



UNIVERSIDADE FEDERAL DE PERNAMBUCO
CENTRO DE TECNOLOGIA E GEOCIÊNCIA
DEPARTAMENTO DE ENGENHARIA DE PRODUÇÃO
PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA DE PRODUÇÃO

LUAN EMERSON SOARES DE LIMA

**MODELAGEM DO ROTEAMENTO DE LEITURISTAS: uma abordagem cluster first
– route second para o problema do carteiro chinês capacitado**

Recife

2021

LUAN EMERSON SOARES DE LIMA

**MODELAGEM DO ROTEAMENTO DE LEITURISTAS: uma abordagem cluster first
– route second para o problema do carteiro chinês capacitado**

Dissertação apresentada ao Programa de Pós-Graduação em Engenharia de Produção da Universidade Federal de Pernambuco, Centro de Tecnologia e Geociências, como requisito parcial para a obtenção do título de mestre em Engenharia de Produção.

Área de concentração: Pesquisa Operacional.

Orientador: Prof. Dr. Sóstenes Luiz Soares Lins.

Recife

2021

Catálogo na fonte
Bibliotecária Rosineide Mesquita G. Luz, CRB-4 / 1368

L732m Lima, Luan Emerson Soares de.
 Modelagem do Roteamento de Leituristas: uma abordagem cluster first - route
 second para o problema do carteiro chinês capacitado / Luan Emerson Soares de
 Lima. - 2021
 123 folhas, il., gráfs., tabs.

 Orientador: Prof. Sóstenes Luiz Soares Lins, PHD.
 Dissertação (Mestrado) – Universidade Federal de Pernambuco. CTG.
 Programa de Pós-Graduação em Engenharia de Produção, 2016.

 Inclui Referências e Apêndices.

 1. Engenharia de Produção. 2. Teoria dos Grafos. 3. Otimização. 4.
 Problema do Carteiro Chinês. 5. Roteamento de Leiturista. I. Lins,
 Sóstenes Luiz Soares (Orientador). II. Título.

UFPE

658.5 CDD (22. ed.)

BCTG/2022-49

LUAN EMERSON SOARES DE LIMA

**MODELAGEM DO ROTEAMENTO DE LEITURISTAS: uma abordagem cluster first
– route second para o problema do carteiro chinês capacitado**

Dissertação apresentada ao Programa de Pós-Graduação em Engenharia de Produção da Universidade Federal de Pernambuco, Centro de Tecnologia e Geociências, como requisito parcial para a obtenção do título de mestre em Engenharia de Produção. Área de concentração: Pesquisa Operacional.

Aprovado em: 25 / 02 / 2021.

BANCA EXAMINADORA

Prof. Dr. SÓSTENES LUIZ SOARES LINS (Orientador)
Universidade Federal de Pernambuco

Prof^a. Dr. ISIS DIDIER LINS (Examinador Interno)
Universidade Federal de Pernambuco

Prof. Dr. EMERSON ALEXANDRE DE OLIVEIRA LIMA (Examinador Externo)
Universidade de Pernambuco

AGRADECIMENTOS

Diante dos desafios do mestrado gostaria de agradecer a oportunidade de estudar oferecida pelo Programa de Pós-Graduação em Engenharia de Produção (PPGEP) da Universidade Federal de Pernambuco (UFPE). Gostaria de agradecer à Coordenação de Aperfeiçoamento de Pessoal de Nível Superior (CAPES) e ao Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq) pelo aporte financeiro despendido.

Meus agradecimentos a minha família e amigos, especialmente Andersson Silva. Gostaria de agradecer ao meu orientador Sóstenes Lins e ao restante da banca de mestrado nas pessoas de Ísis Didier e Emerson Alexandre.

RESUMO

O Problema de Roteamento de Leituristas (PRL) é um problema de otimização que contempla setores ligados ao fornecimento de gás natural, energia elétrica ou água encanada, onde empresas fornecedoras necessitam periodicamente mobilizar trabalhadores para vistoria e emissão de faturas dos pontos de consumo. O PRL trata as ruas da malha urbana estudada como arestas, às arestas estão relacionadas distâncias a serem percorridas, a essas arestas também estão associadas demandas de tempo para atravessamento e vistoria dos pontos de consumo. O PRL, por sua vez, consiste em encontrar rotas que minimizem a distância percorrida por leituristas dado uma carga de trabalho previamente atribuída, isso significa que as arestas dotadas de demandas de tempo para serem atravessadas e vistoriadas devem estar designadas aos leituristas buscando respeitar a carga de trabalho estabelecida. Por meio de uma revisão sistemática de literatura identificaram-se lacunas presentes na literatura, como o baixo número de publicações concernentes a técnicas de resolução do PRL. O presente trabalho propõe uma abordagem *cluster first – route second* para a resolução do PRL. A abordagem proposta é aplicada em duas fases, na primeira fase agrupa-se as ruas da localidade estudada em *clusters* por meio da aplicação do Problema das p -Medianas, a atribuição da demanda de cada rua ou segmento de rua aos *clusters* é limitada pela capacidade do leiturista, logo cada *cluster* corresponde a uma rota. Em uma segunda fase, a conectividade de cada *cluster* é verificada, caso o subgrafo de um *cluster* não seja conexo, as arestas que tornam esse subgrafo conexo são atribuídas e aplica-se o Problema do Carteiro Chinês. A abordagem proposta foi aplicada em duas localidades: um bairro da cidade de Recife e na cidade de Flores, ambas no Estado de Pernambuco. Para cada localidade foram criados cenários e situações, que serviram para realizar a análise de sensibilidade do modelo e permitir inferências. Verificou-se que, para a resolução do PRL em pequenas instâncias o modelo possui tempo de execução razoável, podendo ser aplicado a resolução do PRL em cidades e localidades de pequeno porte.

