

UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ  
LICENCIATURA EM MATEMÁTICA

STEPHANY PRISCILA CORREIA

**OTIMIZAÇÃO DE ROTAS PARA A ENTREGA DE CORRESPONDÊNCIAS**

TRABALHO DE CONCLUSÃO DE CURSO

CORNÉLIO PROCÓPIO  
2019



STEPHANY PRISCILA CORREIA

## **OTIMIZAÇÃO DE ROTAS PARA A ENTREGA DE CORRESPONDÊNCIAS**

Trabalho de Conclusão de Curso apresentado ao Curso de Licenciatura em Matemática da Universidade Tecnológica Federal do Paraná como requisito parcial para obtenção do grau de “Licenciado em Matemática”.

Orientador: Prof<sup>a</sup>. Dr<sup>a</sup>. Claudia Fink

CORNÉLIO PROCÓPIO

2019





Ministério da Educação  
**Universidade Tecnológica Federal do Paraná**  
Câmpus Cornélio Procópio  
Diretoria de Graduação  
Departamento de Matemática  
Curso de Licenciatura em Matemática



---

## FOLHA DE APROVAÇÃO

### BANCA EXAMINADORA

---

Prof<sup>a</sup>. Dr<sup>a</sup>. Claudia Fink  
(Orientador)

---

Prof.<sup>o</sup> Dr<sup>o</sup> Josimar da Silva Rocha

---

Prof<sup>a</sup>. Dr<sup>a</sup>. Glauca Maria Bressan



## AGRADECIMENTOS

Algumas pessoas marcam a nossa vida para sempre, umas porque nos vão ajudando na construção, outras porque nos apresentam projetos de sonho e outras ainda porque nos desafiam a construí-los.

Esta fase da minha vida é muito especial e não posso deixar de agradecer a Deus por toda força, ânimo e coragem que me ofereceu para ter chegado até aqui. A Ele eu devo minha gratidão.

Aos meus pais Reinaldo e Adriana, meus maiores mestres, cujos ensinamentos guardo com grande carinho e consideração para toda vida. Se hoje tenho este título, dedico a vocês.

Aos meus irmãos Stély e Reinaldo Júnior, que sempre torceram por mim, mesmo que de longe.

A minha orientadora, Prof<sup>a</sup>. Dr<sup>a</sup>. Claudia Fink, pela orientação, confiança, dedicação, paciência e, principalmente, pela amizade e ensinamentos durante todo o período do trabalho.

Só tenho agradecer as minhas amigas, em especial Mirian, que esteve comigo ao longo da vida. As minhas amigas Jéssica, Glaucia e Débora, que conheci durante a graduação. Obrigada por todos os momentos em que fomos estudiosas, brincalhonas e cúmplices. É muito gratificante quando você percebe que tem alguém pra te ajudar, pra estar do seu lado. Obrigada pela paciência, pelo sorriso, pelo abraço, pela mão que sempre se estendia quando eu precisava. Esta caminhada não seria a mesma sem vocês. Minha eterna gratidão.

Agradeço também aos membros da banca examinadora, Prof<sup>a</sup>. Dr<sup>a</sup>. Glaucia Maria Bressan e o Prof<sup>o</sup>. Dr<sup>o</sup>. Josimar da Silva Rocha pela disponibilidade de participar e pelas contribuições valiosas.

A todos que direta ou indiretamente fizeram parte da minha formação, o meu muito obrigada.



*"Você só vive uma vez.  
É sua obrigação  
aproveitar a vida  
da melhor forma possível."  
(Como eu era antes de você)*



## RESUMO

CORREIA, Stephany Priscila. **OTIMIZAÇÃO DE ROTAS PARA A ENTREGA DE CORRESPONDÊNCIAS**. 2019. 58 f. Trabalho de Conclusão de Curso (Graduação) – Licenciatura em Matemática. Universidade Tecnológica Federal do Paraná. Cornélio Procópio, 2019

O presente trabalho tem por objetivo apresentar e implementar o algoritmo do Problema do Carteiro Chinês (PCC), que consiste em determinar um caminho mínimo que se inicie em algum vértice do grafo, passe por todas as arestas ao menos uma vez e retorne ao seu vértice inicial. Para contextualizar tal problema, utilizou-se o percurso de um carteiro em um bairro do Município de Bandeirantes, oeste do Paraná, para otimizar a rota percorrida por ele. Realizou-se previamente um estudo sobre a Teoria dos Grafos e o Problema do Carteiro Chinês (PCC) de acordo com suas variações. Foram utilizados os software Excel, LINDO, DEV-C++ e Xpress para implementar o algoritmo do Carteiro Chinês Não Direcionado (PCCND). O algoritmo desenvolvido foi aplicado no problema real de entrega de correspondências e também no exemplo do Problemas das Pontes de Königsberg.

**Palavras-chave:** Teoria dos Grafos. Problema do Carteiro Chinês (PCC). Problema do Carteiro Chinês Não Direcionado (PCCND) Roteirização. Entrega de Correspondências.



## ABSTRACT

CORREIA, Stephany Priscila. **ROUTING OPTIMIZATION FOR POSTAL DELIVERY SERVICES**. 2019. 58 f. Trabalho de Conclusão de Curso (Graduação) – Licenciatura em Matemática. Universidade Tecnológica Federal do Paraná. Cornélio Procópio, 2019

The present work aims to present and implement the algorithm of the Chinese Postman Problem (CPP), which consists of determining a minimum path that starts at some vertex of the graph, passes through all the edges at least once and returns to the initial vertex of it. To contextualize this problem, the route of a postman in a neighborhood in Bandeirantes city, western Paraná, was used to optimize the route traveled by him. A study on the theory of graphs and the problem of the Chinese postman according to its variations was previously carried out. Excel, LINDO, DEV-C++ and Xpress software was used to implement the Non-Directed Chinese Postman algorithm. The developed algorithm was applied in the real problem of correspondence delivery and also in the example of Problems of the Königsberg Bridges.

**Keywords:** Graphs theory. Chinese Postman Problem (CPP). Non-Directed Chinese Postman Problem. (NDCPP). Scripting. Correspondence Delivery.



## LISTA DE FIGURAS

FIGURA 1 – Mapa do Paraná, Bandeirantes em Destaque . . . . .	23
FIGURA 2 – Mapa de Bandeirantes. Divisões de dois lados A e B . . . . .	25
FIGURA 3 – Exemplo de representação de um grafo. . . . .	28
FIGURA 4 – Exemplo de grafos simples. . . . .	28
FIGURA 5 – Exemplo de representação gráfica de uma rede orientada. . . . .	29
FIGURA 6 – Grafo Conexo. . . . .	30
FIGURA 7 – Grafo Desconexo. . . . .	30
FIGURA 8 – Exemplos de árvore geradora do grafo da Figura 4. . . . .	31
FIGURA 9 – Exemplo de subgrafos com três arcos que não são árvore geradora do grafo da Figura 4. . . . .	31
FIGURA 10 – Grafo Euleriano. . . . .	33
FIGURA 11 – A cidade de Königsberg. . . . .	33
FIGURA 12 – Grafo que representa a cidade de Königsberg. . . . .	34
FIGURA 13 – Grafo Euleriano. . . . .	35
FIGURA 14 – Circuito Euleriano. . . . .	35
FIGURA 15 – Modelo matemático do PCCND gerado com código no DEV-C++ implementado no LINDO . . . . .	40
FIGURA 16 – Solução do Problema das Pontes de Königsberg utilizando o modelo de Bodin (1983) no software LINDO. . . . .	41
FIGURA 17 – Imagem da região 364-B e o grafo gerado para esta região . . . . .	43
FIGURA 18 – Código escrito em C para gerar a formulação matemática no formato lido pelo Xpress . . . . .	44
FIGURA 19 – Modelo matemático reproduzido no software Xpress . . . . .	45



## LISTA DE TABELAS

TABELA 1 – Divisões de cada carteiro . . . . .	24
TABELA 2 – Grau de cada vértice. . . . .	28
TABELA 3 – Matriz Distância. . . . .	40
TABELA 4 – Solução obtida usando o Xpress para resolver o problema do PCCND gerado	45



## LISTA DE SIGLAS E ABREVIATURAS

DP	Distrito Postal do carteiro
ECT	Empresa Brasileira de Correios e Telégrafos
LINDO	(Linear, Interactive and Discrete Optimizer)
PCC	Problema do Carteiro Chinês
PCCND	Problema do Carteiro Chinês Não Orientado
PCV	Problema do Caixeiro Viajante



## SUMÁRIO

<b>1</b>	<b>INTRODUÇÃO</b>	<b>21</b>
1.1	OBJETIVOS	22
1.1.1	OBJETIVO GERAL	22
1.1.2	OBJETIVOS ESPECÍFICOS	22
1.2	JUSTIFICATIVA	22
1.3	ORGANIZAÇÃO DO TRABALHO	22
<b>2</b>	<b>COLETA DE DADOS NO MUNICÍPIO DE BANDEIRANTES</b>	<b>23</b>
2.1	CARACTERIZAÇÃO DO MUNICÍPIO DE BANDEIRANTES	23
2.2	DESCRIÇÃO DO PROCESSO DE ENTREGA DAS CORRESPONDÊNCIAS	23
2.3	DESCRIÇÃO DO PROBLEMA A SER ESTUDADO	25
<b>3</b>	<b>FUNDAMENTAÇÃO TEÓRICA</b>	<b>27</b>
3.1	GRAFOS	27
3.1.1	REPRESENTAÇÃO MATRICIAL DE GRAFOS	32
3.2	PROBLEMAS DE ROTEAMENTO DE ARCOS	32
3.3	GRAFOS DE EULER E O PROBLEMA DAS PONTES DE KÖNIGSBERG	32
3.4	CIRCUITOS EULERIANOS	34
3.5	PROBLEMA DO CARTEIRO CHINÊS	35
<b>4</b>	<b>METODOLOGIA</b>	<b>39</b>
4.1	MÉTODO MATEMÁTICO	39
4.1.1	PROBLEMA DAS PONTES DE KÖNIGSBERG RESOLVIDO NO SOFTWARE LINDO	40
4.2	TECNOLOGIAS UTILIZADAS	41
<b>5</b>	<b>IMPLEMENTAÇÃO COMPUTACIONAL E ANÁLISE DOS RESULTADOS</b>	<b>43</b>
5.1	ETAPAS DA IMPLEMENTAÇÃO COMPUTACIONAL	43
5.2	ANÁLISE DOS RESULTADOS	44
<b>6</b>	<b>CONCLUSÃO</b>	<b>47</b>
6.1	LIMITAÇÕES	47
	<b>REFERÊNCIAS</b>	<b>49</b>
<b>A</b>	<b>APÊNDICE A: CÓDIGO DO MODELO MATEMÁTICO UTILIZANDO O SOFTWARE DEV-C++ PARA IMPLEMENTAR NO SOFTWARE LINDO</b>	<b>51</b>
<b>B</b>	<b>APÊNDICE B: PARTE DA MATRIZ DISTÂNCIA</b>	<b>56</b>
<b>C</b>	<b>APÊNDICE C: CÓDIGO DO MODELO MATEMÁTICO UTILIZANDO O SOFTWARE DEV-C++ PARA IMPLEMENTAR O MODELO NO SOFTWARE XPRESS</b>	<b>57</b>



## 1 INTRODUÇÃO

Os Correios tiveram sua origem no Brasil em 25 de janeiro de 1663 e, em 1969, sofreram nova configuração com a criação da Empresa Brasileira de Correios e Telégrafos (ECT), vinculada ao Ministério das Comunicações.

Nos últimos anos, um dos problemas que a ECT vem enfrentando é a realização de concurso público para o ingresso de novos funcionários. Com a falta de funcionários, as empresas de Correios no Brasil tiveram que diminuir um Distrito Postal do carteiro (DP), que significa determinar novas áreas de atendimento para a entrega de correspondências para cada carteiro, aumentando um trecho pra cada e dividindo em duas regiões, A e B, para que seja feita a entrega alternada e que a carga transportada seja aproximadamente a mesma para todos.

Com o aumento de volumes de objetos durante o dia, houve então a necessidade de melhorar o processo de entrega, que é composto, basicamente, da separação, ordenação e entrega de objetos, etapas estas que são realizadas pelos oito carteiros que trabalham na agência de correios na cidade de Bandeirantes, localizada no oeste do Paraná.

