1 – Representação dos bairros em estudo no OSM

Bairro Engenho do Meio	Representação no OSM
	
Bairro Cordeiro	Representação no OSM
	

Fonte: Elaborado pelos autores (2020)

A versão original do algoritmo de Fleury é destinada a grafos não direcionadas, porém, através de uma adaptação, ela pode ser aplicada em grafos direcionados. O pseudocódigo para o algoritmo de Fleury modificado para grafos Eulerianos é destacado na Figura 2.

Figura 2 – Pseudocódigo do Algoritmo de Fleury para grafos direcionados

Algoritmo de Fleury: Modificado para grafos direcionados

Entrada:
 $G = (N, A)$ // Grafo Euleriano
 $G' = G$ com $G' = (N', A')$ // Cópia do grafo original para retornar o *tour*
 $T = \{\emptyset\}$ // Uma lista vazia T para armazenar o *tour*
 $n_1 \in N'$ // O nó inicial pertence ao grafo G'

Início:
 $T = [n_1]$ // O *tour* inicia no nó inicial
 $i = n_1$ // O nó i recebe o nó inicial
Enquanto $A \neq \{\emptyset\}$ **faça:** // Enquanto o conj. de arcos não for vazio, faça
 Escolher um arco $a(i, j)$ adjacente de i , desde que não seja uma ponte
 $T = T \cup j$ // Acrescenta o nó rotulado ao *tour*
 $A' = A' - (i, j)$ // Retirar o arco $a(i, j) \in A'$ do grafo
 $i = j$ // Será atribuído ao nó i o nó da próxima iteração j
fim do enquanto
Retorne T

fim

Saída:
 T contendo um *tour* Euleriano

Fonte: Elaborado pelos autores (2020)

Após a geração da solução, a resposta do programa será comparada com as rotas atuais para os bairros conforme os dados fornecidos pela empresa prestadora de serviço da cidade para a coleta de lixo. Essa etapa contribuirá com a validação do modelo proposto.

4 RESULTADOS COMPUTACIONAIS

O programa elaborado foi implementado em linguagem de programação *Python* com o Modelo de Programação Linear (MPL), onde foi executado em um microcomputador com uma CPU Intel Core i3 com 2 GHz, 4 GB de RAM e sistema operacional Windows 10 Home (x64 bits). Foi utilizado o *software Python* versão 3.7. O MPL foi resolvido pelo *solver Soplex*, presente no pacote do *software SCIP Optimization Suite*. E o algoritmo de Fleury foi implementado no programa em *Python*.

O grafo representativo para o bairro Engenho do Meio dado pelo *OSM* será chamado aqui de GEM, assim como, para o bairro Cordeiro será chamado de GCo. Os dados de saída do *OSM* para o GEM são exemplificados na Tabela 1 para os primeiros oito arcos, e segue para todos os demais. “Nó inicial” representa o *outdegree* do nó i e “Nó final” representa o *indegree* no nó j .

Tabela 1 – Matriz de distâncias para o bairro Engenho do Meio

Nó inicial	Nó final	Distância (metros)
1	112	132
2	1	55
2	14	97
3	2	56
4	3	63
4	16	96
5	4	58
6	5	51
...

Fonte: Elaborado pelos autores (2020)

O grafo GEM apresentou 115 nós e 200 arcos, enquanto que GCo declarou 332 nós e 604 arcos. Logo, o MPL identificou 200 variáveis de decisão para o GEM e 332 variáveis de decisão para o GCo. Em seguida, o modelo criou o código MPL e resolveu pelo *Soplex*, que determinou a distância total de cada percurso para os bairros. As respostas são mostradas na Figura 3.

Figura 3 – Resposta do *Soplex* para os grafos

GEM	GCo
<pre> 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 --- unscaling 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 </pre>	<pre> 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 --- unscaling 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 </pre>

Fonte: Elaborado pelos autores (2020)

Segundo a Figura 3, o modelo chegou ao ótimo nos dois casos, com 23,317 km a serem percorridos no bairro Engenho do Meio e 89,643 km para cobrir todo o bairro Cordeiro. Comparando as respostas do MPL para os dois casos e a abordagem utilizada pela empresa coletora do lixo urbano nestes bairros, pode-se comparar o observado e o esperado (Tabela 2).