Palavras-chave: teoria dos grafos; otimização; problema do carteiro chinês; roteamento de leiturista.

ABSTRACT

Meter Reading Problem (MRP) is a combinatorial optimization problem that affects suppliers of natural gas, electricity or water, where companies need to periodically mobilize workers to inspect and bill consumption points. MRP treats the streets of the studied urban network as edges, edges are linked to the distances to be covered, and these edges are also associated with crossing time and time to bill consumption points. MRP seeks to find routes that minimize the distance covered by readers, given a previously assigned workload, this means that the edges endowed with time demands to be crossed and inspected must be assigned to readers in order to respect the established workload. Through a systematic review of the literature carried out in this work, gaps in the literature regarding techniques for solving the MRP were identified, therefore, the present work proposes a cluster first – route second approach to solve the MRP. The proposed approach is applied in two phases, in the first phase the streets of the studied location are grouped into conglomerates through the application of the p -Median Problem, the allocation of the demand of each road or road segment to the conglomerates is limited by the capacity of the reader, then each cluster corresponds to a route. In a second phase, the connectivity of each cluster is verified, if the subgraph of a cluster is not connected, the edges that make this subgraph connected are assigned and the routing is done through the Chinese Postman Problem. The proposed approach was applied in two locations: a neighborhood in the city of Recife and in the city of Flores, both locations are in the State of Pernambuco. For each location, scenarios and situations were created, which served to carry out the sensitivity analysis of the model and allow inferences. It was found that for the resolution of the MRP in small instances the model has reasonable execution time, therefore, the proposed approach can be applied in small cities and towns.

Keywords: graph theory; optimization; chinese postman problem; meter reading problem.

LISTA DE FIGURAS

Figura 1 –	Grafo não direcionado (a), direcionado (b) e misto (c).....	20
Figura 2 –	Transformação de um grafo não dirigido (a) em dirigido (b).....	20
Figura 3 –	Grafo G e Grafo F.....	21
Figura 4 –	Grafo G e o subgrafo G'.....	21
Figura 5 –	Trilha afdefb (a) e caminho acbfe (b).....	22
Figura 6 –	Circuito afdefbca (a) e ciclo fbcadef (b).....	23
Figura 7 –	Grafo ponderado.....	23
Figura 8 –	Pontes de Königsberg.....	24
Figura 9 –	Grafo não direcionado do problema das pontes de Königsberg.....	24
Figura 10 –	Grafo não direcionado (a) e matriz de adjacências (b)	26
Figura 11 –	Grafo direcionado (a) e matriz de adjacências (b)	26
Figura 12 –	Grafo não direcionado (a) e matriz de incidência (b)	27
Figura 13 –	Grafo direcionado (a) e matriz de incidência (b)	27
Figura 14 –	Grafo orientado (a) e lista de adjacência (b)	28
Figura 15 –	Estudo do comportamento assintótico	30
Figura 16 –	Conceito de redução polinomial	32
Figura 17 –	Conceito de complexidade	33
Figura 18 –	Fluxograma do processo de revisão sistemática da literatura	49
Figura 19 –	RSL referente ao PRL	52
Figura 20 –	RSL referente ao PCCC	55
Figura 21 –	RSL referente ao PRAC	21
Figura 22 –	Etapas de execução da metodologia desta pesquisa	60
Figura 23 –	Etapas de execução da abordagem cluster first – route second	62
Figura 24 –	Algoritmo de busca em profundidade modificado	67
Figura 25 –	Algoritmo de Dijkstra modificado	67
Figura 26 –	Decalque do espaço urbano	70
Figura 27 –	Decalque do espaço urbano com vértices e arestas (a) e pontos médios (b).....	70
Figura 28 –	Clusters na rede urbana (a) e clusters (b)	73
Figura 29 –	Vista aérea do bairro de Engenho do Meio	76
Figura 30 –	Vista aérea da cidade de Flores	87