Neste trabalho, pretende-se otimizar as rotas de entrega de correspondências no município de Bandeirantes, utilizando o Problema do Carteiro Chinês, para minimizar a rota percorrida pelo carteiro. Foram utilizados conceitos básicos sobre a Teoria dos Grafos, que cria uma tendência para solucionar vários problemas nos quais a matemática aplicada é fundamental, justificando assim o seu estudo, embasando em referências teóricas e em suas aplicações, por meio da análise dos dados apresentados como resultado pelo softwares. Um problemas clássicos de otimização, dentro da Teoria de Grafos, são Problema do Caixeiro Viajante (PCV) e o Problema do Carteiro Chinês (PCC) [Goldbarg e Luna 2005]. Também foram utilizados dados de um bairro real, no qual atua um carteiro. No grafo que representa esse bairro, foi implementado no software DEV-C++ o modelo matemático do Problema do Carteiro Chinês Não Orientado (PCCND) e aplicado no software Xpress do qual foi possível extrair a solução ótima.

Alguns trabalhos relacionados à otimização dos serviços postais, principalmente aos que abordaram o Problema do Carteiro Chinês, podem ser citados: [Smiderle, Steiner e Wilhelm 2004] é apresentada uma metodologia para o problema de cobertura de arcos, para otimizar a rota percorrida para cada leiturista de forma a minimizar o tempo despendido com seu deslocamento na cidade de Pato Branco, Paraná. Foi utilizado uma combinação de técnicas da área da Pesquisa Operacional, como a heurística clássica de Teitz e Bart para a obtenção das 12-medianas necessárias para o problema, onde foram designados os pontos a cada mediana por meio do algoritmo de Gillett e Johnson. O algoritmo do Carteiro Chinês foi utilizado para obter o roteamento em cada grupo de atendimento. Os autores verificaram resultados satisfatórios, obtendo uma melhoria de 6,15%.

[Costa et al. 2001] propuseram um estudo para a otimização do serviço de correspondências realizado pela Empresa de Correios e Telégrafos (ECT) na cidade de Curitiba, Paraná. A resolução do problema foi feita em duas fases. Na primeira fase, foram utilizados algoritmos clássicos de solução do problema de roteamento capacitado de vários veículos e de vários depósitos. Na segunda fase, para determinar a sequência de pontos, foram utilizados os resultados da fase 1. Foram testados vários algoritmos clássicos de solução do problema do caixeiro viajante, comparando-se os seus desempenhos em função do tempo computacional e de distâncias mínimas.

[Gomes et al. 2009] apresentam um estudo geral sobre o Problema do Carteiro Chinês, nas versões simétricas, orientada e mista, do ponto de vista dos algoritmos exatos. Foram utilizadas as implementações exatas dos algoritmos de Sherafat, e de Edmonds e Johnson, adaptada de Burkard e Derigs.

## 1.1 OBJETIVOS

### 1.1.1 OBJETIVO GERAL

O objetivo deste trabalho é otimizar ou alcançar melhorias significativas, no roteiro de entrega de correspondências dos carteiros, na cidade de Bandeirantes, localizada no Estado do Paraná, utilizando o Problema do Carteiro Chinês.

### 1.1.2 OBJETIVOS ESPECÍFICOS

- a) Explorar o problema da rota de correspondências na cidade de Bandeirantes/PR, utilizando dados reais;
- b) Resolver o Problema do Carteiro Chinês para obter a rota de entrega das correspondências na cidade de Bandeirantes/PR;
- c) Solucionar o problema utilizando o software LINDO para análise dos resultados;
- d) Encontrar a melhor solução para o problema, ou seja, a solução ótima, com o intuito de minimizar o custo de operação e entrega das correspondências.

## 1.2 JUSTIFICATIVA

A dificuldade para se definir o trajeto a ser realizado pelos carteiros incide diretamente na rota que eles vão fazer durante o dia. Portanto, a definição de uma rota que minimize esse trajeto se faz fundamental para reduzir o custo operacional com esta função.

Nesse sentido, pensando na redução de custo, o presente trabalho visa minimizar a rota percorrida pelos carteiros na agência de Correios no município de Bandeirantes/PR. Otimizar essa atividade acarretaria em uma possível diminuição na fadiga dos carteiros, traria economia de combustível e, uma diminuição nos atrasos de correspondências. O trabalho torna-se oportuno ao introduzir e apresentar soluções de problemas de roteirização que podem ser utilizados para otimização dessa situação.

## 1.3 ORGANIZAÇÃO DO TRABALHO

O trabalho está estruturado em seis capítulos, incluindo esta introdução.

No segundo capítulo é realizado o estudo e a estruturação do problema: foram obtidos dados com a empresa de Correios da cidade de Bandeirantes, PR, através de um questionário entregue via e-mail para um funcionário da empresa.

O terceiro capítulo destina-se à fundamentação teórica.

O quarto capítulo detalha a metodologia utilizada para resolver o problema da entrega de correspondência, utilizando como estudo de caso a cidade de Bandeirantes.

O quinto capítulo é descritos as etapas da implementação computacional e a análise dos resultados.

O sexto capítulo são as conclusões do trabalho.

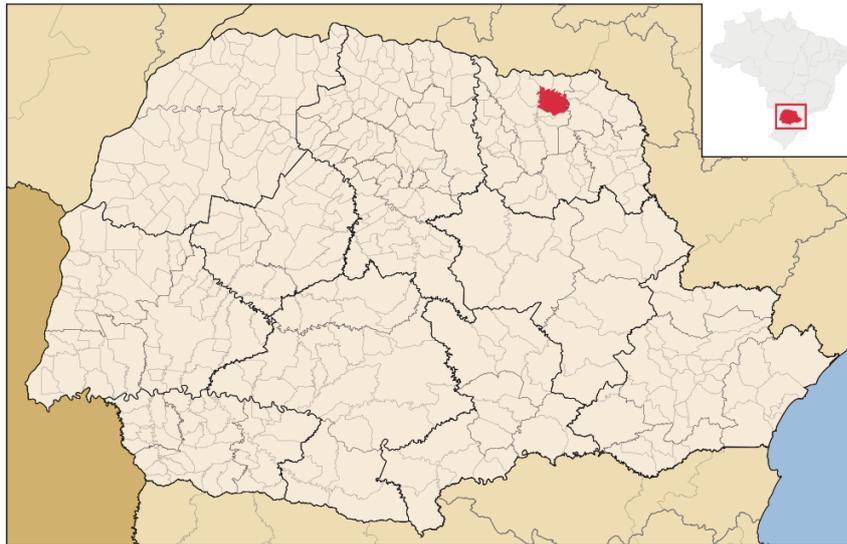
## 2 COLETA DE DADOS NO MUNICÍPIO DE BANDEIRANTES

Neste capítulo são descritos: a caracterização da cidade de Bandeirantes, a descrição do processo de entrega das correspondências e a descrição do problema a ser estudado.

### 2.1 CARACTERIZAÇÃO DO MUNICÍPIO DE BANDEIRANTES

A cidade de Bandeirantes/PR está localizada geograficamente a  $23^{\circ} 06' 36''$  de latitude Sul e a uma longitude  $50^{\circ} 27' 28''$  oeste, estando a uma altitude de 420 metros, conforme destaque na Figura 1. O Município de Bandeirantes possui uma área total de  $446,301 \text{ km}^2$ , representando 0,2246 por cento do estado, 0,0794 por cento da região e 0,0053 por cento de todo o território brasileiro.

**Figura 1 – Mapa do Paraná, Bandeirantes em Destaque**



Fonte: [Abreu 2006]

### 2.2 DESCRIÇÃO DO PROCESSO DE ENTREGA DAS CORRESPONDÊNCIAS

Com o objetivo de investigar o funcionamento do correio e o trabalho dos carteiros, na cidade de Bandeirantes, foi elaborado um questionário e entregue ao funcionário da empresa.

O entrevistado foi contatado e concordou em fazer parte deste trabalho. A entrevista foi realizada via e-mail, visto que o entrevistado não se importou. Além de responder ao questionário, este, por sua vez, complementou as questões, procurando acrescentar algum comentário que julgasse importante. Após a coleta dos dados, todas as informações foram analisadas e reescritas.

Para efetuar a análise dos dados, a resposta do entrevistado foi agrupada por categorias. No primeiro momento, foi descrito o processo de divisão das correspondências, se por região, por exemplo. No segundo momento, como são feitas as entregas dos objetos (carro, moto ou a pé) e no terceiro momento, o funcionamento das rotas.

Na Empresa Brasileira de Correios, cada unidade tem suas particularidades. Na cidade de Bandeirantes, por exemplo, existem seis distritos, ou seja, a cidade é dividida em cinco regiões e um distrito especial que é atendido com o carro. O carro roda o dia todo e atende pontos da cidade toda, sendo assim parte do correio quatro vezes ao dia. As correspondências que ficam

no carro são encomendas do tipo SEDEX, PAC e MALOTES. São atendidos três distritos de moto e dois distritos de bicicleta e o outro a pé. Os de moto, são os bairros mais afastados da cidade e, na região mais central (por exemplo, os comércios, escritórios, etc.) os carteiros pedestres.

Em Bandeirantes, com seus 31.526 habitantes [IBGE] chegam até ao correio aproximadamente 3.000 cartas por dia. Destas 3.000 cartas, são objetos tais como: correspondências, encomendas, folhetos de propaganda, malotes, entre outros.

As correspondências chegam todas misturadas da cidade toda. E então os carteiros fazem a separação, que funciona da seguinte forma: primeiramente separam por distritos, por exemplo, cada bairro tem o número do distrito a qual pertence. Após a separação por distrito, que é chamado de Triagem, é feita a separação por logradouro, que é a separação por rua. Cada carteiro pega sua caixeta com as correspondências do distrito e começa a separar por rua. Existe uma mesa com várias posições. Cada uma com o nome de uma rua. Cada carteiro começa a ordenar sua correspondência conforme a ordem de entrega, por exemplo, se o carteiro fizer a rua sentido centro-bairro, é de menor para o maior a numeração. Caso contrário, do maior para o menor e assim, sucessivamente.

Esse serviço é realizado de segunda-feira a sexta-feira, sendo que a região A é atendida em um dia e a região B no outro dia, como mostraremos mais adiante. Na Tabela 1, pode-se observar como é feita a divisão das regiões da cidade que pertencem a cada distrito.

**Tabela 1 – Divisões de cada carteiro**

DIVISÕES DAS REGIÕES DE CADA CARTEIRO		
CARTEIRO	DISTRITO	BAIRROS
MOTO 1 e BICICLETA 1	365	Ana Rosa, Novo Paraíso, Habital Brasil, San Rafael, IBC, Vila Rubi, Itapeva e Maria Alice.
MOTO 2	361	Vila Maria, Moretti, São Vicente, Jardim Yara, Pombal 1 e 2, São Geraldo e Jardim das Torres.
MOTO 3	363	Invernada, Lordani, Jardim União, Bela Vista, Yukith Matida, Carvalho Henrique, Celso Fontes e Primavera.
BICICLETA 2	362	Avenida Bandeirantes, Vila União, Centro até Azarias, Clube Guaíra.
PEDESTRE 1	364	IBC da Francisca Morilha, Luiz Meneghel, José Mário Junqueira, etc.

**Fonte: Autoria Própria**

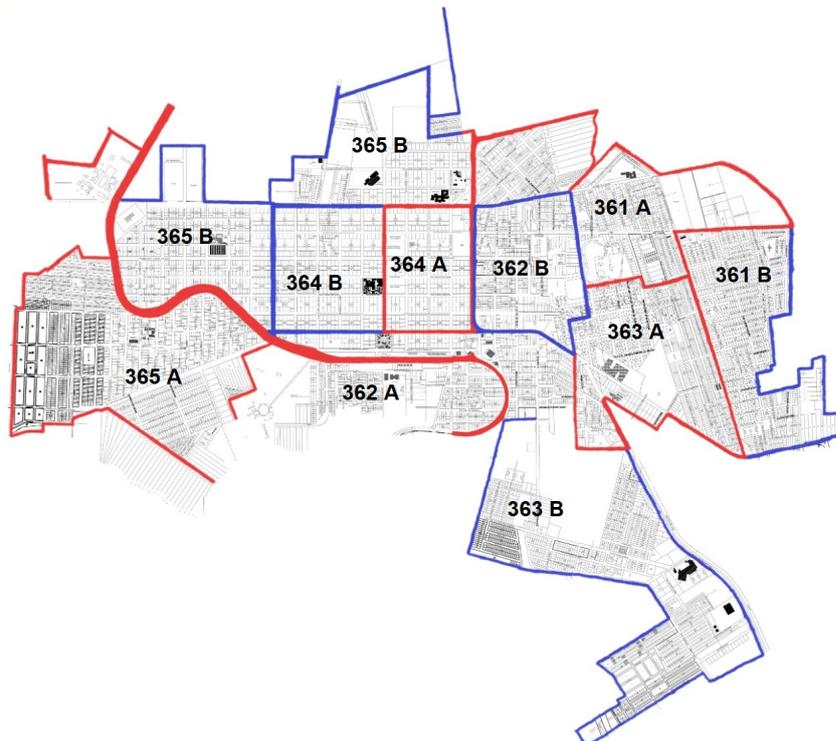
Para fazer esta divisão, foi feita uma contagem de cartas e foram medidas as ruas. Este processo é feito a cada três anos mais ou menos. Ou seja, eles contam o tempo de uma carta

registrada que precisa ser assinada e que leva em média três minutos mais a percorrida da rua. Outro ponto importante, seria o tempo gasto de cada carteiro para percorrer um trecho de rua e para entregar as correspondências, quantidades de objetos a serem entregues e modo de fazer cada trajeto (modo em U - pode percorrer um lado do trajeto, fazer o contorno no final do mesmo e retornar pelo outro lado; modo em Z - pode percorrer o trajeto cruzando a rua de um lado para o outro ou modo em L - pode percorrer apenas um lado da rua, pois o outro lado não há entregas).