O programa se mostrou eficiente pela sua rápida resposta, mesmo para o bairro Cordeiro com uma quantidade bem maior de vértices e arestas representadas no grafo e que obteve alguns arcos sendo necessário ser percorridos 3 vezes para alcançar os demais trechos de ruas.

Tabela 2 – Comparação dos resultados do programa e a rota atual

Bairros/Grafos	Tempo de resolução (seg)	Solução MPL (km)	Rota atual (km)	Diferença percentual (%)
GEM	0,721	23,317	26,71	12,70
GCo	0,932	89,643	98,36	8,86

Fonte: Elaborado pelos autores (2020)

A Tabela 2 mostra o quanto a rota traçada pelo programa gerou uma solução factível e de qualidade ótima comparada com a distância total da rota utilizada pela atual empresa. Essa redução se propaga de forma direta não apenas no percurso reduzido, mas em menor tempo de execução da coleta e na redução do consumo do óleo combustível do caminhão, e de forma indireta, com a emissão de gases poluentes, manutenção do veículo, quebra de peças e troca de pneus.

Conforme já havia sido confirmado pela empresa da coleta, que as vezes certas ruas não eram percorridas, o *tour* da solução alcançou menor distância alcançando todas as ruas, embora algumas ruas precisem ser percorridas mais de uma vez para alcançar certos segmentos de ruas, ainda assim garantiu melhor solução do que a rota atual. A garantia de que todas as ruas com demanda de lixo serão percorridas assegura maior satisfação dos moradores com as coletas em dias em sua região.

O *tour* do caminhão de lixo conforme traçado pela terceira etapa do programa, descreve de forma precisa a ordem de ruas que deverão ser visitadas pelo veículo. Abaixo, apresenta-se no Quadro 2 a sequência para o bairro Engenho do Meio dada como resposta pelo programa.

Quadro 2 – Sequência de nós para o bairro Engenho do Meio

<i>Resposta ótima encontrada para o Bairro Engenho do Meio</i>
1-112-113-6-5-4-3-2-14-26-38-39-40-41-42-58-109-70-69-68-74-61-60-111-68-74-67-66-63-62-67-66-65-75-83-84-85-81-82-76-74-61-74-76-77-78-73-72-71-70-69-110-59-58-109-70-77-81-86-87-88-96-88-89-98-89-90-85-82-76-82-85-90-99-95-97-95-94-79-80-81-83-84-92-103-101-102-101-100-99-90-85-86-87-93-80-81-86-87-93-94-79-78-73-104-105-106-53-37-38-39-27-15-3-2-1-112-113-114-9-8-7-6-18-19-7-6-18-30-29-17-5-4-16-17-18-30-42-43-31-19-20-21-9-8-20-32-31-30-42-43-44-45-33-21-9-114-115-12-11-10-22-23-11-10-9-21-9-114-115-12-24-36-35-23-24-36-48-49-50-47-35-34-33-21-22-34-46-47-48-49-64-75-76-74-61-52-45-46-51-6251-52-45-52-61-60-59-58-57-56-55-54-53-37-25-13-14-15-16-28-40-41-29-28-27-26-25-13-1.

Fonte: Elaborado pelos autores (2020)

5 CONSIDERAÇÕES FINAIS

Este trabalho apresentou um programa para resolução do Problema do Carteiro Chinês Direcionado atuando na determinação de rotas mínimas para veículos coletores de lixo urbano. O programa descreve a rota ótima em três passos: escrever o Modelo de Programação Linear que determina a quantidade mínima de vezes que cada segmento de rua deve ser percorrido, resolvê-lo por um solucionador de Programação Linear e traçar a rota em um grafo Euleriano.