LISTA DE TABELAS

Tabela 1 –	Atributos do espaço urbano estudado	71
Tabela 2 –	Clusters e demandas	73
Tabela 3 –	Clusters	74
Tabela 4 –	Relação de casos para o problema do Engenho do Meio	77
Tabela 5 –	Engenho do Meio: cenários e situações	78
Tabela 6 –	Engenho do Meio: gCpMP para os casos 1 e 2	79
Tabela 7 –	Engenho do Meio: gCpMP para os casos 3 e 4	79
Tabela 8 –	Engenho do Meio: gCpMP para os casos 5 e 6	79
Tabela 9 –	Engenho do Meio: gCpMP para os casos 7 e 8	80
Tabela 10 –	Engenho do Meio: DFS modificado para os casos 1 e 2	81
Tabela 11 –	Engenho do Meio: cluster 1 para o caso 1	82
Tabela 12 –	Engenho do Meio: DFS modificado para os casos 3 e 4	82
Tabela 13 –	Engenho do Meio: cluster 2 para o caso 3	82
Tabela 14 –	Engenho do Meio: cluster 2 e 4 para o caso 4	83
Tabela 15 –	Engenho do Meio: DFS modificado para os casos 5 e 6	83
Tabela 16 –	Engenho do Meio: DFS modificado para os casos 7 e 8	84
Tabela 17 –	Engenho do Meio: cluster 2 para o caso 7	85
Tabela 18 –	Engenho do Meio: cluster 2 para o caso 8	85
Tabela 19 –	Engenho do Meio: PCC para os casos 1 e 2	85
Tabela 20 –	Engenho do Meio: PCC para os casos 3 e 4	86
Tabela 21 –	Engenho do Meio: PCC para os casos 5 e 6	86
Tabela 22 –	Engenho do Meio: PCC para os casos 7 e 8	86
Tabela 23 –	Relação de casos para o problema de Flores	88
Tabela 24 –	Flores 1: cenário e situações	89
Tabela 25 –	Flores 1: gCpMP para o caso 1	89
Tabela 26 –	Flores 1: gCpMP para o caso 2	90
Tabela 27 –	Flores 1: gCpMP para o caso 3	90
Tabela 28 –	Flores 1: gCpMP para o caso 4	90
Tabela 29 –	Flores 1: DFS modificado para o caso 1	91
Tabela 30 –	Flores 1: DFS modificado para o caso 2	91
Tabela 31 –	Flores 1: DFS modificado para o caso 3	91

Tabela 32 –	Flores: cluster 7 para o caso 3	91
Tabela 33 –	Flores: DFS modificado para o caso 4	92
Tabela 34 –	Flores: cluster 2 para o caso 4	92
Tabela 35 –	Flores 1: PCC para o caso 1	92
Tabela 36 –	Flores 1: PCC para o caso 2	93
Tabela 37 –	Flores 1: PCC para o caso 3	93
Tabela 38 –	Flores 1: PCC para o caso 4	93
Tabela 39 –	Flores 2: cenário e situações	94
Tabela 40 –	Flores 2: gCpMP para o caso 1	95
Tabela 41 –	Flores 2: gCpMP para o caso 2	95
Tabela 42 –	Flores 2: gCpMP para o caso 3	95
Tabela 43 –	Flores 2: gCpMP para o caso 4	96
Tabela 44 –	Flores 2: DFS modificado para o caso 1	96
Tabela 45 –	Flores 2: cluster 2 para o caso 4	96
Tabela 46 –	Flores 2: DFS modificado para o caso 2	97
Tabela 47 –	Flores 2: DFS modificado para o caso 3	97
Tabela 48 –	Flores 2: DFS modificado para o caso 4	97
Tabela 49 –	Flores 2: cluster 2 para o caso 4	98
Tabela 50 –	Flores 2: PCC para o caso 1	98
Tabela 51 –	Flores 2: PCC para o caso 2	98
Tabela 52 –	Flores 2: PCC para o caso 3	98
Tabela 53 –	Flores 2: PCC para o caso 4	99

LISTA DE ABREVIATURAS E SIGLAS

A	Conjunto de Arcos de um Grafo
ARP	Arc Routing Problem
CARP	Capacitated Arc Routing Problem
CpMP	Capacitated p -Median Problem
DFS	Depth-First Search
E	Conjunto de Arestas de um grafo
G	Grafo
GAP	Generalized Assignment Problem
$gCpMP$	generic Capacitated p -Median Problem
GRASP	Greedy Randomized Adaptive Search Procedure
HGA	Hybrid Genetic Algorithm
MRP	Meter Reader Problem
N	Conjunto de nós/vértices de um grafo
NP	Nondeterministic polynomial time
NP - completo	Nondeterministic polynomial time complete
NP - difícil	Nondeterministic polynomial time hard
NRP	Node Routing Problem
P	Polynomial
pMP	p -Median Problem
PCC	Problema do Carteiro Chinês
PCCC	Problema do Carteiro Chinês Capacitado
PLIM	Programação Linear Inteira Mista
PR	Path-Relinking
PRA	Problemas de Roteamento em Arcos
PRAC	Problema de Roteamento de Arcos Capacitados
PRACA	Problema de Roteamento de Arcos Capacitados Abertos
PRL	Problema de Roteamento de Leituristas
PRN	Problemas de Roteamento em Nós
PRV	Problema de Roteamento de Veículos
RFID	Radio Frequency Identification
RSL	Revisão Sistemática da Literatura

SOCAL	Southern California Gas Company
V	Conjunto de Vértices de um grafo
VND	Variable Neighborhood Descent