### 2.3 DESCRIÇÃO DO PROBLEMA A SER ESTUDADO

Atualmente, a região do município de Bandeirantes está dividida em cinco distritos, sendo que cada distrito é atendido por cada carteiro (moto, bicicleta, pedestre) e um distrito atendido com o carro, que faz a cidade toda. Com a falta de funcionários, as empresas de Correio tiveram que dividir todos os distritos em duas regiões, lado A e lado B. No lado A tracejado em vermelho é feita a entrega de correspondência em um dia e, o lado B tracejado em azul, no outro dia, passando sempre um dia sim e um dia não, conforme mostrado na Figura 2.

**Figura 2 – Mapa de Bandeirantes. Divisões de dois lados A e B**



**Fonte: Disponibilizado pelo funcionário da Empresa de Correio.**

Nesse sentido, cada carteiro ficou responsável por cada bairro da região para a entrega de correspondência, como mencionado na Tabela 1. Os distritos 362 e 363 são as regiões que tem mão única, já os distritos 361, 364 e 365, são as regiões de mão dupla.

Para o desenvolvimento deste trabalho, será implementado o algoritmo do PCC apenas em um distrito, neste caso, o distrito 364-B (mão dupla). Vale ressaltar que deve-se levar em consideração, variáveis como o tempo de duração das entregas, a quantidade de quilômetros percorridos, o número de viagens necessárias para que a entrega das correspondências seja

finalizada com sucesso e, diversas condições climáticas como intenso calor, frio ou chuva. Neste trabalho, será levado em consideração somente a distância que o carteiro percorre fazendo cada rota.

### 3 FUNDAMENTAÇÃO TEÓRICA

Neste capítulo, serão formalizados conceitos básicos de grafos e do Problema do Carteiro Chinês (PCC) e, suas principais características, através de algumas definições e exemplos. Tais conceitos serão importantes para o nosso trabalho nos capítulos seguintes.

#### 3.1 GRAFOS

A teoria de grafos é um assunto antigo com muitas aplicações modernas e, estuda a relação entre os elementos de um determinado conjunto e nos permite modelar vários problemas em Matemática Aplicada, Engenharia, Física, Química, dentre outros. Podemos citar alguns problemas famosos tais como O Problema do Carteiro Chinês (PCC) e o Problema do Caixeiro Viajante (PCV). O PCC consiste em encontrar a menor distância a ser percorrida tendo como restrição percorrer todas as arestas do grafo. Já o PCV consiste em encontrar um circuito que possua a menor distância, começando por uma cidade qualquer e a ela retornando visitando cada cidade apenas uma vez.

As ideias básicas de grafos foram introduzidas no século XVIII, mais precisamente no ano de 1736, pelo famoso matemático suíço Leonhard Euler (1707–1783). Euler iniciou seus estudos em grafos discutindo um enigma, hoje conhecido como O Problema das Pontes de Königsberg, o qual ele resolveu e determinou um método geral para problemas do mesmo tipo.

Podemos identificar grafos como conjuntos fundamentais, os quais se mantêm uma relação entre si, tais como na representação de rotas de transporte (um mapa de estradas, por exemplo), rede de comunicação (como em uma rede de computadores) ou rotas de distribuição de produtos ou serviços (como dutos de água ou gás, etc.).

Grande parte das definições e termos relacionados à teoria de grafos usada neste trabalho é clássico e pode ser encontrado em vários trabalhos de roteamento de arcos, como por exemplo, na pesquisa de [KONOWALENKO] e [Arenales et al. 2015].

Usualmente, é utilizada a notação:

- **G** para grafo, ou outra letra qualquer, desde que seja maiúscula.
- **V** ou **N** para o conjunto dos elementos, dos vértices ou nós.
- **A** ou **E** para o conjunto dos arcos ou arestas que indica a relação entre os vértices ou nós.

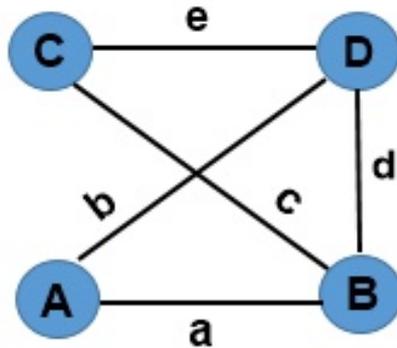
Ressaltamos que neste trabalho, serão adotadas as notações:  $G = (N, E)$ , ou simplesmente  $G$ , para descrever o grafo correspondente ao conjunto de vértices e arestas. E também, vamos supor que o número de elementos fundamentais é finito, de modo que podemos enumerá-los.

**Definição 1 (grafo e rede):** *Seja  $N$  um conjunto finito, cujos elementos são chamados nós (ou vértices) e  $E$  um conjunto de pares de nós, cujos elementos são chamado de arestas. O par  $G = (N, E)$  é chamado grafo. Uma rede é um grafo cujo nós e/ou arestas têm valores associados.*

**Observação 1** *Os termos grafos e redes são usados como sinônimos, então não fazemos distinção entre os dois.*

**Exemplo 1** *Considere um conjunto de nós  $N = \{A, B, C, D\}$  e o conjunto de arestas  $E = \{(A, B), (A, D), (D, C), (B, C), (B, D)\}$ . A representação gráfica do grafo  $G = (N, E)$  é ilustrada na Figura 3, abaixo.*

Figura 3 – Exemplo de representação de um grafo.



Fonte: Autoria Própria.

**Definição 2 (grau de um vértice)** É o número de vezes em que as arestas incidem sobre um determinado vértice. Na Tabela 2, por exemplo, temos as seguintes arestas incidentes do vértice (A, B, C e D), dado na Tabela 2 abaixo.

Tabela 2 – Grau de cada vértice.

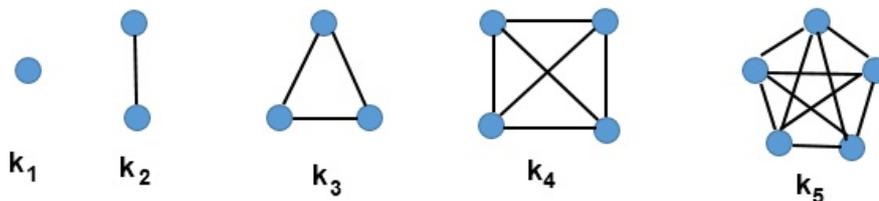
Vértice	Grau do Vértice
A	2
B	3
C	2
D	3

Fonte: Autoria Própria.

**Definição 3 Grafo Completo Simples** Um grafo é dito ser completo quando há uma aresta entre cada par de seus vértices. Estes grafos são designados por  $K_n$ , onde  $n$  é a ordem do grafo. Um grafo  $K_n$  possui o número máximo possível de arestas para um dado  $n$ . Ele é, também regular  $(n-1)$  pois todos os seus vértices tem grau  $n-1$ .

Na figura 4 são apresentados exemplos de grafo completo simples.

Figura 4 – Exemplo de grafos simples.

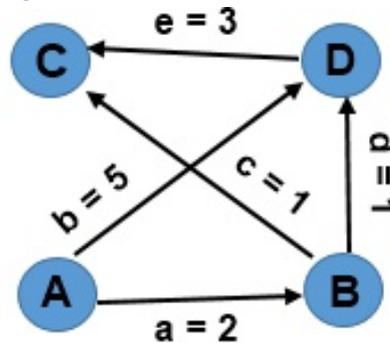


Fonte: Autoria Própria

**Definição 4 (grafo orientado e rede orientada):** Um grafo  $G = (N, E)$  no qual as arestas são pares ordenados (subconjuntos de  $N \times N$ ) é chamado grafo orientado ou dígrafo. Neste caso, o par ordenado  $(i, j)$  é chamado arco, e  $i$  é o nó inicial e  $j$  o nó final. Uma rede orientada é um grafo orientado cujos nós e/ou arcos têm valores associados. Um grafo orientado é representado graficamente de forma análoga ao grafo, porém, o arco  $(i, j)$  é representado por uma flecha de  $i$  para  $j$ , indicando a orientação relevante.

**Exemplo 2** Considere um conjunto de nós  $N = \{A, B, C, D\}$  e o conjunto de arcos  $E = \{(A, B), (A, D), (D, C), (B, C), (B, D)\}$  e os valores associados a cada arco,  $a = 2, b = 5, c = 1, d = 1$  e  $e = 3$ . A representação gráfica da rede orientada é ilustrada na Figura 5, abaixo.

Figura 5 – Exemplo de representação gráfica de uma rede orientada.



Fonte: Autoria Própria.

**Definição 5 (caminho):** Um caminho de um nó  $i_0$  a um nó  $i_k$  é uma sequência de arcos  $C = (i_0, i_1), (i_1, i_2), (i_2, i_3), \dots, (i_{k-1}, i_k)$  no qual o nó inicial de cada arco é o nó final do arco imediatamente anterior da sequência e  $i_0, i_1, i_2, i_3, \dots, i_k$  são todos nós distintos. Portanto, todos os arcos em um caminho estão direcionados coerentemente, apontando do nó  $i_0$  ao nó  $i_k$ .

**Exemplo 3** No grafo da Figura 5, um caminho possível do nó A ao nó C, seria:  $\{(A, B), (B, D), (D, C)\}$ .

**Definição 6 (cadeia):** Uma cadeia é uma estrutura similar à de um caminho, exceto que os arcos não precisam estar coerentemente orientados, ou seja, uma cadeia é uma sequência de arcos de modo que cada arco tem exatamente um nó em comum com o arco imediatamente anterior na sequência.

**Observação 2** : Todo caminho é uma cadeia, mas nem toda cadeia é um caminho.

**Exemplo 4** No grafo da Figura 5, uma cadeia do nó A ao nó D, poderia ser:  $\{(A, B), (B, C), (D, C)\}$ .

**Definição 7 (circuito, ciclo):** Um circuito é um caminho fechado, ou seja, é um caminho de um nó  $i_0$ , a um nó  $i_k$ , em que  $i_k = i_0$ . O correspondente ao circuito, no caso da cadeia, é denominado ciclo, ou seja, o ciclo é uma cadeia fechada (um nó, inicial ou final, do primeiro arco da sequência coincide com um nó, inicial ou final, do último arco da sequência).

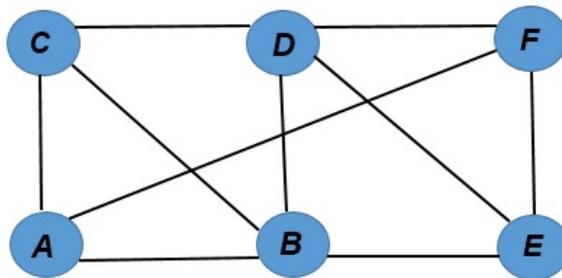
**Exemplo 5** Na Figura 5, não há nenhum circuito, mas apenas ciclos. Por exemplo:  $\{(B, C), (D, C), (B, D)\}$  e  $\{(A, D), (B, D), (A, B)\}$ .

Um caminho ou um circuito é dito **euleriano** se ele contém todas as arestas de um grafo. Um grafo que contém um circuito euleriano é um **grafo euleriano**.

Um circuito que passa por todos os vértices de um grafo, sem que repita o mesmo vértice mais de um vez, é denominado **circuito hamiltoniano**. Um grafo que contém um circuito hamiltoniano é um **grafo hamiltoniano**.