O programa foi implementado em linguagem de programação *Python* e aplicado em dois bairros na cidade de Recife-PE (Engenho do Meio e Cordeiro). O resultado do programa foi comparado com as rotas atuais empregadas pela empresa local responsável pela coleta de lixo da cidade. As respostas se mostraram bem melhores do que as rotas atuais com uma redução da distância total de 12,7% para o bairro Engenho do Meio e 8,86% para o Cordeiro. Essas reduções contribuem de forma direta na tomada de decisão para o percurso do caminhão da coleta e na economia dos custos relacionados ao consumo de combustível, a jornada de trabalho para as coletas, desgaste de pneus e peças, como também, diminuir a emissão de gases poluentes providos pelo veículo. Por muitas vezes, algumas ruas deixavam de ser atendidas, levando a reclamação dos moradores, o que no modelo não ocorre devido todos os segmentos de ruas serem percorridos pelo menos uma vez.

A principal limitação do trabalho foi trabalhar com a maioria das ruas em sentido único, o que faz com que alguns segmentos de ruas precisassem ser percorridos mais de uma vez para balancear a rota e atingir todas as ruas do bairro. Mesmo assim, o modelo ainda foi melhor do que a prática. Para trabalhos futuros, pretende-se expandir a aplicação do programa para grafos não direcionados (para traçar rotas para carteiros ou leituristas) e grafos mistos (onde veículos precisam atender ruas com sentido único e sentido duplo no mesmo bairro). Outra sugestão é aplicar em mais bairros da cidade do estudo para determinar rotas ótimas e compará-las com as rotas atuais e por fim, traçar uma estratégia logística futura.

AGRADECIMENTOS

Os autores agradecem ao apoio e financiamento da Coordenação de Aperfeiçoamento de Pessoal de Nível Superior (CAPES) e do Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq) para a realização deste trabalho.

REFERÊNCIAS

ABRELPE – Associação Brasileira de Empresas de Limpeza Pública e Resíduos Especiais. (2017). Panorama dos resíduos sólidos no Brasil. Disponível em: <<http://abrelpe.org.br/download-panorama-2017/>>. Acessado: 2019-02-01.

Ahuja, R., Magnanti, T. e Orlin, J. (1993). Network Flow: theory, algorithms, and applications. Upper Saddle River, New Jersey: Prentice Hall.

Assad, A. A. e Golden, B. L. (1995). Arc routing methods and applications. *Handbooks in operations research and management science*, 8:375-483.

Baker, E. K. (1983). An exact algorithm for the time constrained travelling salesman problem. *Operations Research*, 31(5):938-945.

Beliën, J., De Boeck, L. e Van Ackere, J. (2012). Municipal solid waste collection and management problems: a literature review. *Transportation Science*, 48(1):78–102.

Beltrami, E. e Bodin, L. (1974). Networks and vehicle routing for municipal waste collection. *Networks*, 4:65-94.

Bertsekas, D. P. (1998). *Network Optimization: Continuous and Discrete Models*. Belmont: Athena Scientific.

Christofides, N. (1975). *Graph Theory: An Algorithmic Approach*. Academic Press, London.

Dos Santos, C. G. (2016). Uma proposta de modelagem matemática para um problema de roteirização periódica em arcos capacitados com múltiplas tarefas. Tese (Doutorado) – Programa de Pós-Graduação em Métodos Numéricos em Engenharia, Universidade Federal do Paraná. Curitiba (PR).

Dror, M. (Ed.). (2012). *Arc routing: theory, solutions and applications*. Springer Science & Business Media, New York.

Edmonds, J. (1965). The Chinese postman problem. *Operations Research*, 13, Supplement 1, B-73.

Edmonds, J. e Johnson, E. L. (1973). Matching, Euler tours and the Chinese postman. *Mathematical programming*, 5(1):88-124.

Eiselt, H.A., Gendreau, M. e Laporte, G. (1995). Arc routing problems. Part I: The Chinese postman problem. *Operations Research*, 43(2):231-242.

Ganga, G. M. D. (2012). Trabalho de conclusão de curso (TCC) na engenharia de produção: um guia prático de conteúdo e forma. São Paulo: Atlas.

Grakova, E. et al. (2018). Waste Collection Vehicle Routing Problem on HPC Infrastructure. In: *IFIP International Conference on Computer Information Systems and Industrial Management*, Springer, Cham, p. 266-278.

Guan, M. (1962). Graphic programming using odd or even points. *Chinese Math*, 1:273-277.