SUMÁRIO

1	INTRODUÇÃO	14
1.1	DESCRIÇÃO DO PROBLEMA	14
1.2	JUSTIFICATIVAS	16
1.3	OBJETIVOS GERAL E ESPECÍFICOS	17
1.3.1	Objetivo geral	17
1.3.2	Objetivos específicos	17
1.4	ESTRUTURA DA DISSERTAÇÃO	18
2	FUNDAMENTAÇÃO TEÓRICA	19
2.1	TEORIA DOS GRAFOS	19
2.1.1	Conceitos	19
2.1.2	Grafos eulerianos	23
2.1.3	Estruturas de dados para grafos	25
2.1.3.1	Matriz de adjacência	25
2.1.3.2	Matriz de incidência	26
2.1.3.3	Lista de adjacências	27
2.2	PROBLEMAS DE ROTEAMENTO	28
2.2.1	Complexidade computacional	29
2.2.2	Problema de roteamento de arcos	33
2.2.3	Problema do carteiro chinês	34
2.2.3.1	Problema do carteiro chinês para grafos não direcionados	34
2.2.3.2	Problema do carteiro chinês para grafos direcionados	36
2.2.3.3	Problema do carteiro chinês para grafos mistos	37
2.2.4	Roteamento de arcos capacitados	38
2.2.4.1	Problema do carteiro chinês capacitado	39
2.2.5	Problema de roteamento de leiturista	41
2.3	PROBLEMA DAS p –MEDIANAS	42
2.4	FÓRMULA DE HARVERSINE	46
3	REVISÃO SISTEMÁTICA DE LITERATURA	48
3.1	ABORDAGENS PARA RESOLUÇÃO DO PROBLEMA DE ROTEAMENTO DE LEITURISTAS	51

3.2	ABORDAGEM CLUSTER FIRST - ROUTE SECOND E PROBLEMA DO CARTEIRO CHINÊS	55
3.3	PROBLEMA DE ROTEAMENTO DE ARCOS E PROBLEMA DA p -MEDIANAS.....	58
4	METODOLOGIA	60
4.1	CLASSIFICAÇÃO DA PESQUISA	61
4.2	ABORDAGEM PARA ROTEAMENTO PROPOSTA	61
4.2.1	Determinar o espaço urbano e carga de trabalho	62
4.2.2	Tratar o espaço urbano como grafo	62
4.2.3	Determinar demanda de cada aresta	63
4.2.4	Definir o número de leitores	64
4.2.5	Determinar matriz de distâncias	64
4.2.6	Aplicar o problema da p-medianas	64
4.2.7	Verificar conectividade	66
4.2.8	Realizar roteamento	68
4.3	MATERIAIS	68
5	ABORDAGEM PROPOSTA	70
6	RESULTADOS E DISCUSSÕES	75
6.1	ENGENHO DO MEIO.....	76
6.2	FLORES 1	87
6.3	FLORES 2	93
6.4	DISCUSSÕES	99
6.4.1	Discussão resultados	99
6.4.2	Discussão revisão sistemática de literatura	102
7	CONCLUSÃO	103
	REFERÊNCIAS	105
	APÊNDICE A – DADOS ENGENHO DO MEIO	115
	APÊNDICE B – DADOS FLORES	120

1 INTRODUÇÃO

Nesse capítulo o problema é contextualizado envolvendo sua descrição, justificativa, objetivos e estruturação do trabalho.

1.1 DESCRIÇÃO DO PROBLEMA

Conforme Slack, Jones e Johnston (2016) a função central de qualquer empreendimento é a produção de bens e/ou serviços, com preços competitivos, que satisfaçam clientes, consumidores ou usuários. Para tanto, organizações requerem de seus sistemas produtivos uma eficiente capacidade de processamento, equilibrando a oferta de bens/serviços com elevado padrão de qualidade e a diminuição de custos operacionais. Haja vista que, o conceito de eficiência e a necessidade de melhoramento constante afeta organizações de caráter público ou privado, com ou sem fins lucrativos, de pequeno ou grande porte.

A modernização de setores ligados ao fornecimento de gás natural, energia elétrica ou água encanada passa pela instalação de medidores eletrônicos que possibilitem medição do consumo e faturamento à distância. Usberti, França e França (2012a) destacam que apesar de a tecnologia de telemedição ser eficiente e estar presente em economias desenvolvidas e algumas localidades ao redor do mundo, a substituição dos medidores depende de altos investimentos e demanda tempo e planejamento. Por outro lado, a força de trabalho humana que realiza a medição do consumo no Brasil mantém-se relativamente barata. Logo, o Problema de Roteamento de Leituristas (PRL) continua atual e extremamente relevante, visto que a atividade de leitura de consumo é realizada periodicamente, as unidades consumidoras estão espalhadas geograficamente ao longo de um território e muitas concessionárias não possuem uma política que estabeleça rotas para verificação do consumo, de modo a otimizar o caminho percorrido pelos leituristas.

Segundo estimativas do Instituto Brasileiro de Geografia e Estatística publicadas no Diário Oficial da União (BRASIL, 2019), o Brasil tinha em 1º de julho de 2019 aproximadamente 210.147.125 habitantes, um incremento de 0,79% em relação a 1º de julho de 2018. Esse crescimento representa em números absoluto um aumento de 1.652.225 pessoas. O crescimento da população impacta diretamente a necessidade de moradia, na abertura de novas vias e no acesso a serviços básicos, como energia elétrica, água e saneamento. A construção de moradia e a abertura de vias, por sua vez, gera a criação de novos pontos de

leitura de água e energia elétrica que podem necessitar de vistoria humana por não serem dotados de leitura remota.