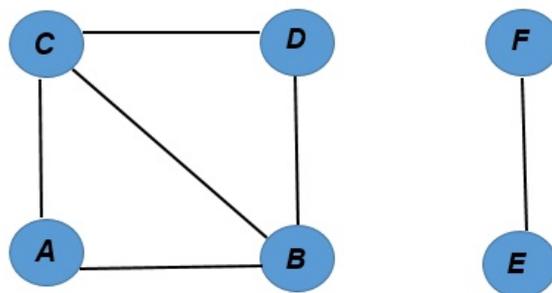
**Definição 8 (grafo conexo e grafo desconexo):** Um grafo  $G = (N, E)$  é conexo se existir um caminho entre qualquer par de vértices. Caso contrário, dizemos que o grafo é desconexo.

Figura 6 – Grafo Conexos.



Fonte: Autoria Própria.

Figura 7 – Grafo Desconexo.



Fonte: Autoria Própria.

**Observação 3 :** No exemplo da Figura 6 e 7, o grafo é não orientado, então não importa o sentido que se percorre as arestas. Mas se o grafo for orientado, precisa obedecer o sentido.

**Definição 9 (grafo fraco ou fortemente conexo):** Um grafo é dito fracamente conectado (ou simplesmente conexo) se existe pelos menos uma cadeia entre quaisquer dois de seus nós, e fortemente conexo se existe pelo menos um caminho de cada nó a todos os demais nós do grafo.

Note que o grafo da Figura 5 não é fortemente conectado pois não há caminho do nó B para o nó A, por exemplo.

**Definição 10 (árvore, subgrafo e árvore geradora de um grafo):** Uma árvore é um grafo conectado sem ciclos. Diz-se que um grafo  $G' = (N', E')$  é um subgrafo de  $G = (N, E)$  se  $N' \subseteq N$

e  $E' \subseteq E$ , com a condição de que, se  $(i, j) \in E'$ , então  $i$  e  $j$  também devem pertencer a  $N'$ . Uma árvore geradora de um grafo  $G$  é um subgrafo de  $G$  que é uma árvore e inclui todos os nós do grafo  $G$ .

Veja a seguir, algumas propriedades úteis para a árvore geradora.

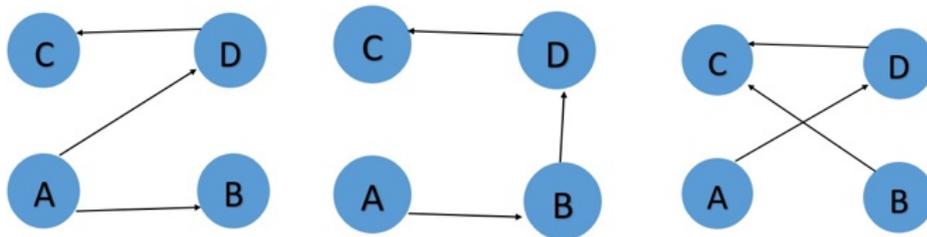
**Proposição 1 (árvore geradora):** Considere um grafo  $G = (N, E)$ , com  $|N| = n$  (isto é,  $G$  tem  $n$  nós), e um subgrafo  $G' = (N, E')$  de  $G$ . As seguintes afirmações são equivalentes:

- i)  $G' = (N, E')$  é uma árvore geradora de  $G$ .
- ii)  $|E'| = n-1$  (isto é,  $G'$  tem  $n-1$  arestas) e  $G'$  é conexo.
- iii)  $|E'| = n-1$  e  $G'$  não tem ciclos.

Em outras palavras, toda árvore geradora de um grafo com  $n$  nós tem  $n-1$  arestas e basta que seja um subgrafo conexo ou sem ciclo.

**Exemplo 6** Na Figura 8 são apresentadas algumas árvores geradoras do grafo da Figura 5, e é possível verificar a validade das propriedades de árvores geradoras.

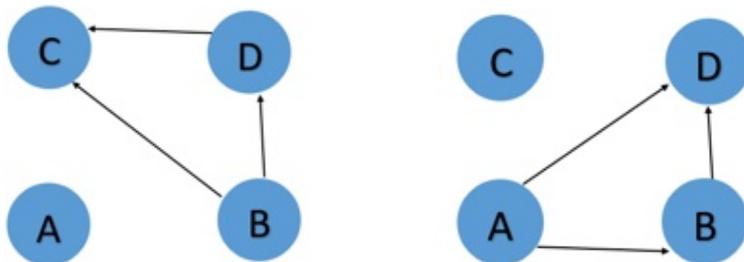
Figura 8 – Exemplos de árvore geradora do grafo da Figura 4.



Fonte: Autoria Própria.

Para evidenciar as propriedades de árvores geradoras, na Figura 9, apresentamos alguns subgrafos do grafo da Figura 5, com  $n-1 = 3$  arestas, mas que não são árvores.

Figura 9 – Exemplo de subgrafos com três arcos que não são árvore geradora do grafo da Figura 4.



Fonte: Autoria Própria.

### 3.1.1 REPRESENTAÇÃO MATRICIAL DE GRAFOS

Existem várias formas de se organizar os dados de um grafo, de modo que eles possam ser introduzidos em um computador. Uma delas seria a forma matricial. Assim, neste trabalho por exemplo, vamos considerar a Matriz de Adjacência, que consiste em uma matriz que representa os dados de um grafo.

**Definição 11 (matriz de adjacência):** Dado um grafo  $G = (N, E)$ , a matriz de adjacência  $M$  é uma matriz de ordem  $n \times n$ , tal que:

$n$  = número de arestas

$M[i,j]$  é o número de aresta de  $i$  a  $j$

**Exemplo 7** Os grafos do Exemplo 1 (não orientado) e do Exemplo 2 (orientado) são representados, respectivamente, pelas seguintes matrizes de adjacências:

$$M_1 = \begin{bmatrix} 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 \\ 1 & 1 & 1 & 0 \end{bmatrix}, M_2 = \begin{bmatrix} 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}$$

A matriz de adjacências de grafos não orientados é sempre simétrica ( $M = M^T$ ). No caso de uma rede, com o valor  $c_{ij} \neq 0$  associado a aresta  $(i, j)$ , podemos colocar essa informação em  $M$ , fazendo  $m_{ij} = c_{ij}$  para todo  $(i, j) \in E$  e  $m_{ij} = 0$  em caso contrário.

### 3.2 PROBLEMAS DE ROTEAMENTO DE ARCOS

Segundo [Goldberg e Luna 2005], os problemas de roteamento, em geral, podem ser classificados em duas grandes classes: Roteamento em Grafos e Roteamento de Veículos propriamente dito.

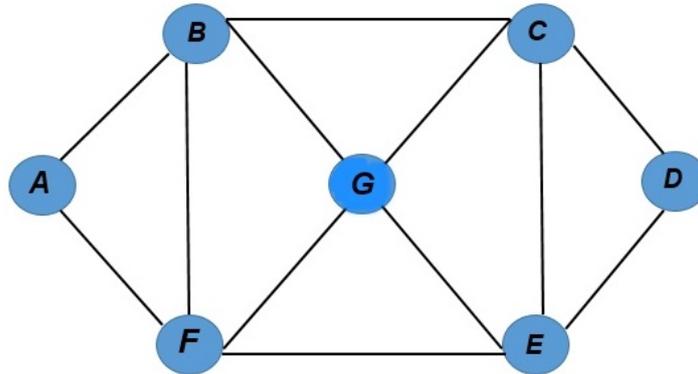
A classe geral dos problemas de Roteamento de Grafos é ainda constituída pelas subclasses: Problema de Roteamento de Nós (associados aos ciclos Hamiltonianos) e Problemas de Roteamento de Arcos (associados aos ciclos Eulerianos).

Segundo [Smiderle, Steiner e Wilhelm 2004], os problemas de cobertura de arcos determinam um caminho mínimo através de uma rede tal que todos os arcos sejam atravessados uma única vez. Este problema, conhecido na literatura como o Problema do Carteiro Chinês (PCC), tem muitas aplicações como, por exemplo, problemas do setor público incluindo varredura de ruas, coleta de lixo, roteamento de carteiros, dentre outros.

### 3.3 GRAFOS DE EULER E O PROBLEMA DAS PONTES DE KÖNIGSBERG

Como mencionado na seção 3.1, um circuito é dito euleriano se ele contém todas as arestas de um grafo. Um grafo que contém um circuito euleriano é um grafo euleriano. O grafo da Figura 10, por exemplo, é um grafo euleriano, pois possui o seguinte circuito Euleriano:  $T = (A, B, C, D, E, C, G, F, E, G, B, F, A)$ .

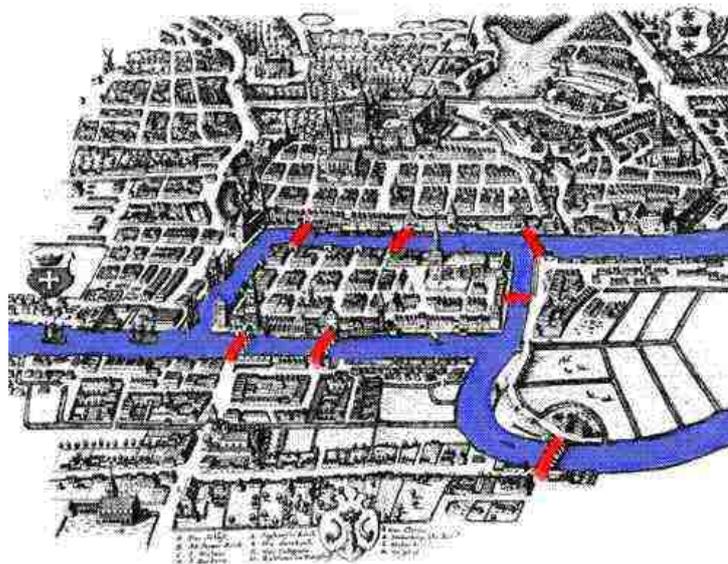
Figura 10 – Grafo Euleriano.



Fonte: Autoria Própria.

Euler foi o primeiro matemático a escrever um documento sobre a teoria de grafos. Ele iniciou seus estudos, sobre tal teoria, estudando e tentando resolver um problema conhecido como "O Problema das 7 Pontes de Königsberg". Tal problema consiste em saber se um indivíduo pode, a partir de um determinado ponto, passar em cada uma das sete pontes exatamente uma vez e voltar ao ponto de origem. A Figura 11, mostra a cidade de Königsberg, mostrando o rio Pregel e as sete pontes.

Figura 11 – A cidade de Königsberg.

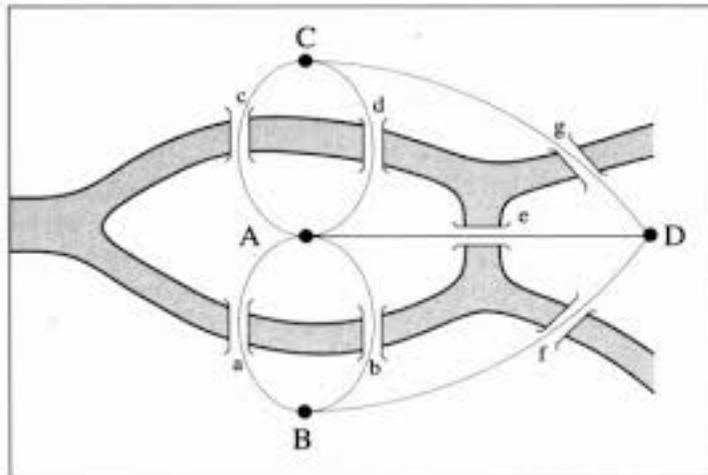


Fonte: Disponível em: <<http://www.engeene.it/algorithmi-estreet-view-i-7-ponti-di-kaliningrad/>>.

Euler foi desafiado a realizar um passeio pelas sete pontes da cidade, mas o detalhe deste passeio é que ele deveria passar uma única vez por cada ponte e retornar ao seu ponto de partida sem passar pela mesma ponte mais de uma vez, onde provou que tal trajeto não era possível. Ele usou uma representação gráfica bem simples para desenhar a situação - associou as pontes a linhas e as regiões de terra a pontos, criando possivelmente o primeiro grafo da história. Este problema pôde ser modelado (e resolvido) utilizando-se da teoria dos grafos. Uma

representação gráfica da região pode ser expressa pelo que chamamos de representação gráfica de um grafo e está expressa na Figura 12, onde os vértices A, B, C, D são as margens e as ilhas, e as arestas correspondem as pontes.

Figura 12 – Grafo que representa a cidade de Königsberg.



Pode-se observar na Figura 12, que para este problema ter solução é preciso traçar um circuito euleriano, ou seja, um circuito contendo todas as arestas.

O Problema das Pontes de Königsberg inspirou o estudo do problema do carteiro chinês, em que o objetivo é determinar um caminho de comprimento mínimo cobrindo cada arco ao menos uma vez. O problema foi relatado de forma simplificada por [Guan 1962]: Um carteiro tem de cobrir sua rota e depois retornar ao Posto de Correio. O problema é encontrar a menor distância a ser percorrida pelo carteiro.