Hannan, M. A. et al. (2018). Capacitated vehicle-routing problem model for scheduled solid waste collection and route optimization using PSO algorithm. *Waste Management*, 71:31-41.

Hochbaum, D. S., Lyu, C. e Ordóñez, F. (2014) Security routing games with multivehicle Chinese postman problem. *Networks*, 64(3):181-191.

Irnich, S. (2008). Solution of real-world postman problems. *European Journal of Operational Research*, 190:52–67.

Jacinto, J. P., Rosa, R. A. e Banos, R. S. (2014). Heurística para solução do problema da coleta de resíduos sólidos domiciliares (RSD) com base no problema do carteiro chinês capacitado com múltiplas viagens (PCCC-MV). *TRANSPORTES*, 22(1):44-55.

Kauffman, A. (1967). *Graphs, Dynamic Programming and Finite Games*. Academic Press, New York.

Keller, M.T. e Trotter, W.T. (2015). *Applied Combinatorics*. Georgia Institute of Technology: Preliminary Edition.

Konowalenko, F., Benevides, P. F., Costa, D. M. B., Barboza, A. O. e Nunes, L. F. (2012). Aplicação de um algoritmo genético para o problema do carteiro chinês em uma situação real de cobertura de arcos. *Revista Ingeniería Industrial*, 11(1).

Lin, Y. e Zhao, Y. (1988). A New Algorithm for the directed Chinese Postman Problem. *Computers & Operations Research*, 15(6):577-584.

Lukman, R. K., Cerinšek, M., Vrtič, P. e Horvat, B. (2018). Improving efficient resource usage and reducing carbon dioxide emissions by optimizing fleet management for winter services. *Journal of cleaner production*, 177:1-11.

Moro, M. F., Andrade, F. A., Dos Santos, B. M., Neto, C. R. P. e Battisti, J. F. (2018). O problema do carteiro chinês aplicado na otimização das rotas de coleta de resíduos recicláveis: um estudo de caso. *Tecno-Lógica*, 22(2):128-135.

Saoub, K. R. (2017). *A Tour Through Graph Theory*. Chapman and Hall/CRC.

Schrijver, A. (2003). *Combinatorial Optimization: polyhedra and efficiency. Algorithms and Combinatorics*. v. 24. Springer Science & Business Media, Berlin.

Shafahi, A. e Haghani, A. (2015). Generalized maximum benefit multiple chinese postman problem. *Transportation Research Part C: Emerging Technologies*, 55:261-272.

Souza, P. S., Gonçalves, N. A. L. e Curi, R. C. (2018). Gestão dos resíduos sólidos no Município de Queimadas (Estado da Paraíba, Nordeste do Brasil) segundo a Política Nacional de Resíduos Sólidos. *Revista Brasileira de Gestão Ambiental e Sustentabilidade*, 5(10):739-752.

Tucker, A. (2012). *Applied combinatorics*. Wiley, 6 ed. New York.

Usberti, F. L., França, P. M. e França, A. L. M. (2008). Roteamento de leituristas: Um problema NP-difícil. In: *Anais do XL SBPO*, João Pessoa-PB. SOBRAPO.

Vecchi, T. P. B. et al. (2015). Uma abordagem sequencial para otimização de rotas dos caminhões de coleta de resíduos sólidos. In: *Congresso de Métodos Numéricos em Engenharia*, Lisboa. Disponível em: <http://www.dem.ist.utl.pt/cm2015/html/CD-Proceedings/PDF/Papers/CMN_2015_submission_255.pdf>. Acessado: 2019-03-03.

Wang, H. F. e Wen, Y. P. (2002). Time-constrained Chinese postman problems. *Computers and Mathematics with applications*, 44:375-387.

Wøhlk, S. e Laporte, G. (2018). A fast heuristic for large-scale capacitated arc routing problems. *Journal of the Operational Research Society*, 68(12):1877-1887.

Yilmaz, M., Çodur, M. K. e Yilmaz, H. (2017). Chinese postman problem approach for a large-scale conventional rail network in Turkey. *Tehnicki Vjesnik-Technical Gazette*, 24(5):1471-1477.