A escolha de uma rota ótima para leitura ou o mais próximo possível disso seria preponderante para a minimização de custos ligados à contratação de pessoal, além de contribuir para a eficiência no planejamento das operações e faturamento do consumo. Trabalhos publicados por Usberti, França e França (2012a), Eglese, Golden e Wasil (2015) e Corberán et al. (2020) apontam o baixo número de pesquisas e trabalhos relacionados ao roteamento de leituristas presentes na literatura. Usberti, França e França (2012a), por sua vez, especificam a necessidade de desenvolvimento de mais estudos na área e a investigação de lacunas presentes na literatura.

Conforme Corberán et al. (2020) problemas de roteamento atraíram a atenção de muitos pesquisadores e profissionais nos últimos 60 anos, devido à sua grande importância econômica e os desafios matemáticos envolvidos em seu estudo e solução.

Segundo Thomaz et al. (2020), os problemas de roteamento podem ser classificados como sendo referentes a nós ou a arcos e arestas. Os Problemas de Roteamento em Arcos (PRAs) abrangem uma série de variações derivadas do Problema do Carteiro Chinês (PCC).

O PCC clássico consiste em encontrar a rota, com o menor custo, que percorra todos os arcos ou arestas de um grafo ao menos uma vez (FOURNIER; SCARSINI, 2019). O PCC sob determinadas restrições referentes ao número de entidades utilizadas para o roteamento e a capacidade dessas entidades pode ser denominado Problema do Carteiro Chinês Capacitado (PCCC) (ORLIS et al., 2020). Usberti et al. (2008) demonstra, que o PCCC é um problema *NP – difícil*, ou seja, não oferece solução satisfatória em tempo polinomial e o PRL pode ser tratado como PCCC quando tem-se um conjunto de leituristas com restrição de carga de trabalho.

Conforme Prins, Labadi e Reghioi (2009), PRAs podem ser resolvidos por quatro tipos de abordagens: Inserção, Mesclagem, *cluster first - route second* ou *route first - cluster second*.

Segundo Dror (2012) métodos *cluster first - route second* são ferramentas consolidadas na literatura para resolver problemas de otimização. Abordagens *cluster first - route second* podem ter em sua fase de agrupamento o Problema das *p*-Medianas. A Aplicação do Problema das *p*-Medianas para o agrupamento de elementos garante *clusters* com excelentes taxas de classificação (KLASTORIN, 1985; BLANCHARD; ALOISE; DESARBO, 2012), além de oferecer robustez no tratamento de *outliers* (KAUFMAN; ROUSSEEUW 2005).

Como lacunas, identificadas após a Revisão Sistemática de Literatura, pode-se destacar o baixo número de publicações referentes ao PRL e a inexistência de um método baseado no Problema das *p*-Medianas para agrupar e posteriormente roteirizar leituristas, garantindo que

cada agrupamento é conexo. Diante disso, o presente trabalho propõe um procedimento baseado em uma abordagem *cluster first - route second* (agrupamento primeiro e roteamento depois) para a resolução do PRL e do PCCC associado ao problema. As ruas da localidade estudada são divididas em segmentos e tratadas como grafo não direcionado. Aos segmentos estão associadas demandas de vistoria expressas pela existência de pontos de leitura e custos de atravessamento expressos pela distância e tempo requerido para percorrer o segmento a pé. No procedimento proposto, a clusterização dos segmentos ocorre por meio da aplicação inicial do Problema das p -Medianas Capacitado, na fase de roteamento, verifica-se a conectividade do subgrafo associado a cada *cluster*. Caso o subgrafo de um *cluster* não seja conexo, acrescenta-se a aresta ou as arestas com peso mínimo que o tornam conexo. Dado o subgrafo de um *cluster*, realiza-se o roteamento do leitorista encontrando-se um circuito euleriano que inicia e termina em um mesmo vértice, para tanto aplica-se o Problema do Carteiro Chinês.

A abordagem proposta diferencia-se das publicações anteriores presentes na literatura para o tratamento do PRL pelo fato dos agrupamentos não possuírem um depósito aleatório em comum de onde as entidades que realizam o roteamento devem partir, dessa forma uma entidade não é obrigada a sair de um depósito e percorrer arestas desnecessárias para alcançar as arestas onde deve realizar o roteamento e depois retornar ao depósito.

Cabe salientar que a abordagem proposta incorpora características do modelo de PCCC de Golden e Wong (1981) como demanda positiva em todas as arestas, o percurso atribuído a cada leitorista tem o mínimo custo possível e as arestas do grafo são percorridas pelo menos uma vez por um conjunto de k leitoristas idênticos cada qual dotado de uma capacidade conhecida Q fixa.