### 3.4 CIRCUITOS EULERIANOS

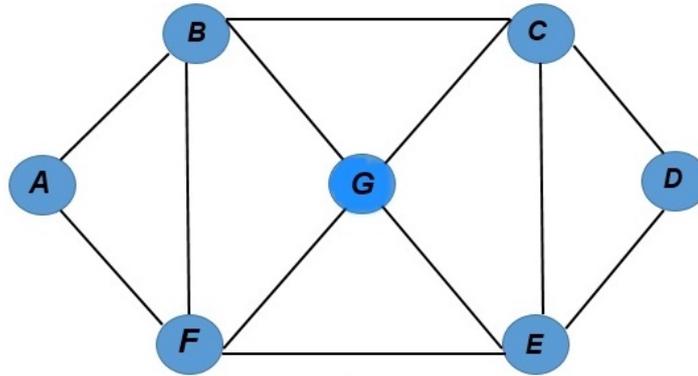
Supondo um grafo não orientado  $G(N, E)$ , fortemente conexo, um circuito que contém todas as arestas do grafo sem que repita a mesma aresta mais de uma vez, é denominado de Circuito Euleriano. E como foi analisado por [Sherafat et al. 2004], nem todo grafo contém um circuito euleriano; quando possui, ele é chamado de grafo euleriano. O teorema básico sobre a existência de um circuito euleriano em um grafo não orientado, é o seguinte:

**Teorema 3.4.1** : *Um grafo fortemente conexo  $G(N, E)$  contém um circuito euleriano, se, e somente se, o grafo não tem nenhum nó de grau ímpar.*

**Demonstração 1** *Suponha que o grafo seja euleriano (Figura 13). Então  $G$  possui um circuito (passeio fechado) euleriano (Figura 14). Se contarmos para cada nó, a entrada e saída dele, ao final de todo percurso teremos um conjunto de números pares. Suponha agora que temos um grafo  $G$  onde todos os seus nós têm grau par.*

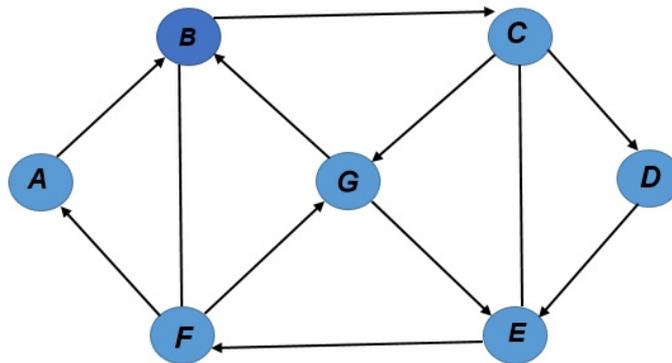
*Escolha um nó  $i$  qualquer e comece a percorrê-lo sem repetir arestas, até não existirem arestas a serem percorridas, a partir do vértice corrente. Como todos os nós têm grau par então o último nó alcançado é o nó  $i$ . Se o circuito  $C$  contiver todas as arestas de  $G$  então a demonstração está concluída, caso contrário existirão arestas não percorridas.*

Figura 13 – Grafo Euleriano.



Fonte: [Prestes 2016]

Figura 14 – Circuito Euleriano.



Fonte: [Prestes 2016]

Como o grafo é conexo então existe um caminho entre qualquer par de vértices. Logo, existe algum caminho entre algum vértice do circuito até uma aresta  $q$  não incluída em  $C$ . Imagine que este caminho seja formado pela aresta  $(j, k)$ , onde  $j$  pertence ao circuito  $C$  e  $k$  pertence a aresta  $q$ . Se isto ocorrer deve-se percorrer o grafo a partir de  $j$  visitando todas as novas arestas sem acessar nenhuma aresta em  $C$ . Este novo circuito  $C'$  pode ser unido ao circuito  $C$  formando um único circuito.

Agora basta percorrer  $C$  a partir de  $j$  e quando retornar a  $j$  começar a percorrer  $C'$ . Repete-se este processo até que todas as arestas tenham sido visitadas. No final teremos um único circuito formado pela união de vários circuitos. O circuito resultante é chamado euleriano, assim como o grafo.

### 3.5 PROBLEMA DO CARTEIRO CHINÊS

Este problema foi elaborado pela primeira vez em 1962 por Mei-Ku Kwan, um matemático chinês. Ele aborda os grafos eulerianos com arestas valorizadas, ou seja, as arestas possuem valores que podem ser distância, tempo de percurso, custo, etc.

Agora, suponhamos que temos um bairro em que o carteiro deverá entregar as cartas por todas as ruas que começa e termina no ponto de distribuição. O problema é de identificar essa rota de maneira a minimizar a distância total percorrida. Essa situação pode ser representada por um grafo, onde as arestas correspondem as ruas e os vértices correspondem ao cruzamento das ruas. Caso a descrição das ruas apresentar um grafo euleriano, é óbvio que o carteiro deverá percorrer as ruas passando pelo circuito euleriano, caso o grafo não seja euleriano, poderemos acrescentar ruas (arestas ao problema de modo a minimizar o esforço do carteiro), ou até repetir ruas sempre descrevendo um circuito euleriano.

Segundo [Goldbarg e Luna 2005], um dos mais antigos problemas da teoria de grafos é o da determinação de um percurso sobre um grafo  $G$  que contenha toda aresta de  $G$  exatamente uma vez (Karp, 1975). Tal circuito é denominado de Euleriano, pelo fato de Euler ter sido o primeiro a reportar um estudo sobre a sua determinação, no ano de 1736.

O PCC é um problema de otimização que objetiva cobrir com um percurso (ou *tour*) todos os arcos do grafo, minimizando a distância total percorrida. O percurso do carteiro distingue-se do circuito (ou ciclo) Euleriano por nele ser permitida, se necessário, a repetição de arestas. Claramente no caso de o grafo possuir circuitos Eulerianos, tais circuitos solucionam o problema. O PCC é um exemplo de um problema de roteamento que admite solução em tempo polinomial ([Edmonds e Johnson 1973]).

[Edmonds e Johnson 1973] apresentam um interessante algoritmo para a solução do PCC via matching (emparelhamento). Pode-se resumir o algoritmo da seguinte forma [Goldbarg e Luna 2005]:

INÍCIO

**Ler** o grafo  $G(N, E)$

**Se** todos os nós em  $G$ , o grafo original, possuem grau par então **determinar** um ciclo Euleriano em  $G$  e Fim.

**Organizar** um grafo  $K_n$  da seguinte forma:

**Reunir** todos os vértices de grau ímpar no grafo  $K_n$  e **associar** a cada par de vértices  $i$  e  $j$  no grafo, uma aresta  $(i, j)$  com peso igual ao caminho mais curto que liga  $i$  a  $j$  no grafo  $G$ .

**Determinar** o 1-matching mínimo em  $K_n, M^*$ .

**Para cada aresta** pertencente a  $M^*$  **associar** uma nova aresta em  $G$  no caminho mínimo que ela representa, obtendo um grafo  $G_n$ .

**Determinar** a solução do carteiro chinês que é representada por um ciclo Euleriano em  $G_n$ .

FIM

Desde sua aparição na moderna literatura, o problema do carteiro chinês vem ganhando muita atenção de pesquisadores que tratam de logística, principalmente ligados à logística urbana.

O PCC, de acordo com a natureza da rede, classifica-se em:

- **Problema do Carteiro Chinês Não Direcionado (PCCND):** deseja-se gerar um percurso de custo mínimo sobre um grafo  $G = (N, E)$ , valorado e conexo, a partir de um vértice inicial. Exemplo: Cidades somente com ruas de sentido duplo.
- **Problema do Carteiro Chinês Direcionado (PCCD):** deseja-se gerar um percurso de custo mínimo sobre um grafo  $G = (N, A)$ , valorado e fortemente conexo (f-conexo), a partir de um vértice inicial. Exemplo: Cidades somente com ruas de mão única.
- **Problema do Carteiro Chinês Misto (PCCM):** deseja gerar um percurso de custo mínimo sobre um grafo  $G = (N, E, A)$ , valorado e fortemente conexo (f-conexo), a partir de um vértice  $v_0 \in N$ , origem. Exemplo: Cidades com ruas de mão dupla e mão única.

Basicamente, os problemas resumem-se em transformar o grafo original em um grafo euleriano para cada uma das versões do carteiro chinês.



## 4 METODOLOGIA

Neste capítulo são descritos os métodos aplicados na resolução do problema, procedimentos usados na coleta dos dados e caracterização da pesquisa.

### 4.1 MÉTODO MATEMÁTICO

A cidade de Bandeirantes possui ruas de sentido duplo e outras de sentido único. E, neste trabalho será considerado apenas as ruas de sentido duplo. Assim, será utilizado o PCCND para determinar a rota mínima da entrega de correspondência.

Segundo [Goldbarg e Luna 2005], a solução do problema do carteiro chinês em grafos não orientados e eulerianos reduz-se á determinação do circuito euleriano. Ainda relatam que Kwan Mei-Ko foi o primeiro a relatar PCCND em uma publicação datada de 1962 (MEI-KO, 1962) no periódico "Chinese Mathematics". Em virtude de sua origem, o problema ficou denominado como Problema do Carteiro Chinês.

A formulação matemática do PCCND, segundo [Bodin 1983], é apresentada a seguir:

Minimizar

$$\sum_{i=1}^n \sum_{j=1}^n c_{ij} x_{ij} \quad (4.1)$$

Sujeito a:

$$\sum_{j=1}^n x_{ij} - \sum_{j=1}^n x_{ji} = 0, \quad i = 1, \dots, n \quad (4.2)$$

$$x_{ij} + x_{ji} \geq 1, \quad \forall (i, j) \in A \quad (4.3)$$

$$x_{ij} \geq 0 \text{ e inteiras} \quad (4.4)$$

onde:

$x_{ij}$  = número de vezes em que a aresta  $(i, j)$  é percorrido de  $i$  para  $j$ ;

$c_{ij}$  = comprimento ou o custo da aresta  $(i, j)$ .

No modelo matemático proposto, a função objetivo (4.1) minimiza o custo total, ou seja, no caso do trabalho em estudo, a distância total a ser percorrida. As restrições em (4.2) garantem a continuidade da rota, as restrições em (4.3) que nenhuma aresta deixará de ser considerada e, em (4.4), tem-se que as variáveis do problema são não negativas, inteiras.

Note também que nas restrições em (4.3) é permitido percorrer as arestas mais de uma vez. Essa condição é necessária pois, caso o grafo não seja Euleriano, algumas arestas são repetidas no grafo, fazendo com que todas tenham grau par, para que assim se possa obter a solução.

Agora, vamos aplicar o modelo matemático apresentado acima, no Problema das Pontes de Königsberg, utilizando o grafo da Figura 12.

#### 4.1.1 PROBLEMA DAS PONTES DE KÖNIGSBERG RESOLVIDO NO SOFTWARE LINDO

A Figura 12, do Capítulo 3, exemplifica um grafo por meio de um diagrama. Os pontos (A, B, C e D) representam os vértices, os arcos que fazem a ligação de um ponto para o outro representam as arestas, e as letras (a, b, c, d, e, f e g) representam as distâncias entre esses pontos.

Consideremos então, que o custo de cada distância seja unitária, ou seja, custo igual a 1 (um) e considere ainda que os vértices A, B, C e D sejam respectivamente 0, 1, 2 e 3. Assim, utilizando o software *Excel* podemos formar a matriz distância para o problema em questão, como mostrado na Tabela 3.

**Tabela 3 – Matriz Distância.**

d(i,j)	0	1	2	3
0	0	1	1	1
1	1	0	0	1
2	1	0	0	1
3	1	1	1	0

**Fonte: Autoria Própria.**

Note que não existe nenhuma aresta ligando os vértices 1 e 2, portando as posições (1,2) e (2,1) da matriz recebem o valor zero. Depois de concluída a construção da matriz distância, foi criado um programa no DEV-C++ para imprimir o modelo matemático do PCCND, como mostra a Figura 15. O algoritmo do PCCND desenvolvido no DEV-C++ se encontra no Apêndice A.

Observa-se que no problema das pontes de Königsberg, há duas pontes ligando os vértices A e B e duas pontes ligando o vértices A e C. Assim, é necessário fazer algumas modificações das restrições em (4.3) no modelo utilizado, ou seja, na Figura 14 por exemplo, a restrição 2 indica que  $x_{01} + x_{10} \geq 2$  (pontes entre A e B) e  $x_{02} + x_{20} \geq 2$  (pontes entre A e C).