1.2 JUSTIFICATIVAS

Em termos econômicos o roteamento de leitoristas possibilita a diminuição dos custos operacionais referentes à verificação do consumo de energia elétrica. A partir de uma conversa com o responsável pelo setor de planejamento da vistoria dos pontos de consumo atendidos pela empresa que realiza a distribuição de energia elétrica no Estado de Pernambuco, percebeu-se a inexistência de rotas de leitura padronizadas e a necessidade de otimização desse serviço. A conversa citada ocorreu em março de 2020, sendo comum que o processo de leitura dos relógios ocorra de forma empírica. Isso se deve à escassez de informações sobre técnicas que podem ser utilizadas para a otimização do roteamento por parte dos setores ligados ao planejamento da empresa.

De uma perspectiva científica, destaca-se o baixo número de trabalhos acadêmicos ligados ao roteamento de leituristas e a proposição de uma abordagem *cluster first - route second* que combina um problema exato clássico de otimização combinatória com algoritmos de otimização em grafos para a resolução do PRL.

De um ponto de vista social, rotas otimizadas garantem um percurso aprimorado e a exposição de menos funcionários a radiação solar.

Em termos ambientais, a delegação de menos funcionários para inspeção significa menos custo de deslocamento para as áreas onde ocorrerão a vistoria e consequentemente menos consumo de combustíveis fósseis.

1.3 OBJETIVOS GERAL E ESPECÍFICOS

Neste tópico são especificados o objetivo geral e os objetivos específicos.

1.3.1 Objetivo geral

Propor uma abordagem de duas fases, *cluster first - route second*, para a resolução do Problema de Roteamento de Leituristas, e consequentemente, o solucionamento do Problema do Carteiro Chinês Capacitado.

1.3.2 Objetivos específicos

Os itens abaixo mostram os objetivos específicos do trabalho:

- a) agrupar as arestas do grafo não direcionado associado ao problema a ser resolvido mediante a aplicação do Problema das p -Medianas Capacitado;
- b) verificar a conectividade de cada agrupamento por meio de um Algoritmo de Busca;
- c) aplicar o Algoritmo de Dijkstra para identificar, dado um agrupamento desconexo, as arestas que o tornam conexo;
- d) aplicar o Problema do Carteiro Chinês em cada *cluster* conexo;
- e) implementar a abordagem proposta em casos reais;
- f) avaliar a abordagem proposta em um bairro da cidade de Recife e na cidade de Flores no interior de Pernambuco.

1.4 ESTRUTURA DA DISSERTAÇÃO

O presente trabalho tem 7 capítulos sendo eles: Introdução, Fundamentação Teórica, Revisão Sistemática de Literatura, Metodologia, Abordagem Proposta, Resultados e Discussões, e Conclusão. Abaixo cada capítulo será rapidamente.

Introdução: este capítulo traz uma introdução sobre a pesquisa a ser realizada.

Fundamentação Teórica: oferece uma contextualização sobre assuntos recorrentes necessários para o entendimento e embasamento da dissertação.

Revisão Sistemática de Literatura: busca realizar uma análise sobre publicações na área de estudo e identificar lacunas na literatura que justifiquem o desenvolvimento da dissertação.

Metodologia: traz os métodos e ferramentas utilizadas para alcançar os objetivos especificados.

Abordagem Proposta: contextualiza a abordagem proposta e oferece um exemplo simples de sua aplicação.

Resultados e Discussões: aplica a abordagem proposta em dois casos reais, em um bairro de pequeno porte na cidade de Recife - PE e na cidade de Flores - PE.

Conclusão: finaliza o trabalho e apresenta considerações sobre a pesquisa.

2 FUNDAMENTAÇÃO TEÓRICA

No presente capítulo serão apresentados os principais conceitos e notações referentes a Teoria dos Grafos e suas propriedades, assim como a caracterização do Problema de Roteamento de Arcos, do Problema do Carteiro Chinês, do Problema do Carteiro Chinês Capacitado, do Problema de Roteamento de Leituristas e do Problema das p -Medianas Capacitado. Serão também realizadas considerações sobre a complexidade computacional associada a algoritmos e problemas matemáticos.

2.1 TEORIA DO GRAFOS

A seguir a Teoria dos Grafos será contextualizada.

2.1.1 Conceitos

Segundo Latora, Nicosia e Russo (2017) grafos são instrumentos matemáticos utilizados para representar conexões e redes.

Um grafo pode ser representado por $G = (N, L \cup A)$, onde $N(G)$ é o conjunto finito e não vazio de vértices ou nós de G , $L(G)$ corresponde ao conjunto de arestas que formam os pares ordenados não direcionados dos vértices de G , e $A(G)$, por sua vez, representa o conjunto de arcos que formam os pares ordenados direcionados de diferentes vértices do grafo (LATORA; NICOSIA; RUSSO, 2017).

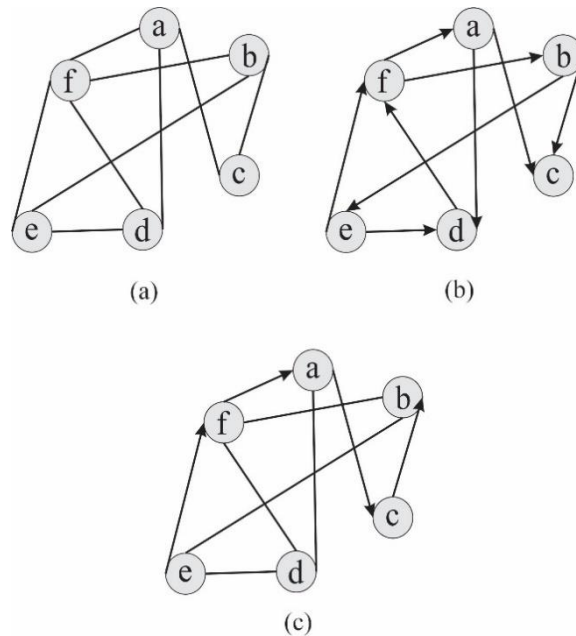
Segundo Latora, Nicosia e Russo (2017) em G uma aresta é constituída por um par ordenado $(i, j) \in L(G)$, sendo i e j os vértices do conjunto $N(G)$ e indicam uma ligação não direcionada entre os vértices. Por outro lado, um arco é formado por um par ordenado $(i, j) \in A(G)$, onde os vértices i e j pertencem ao conjunto $V(G)$ e indicam uma ligação direcionada do vértice i para j .

Quando os vértices i e j se ligam, então eles são conhecidos como adjacentes. Se (i, j) é um arco, então representa a saída do vértice inicial i e a entrada no vértice final. Quando a aresta une um par de nós de i e j com (i, j) ou (j, i) , então ele é não direcionado, uma vez que pode-se percorrer os vértices em ambas as direções (LATORA; NICOSIA; RUSSO, 2017).

Grafos podem ser classificados como dirigidos (a), não dirigidos (b) e mistos (c). A Figura 1 apresenta exemplos dos três tipos de grafos citados. Um grafo dirigido possui todas as arestas direcionadas, um grafo não dirigido possui todas as arestas não direcionadas, um grafo

misto, por sua vez, possui arestas direcionadas (arcos) e não direcionadas (LATORA; NICOSIA; RUSSO, 2017).

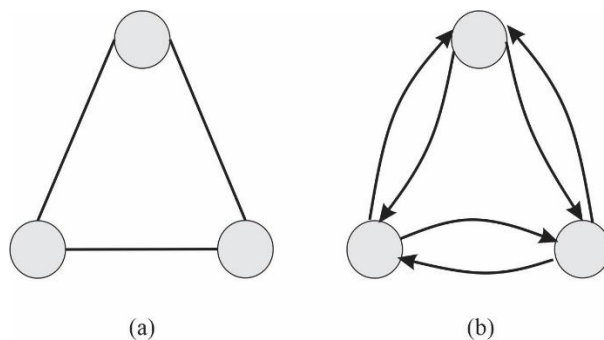
Figura 1 – Grafo não direcionado (a), direcionado (b) e misto (c)



Fonte: O autor (2020).

Em determinadas situações é mais adequado usar grafos direcionados para representar grafos não direcionados. Para tanto, basta substituir o arco não direcionado (i, j) por dois arcos direcionados (i, j) e (j, i) (LATORA; NICOSIA; RUSSO, 2017). A Figura 2 apresenta esta transformação do grafo não dirigido (a) para o dirigido (b).

Figura 2 – Transformação de um grafo não dirigido (a) em dirigido (b)

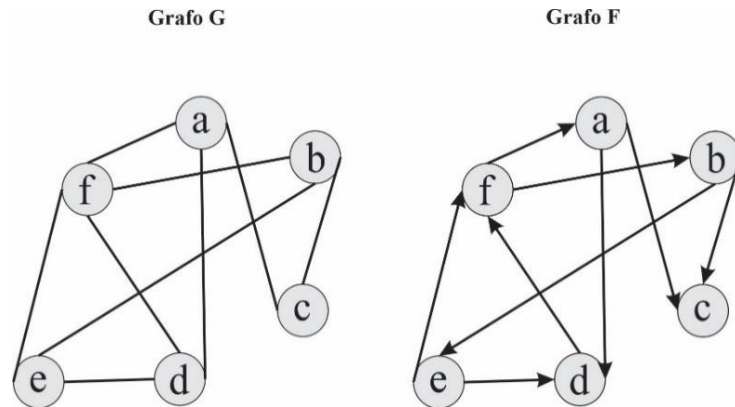


Fonte: O autor (2020).

Para um grafo não dirigido, o grau de um vértice i é expresso por $\deg_G(i)$, que representa o número de arestas incidentes a ele. Em um grafo dirigido, o grau de um vértice qualquer é a soma dos arcos que entram neste vértice (*indegree*) e dos arcos que estão saindo deste vértice

para os demais (*outdegree*) (LATORA; NICOSIA; RUSSO, 2017). A Figura 3 mostra um grafo simples G cujos graus dos vértices são: $\deg_G(a) = \deg_G(b) = 3$, $\deg_G(c) = 2$, $\deg_G(f) = 4$ e um grafo F que tem grau $\deg_G(a) = \deg_{in}(a) + \deg_{out}(a) = 1 + 2 = 3$.