**Figura 15 – Modelo matemático do PCCND gerado com código no DEV-C++ implementado no LINDO**

```

out_exemplodaspontes_LINDO - Bloco de notas
Arquivo Editar Formatar Exibir Ajuda
!-----;
!# Escreve o modelo matemático de Bodin para ser resolvido no LINDO.;
!# Usa a matriz de adjacência do problema que está no arquivo exemplo.txt;
!-----;

! Função objetivo: minimizar o trajeto percorrido;
MIN 1 X01 + 1 X02 + 1 X03 + 1 X10 + 1 X13 + 1 X20 + 1 X23 + 1 X30 + 1 X31 + 1 X32

SUBJECT TO

! Restricao 1: tudo que entra precisa sair;
X10 - X01 + X20 - X02 + X30 - X03 = 0
X01 - X10 + X31 - X13 = 0
X02 - X20 + X32 - X23 = 0
X03 - X30 + X13 - X31 + X23 - X32 = 0

! Restricao 2: percorre todas as arestas ao menos uma vez;
X01 + X10 >= 2
X02 + X20 >= 2
X03 + X30 >= 1
X13 + X31 >= 1
X23 + X32 >= 1

END

! Restricao 3: variáveis sao inteiras;
GIN X01
GIN X10
GIN X02
GIN X20
GIN X03
GIN X30
GIN X13
GIN X31
GIN X23
GIN X32

```

**Fonte: Autoria Própria**

Definido o modelo matemático, já no formato que o programa LINDO lê, a etapa seguinte foi resolvê-lo. Assim, ao resolver o modelo, o programa retorna uma janela, como é mostrado na Figura 16 a seguir.

**Figura 16 – Solução do Problema das Pontes de Königsberg utilizando o modelo de Bodin (1983) no software LINDO.**

The screenshot shows the LINDO Reports Window with the following data:

OBJECTIVE FUNCTION VALUE		
1)	9.000000	
VARIABLE	VALUE	REDUCED COST
X01	2.000000	1.000000
X10	1.000000	1.000000
X02	1.000000	1.000000
X20	1.000000	1.000000
X03	0.000000	1.000000
X30	1.000000	1.000000
X13	1.000000	1.000000
X31	0.000000	1.000000
X23	1.000000	1.000000
X32	1.000000	1.000000
ROW	SLACK OR SURPLUS	DUAL PRICES
2)	0.000000	0.000000
3)	0.000000	0.000000
4)	0.000000	0.000000
5)	0.000000	0.000000
6)	1.000000	0.000000
7)	0.000000	0.000000
8)	0.000000	0.000000
9)	0.000000	0.000000
10)	1.000000	0.000000
NO. ITERATIONS= 52		
BRANCHES= 14 DETERM. = 1.000E 0		

Fonte: Autoria Própria

O campo "OBJECTIVE FUNCTION VALUE" (Valor Ótimo da Função Objetivo) refere-se à distância total que foi percorrida, sendo a solução ótima para o modelo, encontrada pelo LINDO. Assim, com base nos resultados apresentados, o resultado da função objetivo é 9, e no nosso exemplo indica que foram percorridas 9 arestas.

Na coluna "VALUE" (Valor das Variáveis Originais) aparece o valor obtido para as variáveis na ordem em que aparecem na coluna "VARIABLE". Note que,  $x_{01} = 1$ , ou seja, a aresta (0, 1) foi percorrida uma vez do ponto (vértice) 0 e chegando no vértice 1.

A partir dos valores obtidos na solução, podemos construir uma sequência de percorri-mento das arestas. Se considerarmos sair do vértice 0, temos uma sequência de arestas  $x_{01}$ ,  $x_{10}$ ,  $x_{02}$ ,  $x_{20}$ ,  $x_{01}$ ,  $x_{13}$ ,  $x_{32}$ ,  $x_{23}$ ,  $x_{30}$ . Usando a notação a, b, c, d, e, f, g do grafo da Figura 12, temos o caminho percorrido, isto é, a, b, c, d, a, g, f, f, e.

## 4.2 TECNOLOGIAS UTILIZADAS

O algoritmo do modelo matemático foi implementado em um notebook DELL, processador Intel(R) Core i3, 4GB de RAM.

Os dados necessários para a implementação do problema, ou seja, o cálculo das distâncias entre os nós (esquinas) foram obtidas no software *Google Earth*.

A matriz distância, que foi obtida através dos dados do Google Earth, foi construída no software *Excel*.

O código para escrever o modelo matemático do PCCND no formato usado pelo LINDO foi implementado no software *DEV-C++*.

O modelo matemático foi resolvido utilizando software *LINDO*.

O software [Xpress-Optimizer 2004] faz parte do FICO Xpress Optimization Suite. O otimizador FICO Xpress é um solucionador comercial para programação linear e programação linear inteira mista. Esse software foi utilizado para obter a solução ótima do problema. Este último surgiu como alternativa ao software LINDO, um vez que o LINDO não pode ser usado para resolver o problema do carteiro chinês formulado.

Para ter acesso ao *software* Xpress License, a FICO criou a Licença da Comunidade Xpress, disponível para todos. Para maiores informações, consulte as referências.

## 5 IMPLEMENTAÇÃO COMPUTACIONAL E ANÁLISE DOS RESULTADOS

Antes da implementação computacional, os dados foram coletados com a ajuda de um funcionário da empresa de Correios da cidade de Bandeirantes-PR. Durante um mês, foi realizado uma entrevista para se conhecer o funcionamento da empresa e dos carteiros, desde a etapa da separação de correspondências até o processo de entrega.

Como mencionado anteriormente, este trabalho está focado em obter o menor trajeto para a região 364-B. Para obter a distância entre as esquinas do bairro estudado, foi utilizado o *Google Maps*, onde foram marcadas as esquinas (vértices) e medidas as distâncias das ruas entre as esquinas. Na Figura 17a, é possível observar a região 364-B com as ruas destacadas e na Figura 17b o grafo correspondente a essa região, já com as medidas das ruas.

O *Google Maps* é um aplicativo desenvolvido e distribuído pela empresa que leva a logomarca Google, cuja função é disponibilizar um modelo tridimensional do globo terrestre. Este modelo é construído a partir da captura de imagens via satélite, obtidas diversos ângulos de visão, imagens áreas (através de aeronaves) e sistema de informação geográficas 3D. Este aplicativo foi utilizado para a obtenção das distâncias entre os vértices (esquinas).

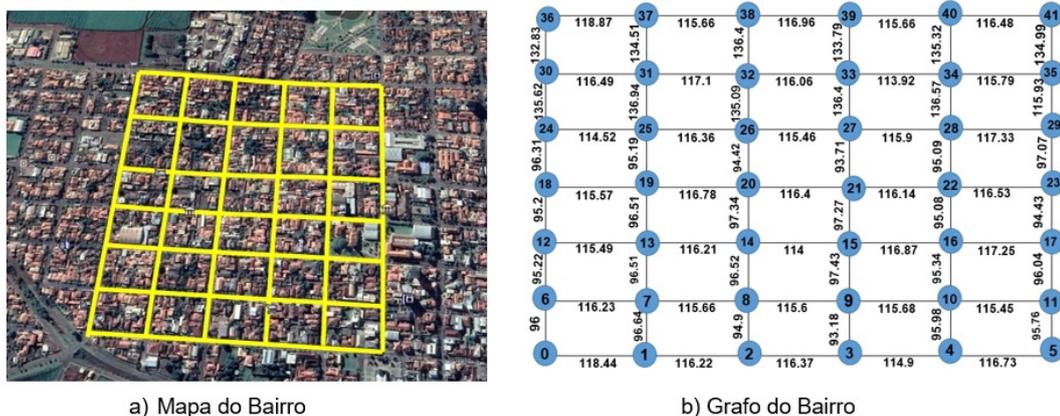
Após calcular todos os vértices (esquinas) e as distâncias entre eles, foi utilizado o *Excel* para formar uma matriz distância do bairro. Depois de concluída a construção da matriz distância, o modelo matemático do PCCND foi escrito com um código implementado no *software DEV-C++* e, em seguida, resolvido usando o *software Xpress*.

### 5.1 ETAPAS DA IMPLEMENTAÇÃO COMPUTACIONAL

A implementação computacional dos algoritmos para solução do problema foi dividida em três etapas: gerar o grafo correspondente a região estudada; criar a matriz distância do grafo; escrever o modelo matemático do PCCND e resolver o modelo.

Na primeira etapa, foi utilizado um grafo que representa o bairro real da cidade. Para a implementação dos algoritmos foi utilizado como referência o mapa da Figura 17, obtendo o grafo referente ao Distrito 364-B. O grafo em questão apresenta 42 vértices e 71 arestas. Tanto o mapa como as distâncias entre cada um dos vértices foram obtidas por meio do *software Google Maps*.

Figura 17 – Imagem da região 364-B e o grafo gerado para esta região



Fonte: Autoria Própria.

Na segunda etapa, foi construída no *Excel* a matriz distância que foi obtida através dos dados coletados na primeira etapa. Uma parte desta matriz pode ser observada no Apêndice B.

Na terceira etapa, depois de concluída a matriz distância, foi utilizado o *software DEV-C++* para a imprimir o modelo matemático do PCCND. No Apêndice B, encontra-se uma parte do programa.

A última etapa, consiste em resolver o modelo gerado utilizando o *software* LINDO e Xpress.

## 5.2 ANÁLISE DOS RESULTADOS

O *software* (Linear, Interactive and Discrete Optimizer) (LINDO), é um *software* desenvolvido pela Lindo Systems Inc., de Chicago, EUA, exclusivo para resolução de modelos matemáticos de programação linear quadrática ou inteira, com capacidade de resolver problemas com até 100.000 variáveis. Suas versões disponíveis variam de acordo com a quantidade de restrições e variáveis, sendo a de maior extensão nomeada *Extended* [Prado 1998].

Deste modo, ao aplicar o modelo matemático no *software LINDO* (versão demo 6.1), não foi possível concluir a análise de resultados, pois a capacidade disponível nesta versão é de 50 variáveis inteiras, onde ocorreu um erro no término da execução constando 51 variáveis inteiras no problema estudado.

Desta forma, fazendo novamente uma análise no mapa da cidade, é possível notar que cada distrito possui quase a mesma quantidade de quadras, conseqüentemente, a mesma quantidade de vértices e arestas. Sendo assim, optamos por permanecer com o mesmo grafo do bairro da Figura 17, e com os mesmos procedimentos feitos anteriormente, só que agora, resolvendo o modelo matemático do PCCND no *software* Xpress. O modelo antes gerado para o LINDO teve que ser reescrito, visto que o Xpress utiliza outros comandos para a solução do problema. O código implementado no DEV-C++ foi reescrito e assim gerado um novo arquivo para o modelo matemático, como mostrado na Figura 18. No Apêndice C, encontra-se o código todo gerado no software DEV-C++.

**Figura 18 – Código escrito em C para gerar a formulação matemática no formato lido pelo Xpress**

```
//Open the out file to write the results
out_file = fopen("exemploP_XPRESS.mos", "a");
fprintf(out_file, " !-----;");
fprintf(out_file, "\n !# Escreve o modelo matemático de Bodin (1983) para ser resolvido no XPRESS.; ");
fprintf(out_file, "\n !# Usa a matriz de adjacencia do problema que está no arquivo exemplo.txt;");
fprintf(out_file, "\n !-----;");

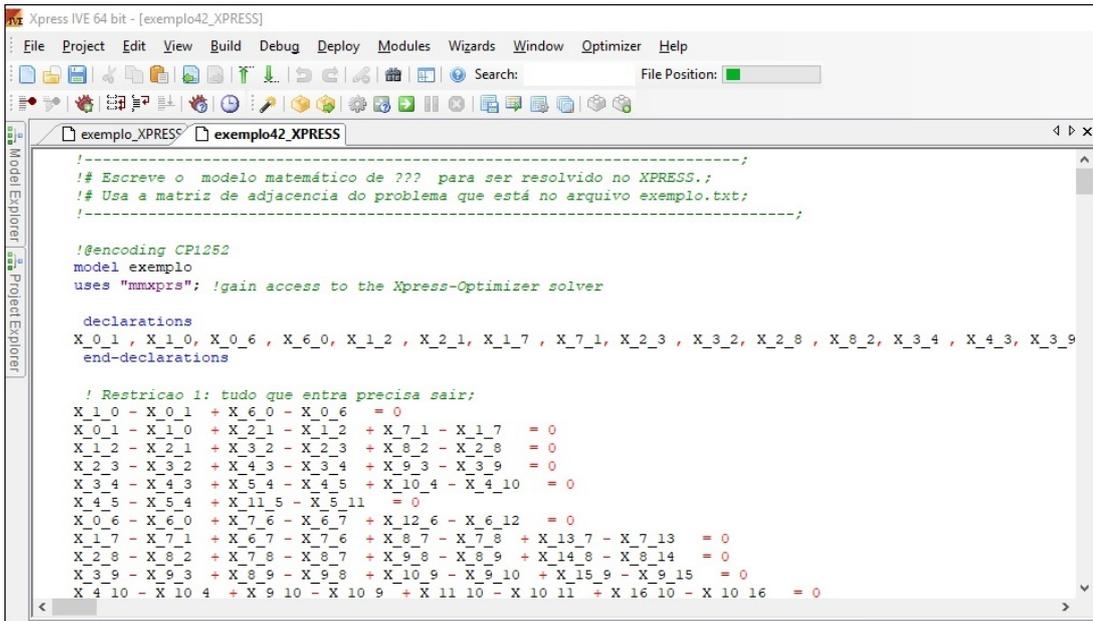
//funções do programa XPress
fprintf(out_file, "\n \n !encoding CP1252 ");
fprintf(out_file, "\n model exemplo ");
fprintf(out_file, "\n uses \"mxxprs\"; !gain access to the Xpress-Optimizer solver ");

//declara as variaveis do modelo
fprintf(out_file, "\n\n declarations ");
cont=0;
for(i=0;i<n;i++)
    for(j=i+1;j<n;j++)
        if( ADJ[i][j] != 0 ) {
            if(cont==0) {
                fprintf( out_file, " \n X %d %d , X %d %d",i,j,j,i);
                cont=1;
            }
            else fprintf( out_file, " , X %d %d , X %d %d",i,j,j,i);
        }
    fprintf( out_file, " : mpvar !se real positiva ");
    fprintf(out_file, "\n end-declarations ");
```

Fonte: Autoria Própria

A Figura 19, a seguir, mostra o modelo que seria resolvido com o Xpress.

Figura 19 – Modelo matemático reproduzido no software Xpress



Fonte: Autoria Própria

Definido a modelo matemática, a etapa seguinte da pesquisa foi solucioná-la utilizando o Software Xpress. Quando resolvido, foi gerado o relatório mostrado na Figura 20, em que:

O "Optimal objective value" refere-se à distância total que foi percorrida, sendo a solução ótima para o modelo, encontrada pelo Xpress. Assim, com base nos resultados apresentados, o resultado da função objetiva 9179,76, indica que foram percorridas 9179,76 arestas.

Note que as variáveis  $x_{0_1} = 0$ , significa que a aresta (0, 1) não percorreu nenhuma vez do ponto 0 ao ponto 1, já as variáveis  $x_{1_0} = 1$ , significa que a aresta (1, 0) percorreu uma vez saindo do ponto 1 e chegando ao ponto 0, e assim sucessivamente.

Tabela 4 – Solução obtida usando o Xpress para resolver o problema do PCCND gerado

$x_{0_1} = 0$	$x_{4_5} = 0$	$x_{8_14} = 1$	$x_{18_12} = 0$	$x_{16_22} = 0$	$x_{26_20} = 0$	$x_{25_26} = 1$	$x_{34_28} = 1$	$x_{33_34} = 1$	$x_{39_38} = 0$
$x_{1_0} = 1$	$x_{5_4} = 1$	$x_{14_8} = 0$	$x_{13_14} = 0$	$x_{22_16} = 1$	$x_{21_22} = 0$	$x_{26_25} = 0$	$x_{29_35} = 0$	$x_{34_33} = 0$	$x_{39_40} = 2$
$x_{0_6} = 1$	$x_{4_10} = 1$	$x_{9_10} = 0$	$x_{14_13} = 1$	$x_{17_23} = 1$	$x_{22_21} = 1$	$x_{25_31} = 0$	$x_{35_29} = 2$	$x_{33_39} = 1$	$x_{40_39} = 0$
$x_{6_0} = 0$	$x_{10_4} = 1$	$x_{10_9} = 1$	$x_{13_19} = 0$	$x_{23_17} = 1$	$x_{21_27} = 0$	$x_{31_25} = 1$	$x_{30_31} = 1$	$x_{39_33} = 0$	$x_{40_41} = 1$
$x_{1_2} = 0$	$x_{5_11} = 1$	$x_{9_15} = 1$	$x_{19_13} = 1$	$x_{18_19} = 1$	$x_{27_21} = 1$	$x_{26_27} = 1$	$x_{31_30} = 0$	$x_{34_35} = 1$	$x_{41_40} = 0$
$x_{2_1} = 1$	$x_{6_7} = 1$	$x_{15_9} = 0$	$x_{14_15} = 0$	$x_{19_18} = 0$	$x_{22_23} = 0$	$x_{27_26} = 0$	$x_{30_36} = 1$	$x_{35_34} = 0$	
$x_{1_7} = 1$	$x_{7_6} = 1$	$x_{10_11} = 1$	$x_{15_14} = 1$	$x_{18_24} = 1$	$x_{23_22} = 1$	$x_{26_32} = 1$	$x_{36_30} = 0$	$x_{34_40} = 0$	
$x_{7_1} = 1$	$x_{6_12} = 1$	$x_{11_10} = 0$	$x_{14_20} = 1$	$x_{24_18} = 0$	$x_{22_28} = 0$	$x_{32_26} = 0$	$x_{31_32} = 1$	$x_{40_34} = 1$	
$x_{2_3} = 0$	$x_{12_6} = 0$	$x_{10_16} = 0$	$x_{20_14} = 0$	$x_{19_20} = 0$	$x_{28_22} = 1$	$x_{27_28} = 0$	$x_{32_31} = 0$	$x_{35_41} = 0$	
$x_{3_2} = 2$	$x_{7_8} = 1$	$x_{16_10} = 1$	$x_{15_16} = 1$	$x_{20_19} = 1$	$x_{23_29} = 0$	$x_{28_27} = 1$	$x_{31_37} = 0$	$x_{41_35} = 1$	
$x_{2_8} = 1$	$x_{8_7} = 1$	$x_{11_17} = 0$	$x_{16_15} = 0$	$x_{19_25} = 1$	$x_{29_23} = 1$	$x_{27_33} = 1$	$x_{37_31} = 1$	$x_{36_37} = 1$	
$x_{3_4} = 0$	$x_{7_13} = 0$	$x_{17_11} = 1$	$x_{15_21} = 0$	$x_{25_19} = 0$	$x_{24_25} = 0$	$x_{33_27} = 0$	$x_{32_33} = 1$	$x_{37_36} = 0$	
$x_{4_3} = 1$	$x_{13_7} = 1$	$x_{12_13} = 0$	$x_{21_15} = 1$	$x_{20_21} = 0$	$x_{25_24} = 1$	$x_{28_29} = 0$	$x_{33_32} = 0$	$x_{37_38} = 1$	
$x_{3_9} = 0$	$x_{8_9} = 1$	$x_{13_12} = 1$	$x_{16_17} = 1$	$x_{21_20} = 1$	$x_{24_30} = 2$	$x_{29_28} = 1$	$x_{32_38} = 1$	$x_{38_37} = 1$	
$x_{9_3} = 1$	$x_{9_8} = 0$	$x_{12_18} = 2$	$x_{17_16} = 0$	$x_{20_26} = 1$	$x_{30_24} = 0$	$x_{28_34} = 0$	$x_{38_32} = 0$	$x_{38_39} = 1$	

Fonte: Autoria Própria

Assim, com base nos resultados apresentados, o resultado da função objetivo é de 9179,75 metros, ou seja, essa é a distância mínima que o carteiro pedestre precisa andar para que todas as arestas tenham sido percorridas.



## 6 CONCLUSÃO

Esse trabalho apresentou um estudo sobre a Otimização de rotas para a entrega de correspondências, com objetivo de minimizar a roteirização dos carteiros, utilizando o método do Carteiro Chinês. Destaca-se como tópicos-chaves para a realização deste trabalho os conceitos básicos sobre a Teoria dos Grafos, o Problema do Carteiro Chinês não Direcionado (PCCND) e o funcionamento dos softwares de otimização (LINDO e Xpress).

O estudo de caso foi realizado para a obtenção de dados claros e objetivos sobre como cada carteiro estabelece suas rotas. Assim, foi feito um levantamento de dados referentes ao funcionamento da empresa de Correios, desde a etapa da separação até o momento da entrega de correspondências realizada pelos carteiros, estabeleceu as rotas de acordo com seu conhecimento adquirido após anos de experiência.

O modelo matemático do PCCND foi aplicado tanto no exemplo do problema das Pontes de Königsberg como no problema real estudado. Os dados do problema real obtido foram modelados usando o Excel e o DEV-C++ e resolvido utilizando o software Xpress, que encontrou a solução ótima para a problemática apresentada.

Após a solução encontrada, não foi possível ser feita a comparação do resultado ótimo obtido pelos dados da empresa de Correios, pois não foi fornecida.

Por fim, por meio da realização do presente estudo, observou-se que a utilização do modelo matemático do PCC e a programação do modelo matemático no software Xpress são capazes de solucionar os modelos matemáticos propostos de forma viável, desta forma, recomenda-se sua utilização sempre que necessário.

Apesar de não ter sido estudado neste trabalho, a metodologia testada aqui pode ser aplicada para os demais distritos postais em que o grafo gerado nestas regiões seja não orientado. Já para os distritos em que o grafo seria orientado ou mesmo misto, é necessário utilizar alguma outra formulação que atenda estas características.

### 6.1 LIMITAÇÕES

Nas etapas da implementação computacional, utilizou-se o software LINDO, este, em sua versão gratuita, suporta modelos matemáticos de até 50 variáveis inteiras. Por este motivo se o modelo matemático gerado por este programa tiver mais que 50 variáveis inteiras este não irá funcionar. Havendo uma licença que não possua limite de variáveis este trabalho pode resolver qualquer instância de problema. Desta forma, teve a necessidade de utilizar o software Xpress para terminar de executar os dados obtidos e chegar na solução desejada.



## REFERÊNCIAS

- ABREU, Raphael Lorenzeto de. **Ficheiro:Paraná Município de Bandeirantes.svg**. 2006. Disponível em: <[https://pt.wikipedia.org/wiki/Ficheiro:Parana\\_Municip\\_Bandeirantes.svg](https://pt.wikipedia.org/wiki/Ficheiro:Parana_Municip_Bandeirantes.svg)>. Acesso em: 5 Set. 2018. Citado na página 23.
- ARENALES, Marcos et al. **Pesquisa operacional: para cursos de engenharia**. [S.l.]: Elsevier Brasil, 2015. Citado na página 27.
- BODIN, Lawrence. Routing and scheduling of vehicles and crews, the state of the art. **Comput. Oper. Res.**, v. 10, n. 2, p. 63–211, 1983. Citado na página 39.
- COSTA, Deise Maria Bertholdi et al. Operations research techniques applied in the post services. **Gestão & Produção**, v. 8, n. 1, p. 37–55, 2001. Citado na página 21.
- EDMONDS, Jack; JOHNSON, Ellis L. Matching, euler tours and the chinese postman. **Mathematical programming**, Springer, v. 5, n. 1, p. 88–124, 1973. Citado na página 36.
- GOLDBARG, Marco Cesar; LUNA, Henrique Pacca L. **Otimização combinatória e programação linear: modelos e algoritmos**. [S.l.]: Elsevier, 2005. Citado 4 vezes nas páginas 21, 32, 36 e 39.
- GOMES, Marcos José Negreiros et al. O problema do carteiro chinês, algoritmos exatos e um ambiente mvi para análise de suas instâncias: sistema xnês. **Pesquisa Operacional**, v. 29, n. 2, p. 323–363, 2009. Citado na página 21.
- GUAN, Meigu. Graphic programming using odd and even points. **Chinese Math.**, v. 1, p. 237–277, 1962. Citado na página 34.
- IBGE. **População de Bandeirantes, 2017**. Disponível em: <<https://cidades.ibge.gov.br/brasil/pr/bandeirantes/panorama>>. Acesso em: 5 Set. 2018. Citado na página 24.
- KONOWALENKO, F. **Problema do carteiro chinês não-orientado e misto para a otimização de rotas na cidade de Irati/PR. 108 f.** Tese (Doutorado) — Dissertação (Mestrado em Ciências). Universidade Federal do Paraná, Curitiba. Citado na página 27.
- LICENSE, FICO Xpress Community. **Atualizações recentes da FICO Xpress Community License**. Disponível em: <[http://content.fico.com/xpress-optimization-community-license?utm\\_source=FICO-Community&utm\\_medium=optimization-homepage](http://content.fico.com/xpress-optimization-community-license?utm_source=FICO-Community&utm_medium=optimization-homepage)>. Acesso em: 18 Jun. 2019. Citado na página 42.
- PRADO, Darci. **Programação linear**. [S.l.]: Editora de Desenvolvimento Gerencial, 1998. Citado na página 44.
- PRESTES, Edson. Teoria dos grafos. **Porto Alegre. 2011a**. Disponível em:< <http://www.inf.ufrgs.br/~prestes/Courses/Graph%20Theory/GrafosA6.pdf>>. Acesso em, v. 4, 2016. Citado na página 35.
- SHERAFAT, Hassan et al. Algoritmos heurísticos de cobertura de arcos. Florianópolis, SC, 2004. Citado na página 34.
- SMIDERLE, Andreia; STEINER, MT Arns; WILHELM, VE. Técnicas da pesquisa operacional aplicadas a um problema de cobertura de arcos. **Trends in Applied and Computational Mathematics**, v. 5, n. 2, p. 347–356, 2004. Citado 2 vezes nas páginas 21 e 32.

XPRESS-OPTIMIZER, Dash. **16.01. 05: Getting Started and Reference Manual**, Dash Optimization Ltd. 2004. Citado na página 42.

## A APÊNDICE A: CÓDIGO DO MODELO MATEMÁTICO UTILIZANDO O SOFTWARE DEV-C++ PARA IMPLEMENTAR NO SOFTWARE LINDO

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>

int main()
{
    /*=====Parametros e variaveis=====*/

    int n;                                //número de nós do grafo
    int i,j;

    FILE *problem_file, *out_file; // criando a variável ponteiro para os arquivos de entrada e de saída
    char current_line[101] = "";
```

```
    /*=====Entrada manual enquanto nao resolve problema do arquivo de entrada=====*/
/*
n=4;
double ADJ[4][4];
for(i=0;i<n;i++)
    for(j=0;j<n;j++)
    {
        ADJ[i][j]=1;
        if(i==j) ADJ[i][j]=0;
    }
ADJ[1][3]=0; ADJ[3][1]=0;

// Imprime dados passados
printf("%d", n);
for(i=0;i<n;i++)
{
    printf("\n linha %d: ", i);
    for(j=0;j<n;j++)
        printf("%.2f ", ADJ[i][j]);
}
*/

/*=====Abre, lê e fecha arquivo dos dados de entrada=====*/
```

```
//Open the file to read
if(!(problem_file = fopen("exemplodaspontes.txt", "r")))
    { printf("The file can not be opened"); return 0; }

fgets(current_line, 100, problem_file); //Lê ate 99 caracteres e salva em current_line. fgets le a linha inteira
fscanf(problem_file, "%d", &n); //lê quantidade de nós = n , fscanf le como se fosse do teclado

//Aloca e carrega matriz adjacencia do grafo de n nós
double ADJ[n][n], aux;
for(i=0;i<n;i++)
{
    for(j=0;j<n;j++)
    {
        fscanf(problem_file, "%lf", &aux);
        ADJ[i][j]=aux;
        printf("%f ", ADJ[i][j]);
    }
}
fclose(problem_file);
```

```

// Imprime dados lidos
printf("%d", n);
for(i=0;i<n;i++)
{
    printf("\n linha %d: ", i);
    for(j=0;j<n;j++)
        printf("%f ", ADJ[i][j]);
}

```

```

//Open the out file to write the results
out_file = fopen("out_exemplodaspones_LINDO.txt", "a");
fprintf(out_file, " !-----");
fprintf(out_file, "\n !# Escreve o modelo matemático de Bodin (1983) para ser resolvido no LINDO. ");
fprintf(out_file, "\n !# Usa a matriz de adjacencia do problema que está no arquivo exemplo.txt.");
fprintf(out_file, "\n !-----");

//Imprime função objetivo
fprintf( out_file, " \n \n ! Função objetivo: minimizar o trajeto percorrido.");
fprintf(out_file, "\n MIN ");
int cont=0;
for(i=0;i<n;i++)
    for(j=0;j<n;j++)
    {
        if( ADJ[i][j] != 0 )
        {
            if(cont==0)
            {
                fprintf(out_file, " %g X%d%d", ADJ[i][j],i,j);
                cont=1;
            }
            else
                fprintf(out_file, " + %g X%d%d", ADJ[i][j],i,j);
        }
    }

fprintf( out_file, " \n \n SUBJECT TO");

```

```

//Imprime restricao 1: tudo que entra precisa sair
fprintf( out_file, " \n \n ! Restricao 1: tudo que entra precisa sair.");
for(i=0;i<n;i++)
{
    cont=0;
    fprintf( out_file, "\n");
    for( j=0;j<n;j++)
        if( ADJ[j][i] != 0 )
        {
            if(cont==0)
            {
                fprintf( out_file, " X%d%d - X%d%d ",j,i,i,j);
                cont=1;
            }
            else fprintf( out_file, " + X%d%d - X%d%d ",j,i,i,j);
        }
    if(cont==1)
        fprintf( out_file, " = 0 ");
}

//Imprime restricao 2: percorre todas as arestas ao menos uma vez.
fprintf( out_file, " \n \n ! Restricao 2: percorre todas as arestas ao menos uma vez.");
for(i=0;i<n;i++)
    for(j=i+1;j<n;j++)
        if( ADJ[i][j] != 0 )
        {
            fprintf( out_file, "\n");
            fprintf( out_file, " X%d%d + X%d%d >= 1",i,j,j,i);
        }

fprintf( out_file, " \n \n END");

```

```
//Imprime restricao 3: variáveis inteiras
fprintf( out_file, " \n \n ! Restricao 3: variáveis sao inteiras;");
for(i=0;i<n;i++)
    for(j=i+1;j<n;j++)
        if( ADJ[i][j] != 0 )
        {
            fprintf( out_file, " \n GIN X%d%d",i,j);
            fprintf( out_file, " \n GIN X%d%d",j,i);
        }

fclose(out_file);

return 1; //retorna 0 caso entre em algum erro
}
```





## B APÊNDICE B: PARTE DA MATRIZ DISTÂNCIA

d(i,j)	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21
0	0	118,44	0	0	0	0	96	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	118,44	0	116,22	0	0	0	0	96,64	0	0	0	0	0	0	0	0	0	0	0	0	0	0
2	0	116,22	0	116,37	0	0	0	0	94,9	0	0	0	0	0	0	0	0	0	0	0	0	0
3	0	0	116,37	0	114,9	0	0	0	0	93,18	0	0	0	0	0	0	0	0	0	0	0	0
4	0	0	0	114,9	0	116,73	0	0	0	0	95,98	0	0	0	0	0	0	0	0	0	0	0
5	0	0	0	0	116,73	0	0	0	0	0	0	95,76	0	0	0	0	0	0	0	0	0	0
6	96	0	0	0	0	0	116,23	0	115,66	0	0	0	95,22	0	0	0	0	0	0	0	0	0
7	0	96,64	0	0	0	0	0	115,66	0	115,6	0	0	0	96,51	0	0	0	0	0	0	0	0
8	0	0	94,9	0	0	0	0	115,6	0	115,68	0	0	0	0	96,52	0	0	0	0	0	0	0
9	0	0	0	93,18	0	0	0	115,6	0	115,68	0	0	0	0	0	97,43	0	0	0	0	0	0
10	0	0	0	0	95,98	0	0	0	115,68	0	115,45	0	0	0	0	0	95,34	0	0	0	0	0
11	0	0	0	0	0	95,76	0	0	0	115,45	0	0	0	0	0	0	96,04	0	0	0	0	0
12	0	0	0	0	0	0	95,22	0	0	0	0	95,2	0	0	0	0	0	95,2	0	0	0	0
13	0	0	0	0	0	0	0	96,51	0	0	115,49	0	0	0	0	0	0	0	96,51	0	0	0
14	0	0	0	0	0	0	0	0	96,52	0	0	0	0	0	114	0	0	0	0	97,34	0	0
15	0	0	0	0	0	0	0	0	0	97,43	0	0	0	0	114	0	116,87	0	0	0	97,27	0
16	0	0	0	0	0	0	0	0	0	0	95,34	0	0	0	0	116,87	0	117,25	0	0	0	0
17	0	0	0	0	0	0	0	0	0	0	0	96,04	0	0	0	0	0	0	0	0	0	0
18	0	0	0	0	0	0	0	0	0	0	0	0	95,2	0	0	0	0	0	0	115,57	0	0
19	0	0	0	0	0	0	0	0	0	0	0	0	0	96,51	0	0	0	0	0	0	116,78	0
20	0	0	0	0	0	0	0	0	0	0	0	0	0	0	97,34	0	0	0	0	0	0	116,4
21	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	97,27	0	0	0	0	0	0

## C APÊNDICE C: CÓDIGO DO MODELO MATEMÁTICO UTILIZANDO O SOFTWARE DEV-C++ PARA IMPLEMENTAR O MODELO NO SOFTWARE XPRESS

```

#include <stdio.h>
#include <stdlib.h>
#include <string.h>

int main(){
    // Parametros e variaveis
    int i,j,cont,n; //n é número de nós do grafo
    double ADJ[n][n]; //matriz de adjacencia do grafo
    FILE *problem_file, *out_file;
    char current_line[101] = "";

    //Open the file to read
    if(!(problem_file = fopen("exemploP.txt", "r")))
        { printf("The file can not be opened"); return 0; }

    fgets(current_line, 100, problem_file); //Lê ate 99 caracteres e salva em current_line. fgets le a linha inteira
    fscanf(problem_file, "%d", &n); //lê quantidade de nós = n , fscanf le como se fosse do teclado

    //Aloca e carrega matriz adjacencia do grafo de n nós
    for(i=0;i<n;i++)
        for(j=0;j<n;j++)
            fscanf(problem_file, "%lf", &ADJ[i][j]);

    fclose(problem_file);

```

```

//Imprime restricao 1: tudo que entra precisa sair
fprintf( out_file, " \n \n ! Restricao 1: tudo que entra precisa sair;");
for(i=0;i<n;i++) {
    cont=0;
    fprintf( out_file, "\n");
    for( j=0;j<n;j++)
        if( ADJ[j][i] != 0 ) {
            if(cont==0) {
                fprintf( out_file, " X_%d_%d - X_%d_%d ",j,i,i,j);
                cont=1;
            }
            else fprintf( out_file, " + X_%d_%d - X_%d_%d ",j,i,i,j);
        }
    if(cont==1)
        fprintf( out_file, " = 0 ");
}

//Imprime restricao 2: percorre todas as arestas ao menos uma vez.
fprintf( out_file, " \n \n ! Restricao 2: percorre todas as arestas ao menos uma vez;");
for(i=0;i<n;i++)
    for(j=i+1;j<n;j++)
        if( ADJ[i][j] != 0 ) {
            fprintf( out_file, "\n");
            fprintf( out_file, " X_%d_%d + X_%d_%d >= 1",i,j,j,i);
        }
}

```

```

//Imprime restricao 3: variáveis inteiras
fprintf( out_file, " \n \n ! Restricao 3: variáveis sao inteiras;");
for(i=0;i<n;i++)
    for(j=i+1;j<n;j++)
        if( ADJ[i][j] != 0 )
            fprintf( out_file, " \n X_%d_%d is_integer; X_%d_%d is_integer;",i,j,j,i);

//Imprime função objetivo
fprintf( out_file, " \n \n ! Função objetivo: minimizar o trajeto percorrido;");
fprintf(out_file, "\n minimize ( ");
cont=0;
for(i=0;i<n;i++)
    for(j=i+1;j<n;j++)
        if( ADJ[i][j] != 0 ) {
            if(cont==0) {
                fprintf(out_file, " %g*X_%d_%d + %g*X_%d_%d", ADJ[i][j],i,j,ADJ[i][j],j,i);
                cnt=1;
            }
            else
                fprintf(out_file, " + %g*X_%d_%d + %g*X_%d_%d", ADJ[i][j],i,j,ADJ[i][j],j,i);
        }
    fprintf(out_file, ") ");

```

```

//Comandos do XPress para imprimir solução
fprintf(out_file, "\n \n writeln(\"Optimal objective value: \", getobjval);");
fprintf(out_file, "\n");
for(i=0;i<n;i++)
    for(j=i+1;j<n;j++)
        if( ADJ[i][j] != 0 ) {
            fprintf(out_file, "\n writeln(\" X_%d_%d = \", getsol( X_%d_%d)) ", i,j,i,j);
            fprintf(out_file, "\n writeln(\" X_%d_%d = \", getsol( X_%d_%d)) ", j,i,j,i);
        }

//Comando do XPress para finalizar modelo
fprintf(out_file, "\n \n end-model ");

//fecha arquivo de impressao
fclose(out_file);

return 1;
}

```