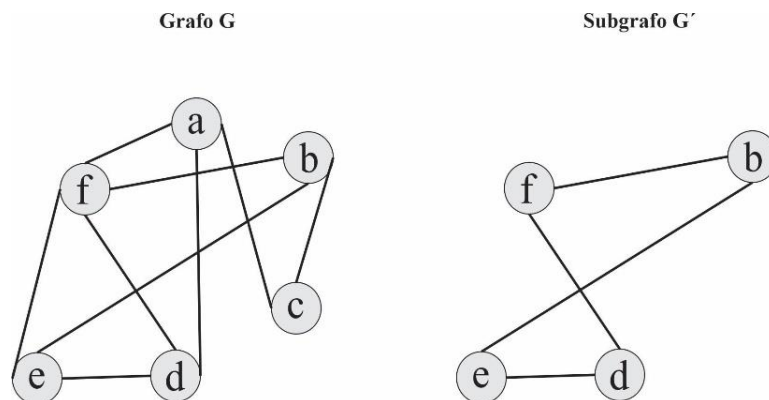
Figura 3 – Grafo G e Grafo F



Fonte: O autor (2020).

Dado um grafo $G = (N, L)$, um subconjunto de vértices e arestas do grafo original é expresso pelo subgrafo $G' = (V', E')$, onde $V' \subseteq N$ e $E' \subseteq L$. Se G' é um subgrafo de G , então G contém G' , conforme a Figura 4 (LATORA; NICOSIA; RUSSO, 2017).

Figura 4 – Grafo G e o subgrafo G'



Fonte: O autor (2020).

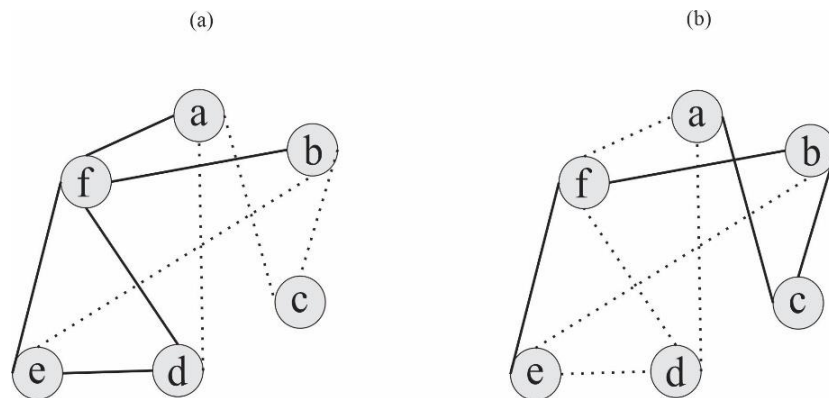
Um grafo $G = (N, L)$ é considerado conexo quando houver para quaisquer dois vértices i e j um caminho conectando i e j , um grafo é considerado desconexo caso contrário (LATORA; NICOSIA; RUSSO, 2017).

Um grafo direcionado G é conexo se o seu grafo não direcionado associado, isto é, o grafo resultante quando a orientação é removida de seus arcos, for conexo. Um grafo direcionado é fortemente conexo se existir um caminho entre todos os pares de nós $(i, j) \in N(G)$ (LATORA; NICOSIA; RUSSO, 2017).

Em um grafo G um passeio $n_1 n_2 \dots n_k$ é uma sequência de vértices não necessariamente distintos $n_1, n_2, \dots, n_k$ tal que $n_i n_{i+1} \in A(G)$, para $1 \leq i \leq k-1$. Diz-se que um passeio é um percurso $n_1 - n_k$ caso os pontos inicial e final são respectivamente, n_1 e n_k (LATORA; NICOSIA; RUSSO, 2017).

Uma trilha em um grafo G é um passeio, tal que, $n_i n_{i+1} \neq n_j n_{j+1}$, $1 \leq i, j \leq k-1$, em uma trilha as arestas não se repetem. Um caminho por sua vez, é um percurso, tal que $n_i \neq n_j$ exceto possivelmente $n_1 = n_k$, dessa forma, em um caminho os vértices são todos distintos, exceto o ponto inicial que pode ser igual ao ponto final (LATORA; NICOSIA; RUSSO, 2017). A Figura 5 mostra a trilha $afdefb$ (a) e o caminho $acbfe$ (b).

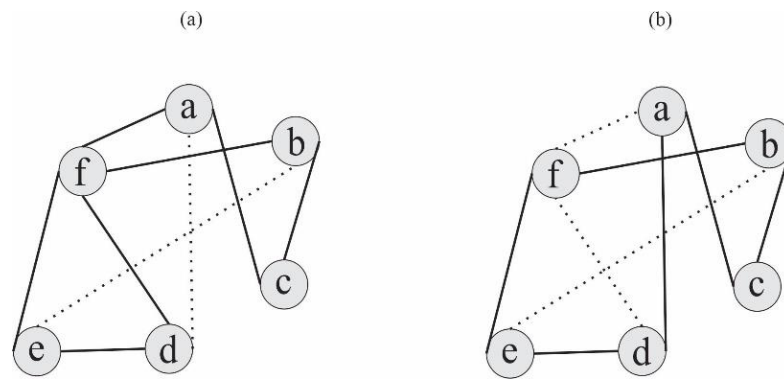
Figura 5 - Trilha $afdefb$ (a) e caminho $acbfe$ (b)



Fonte: O autor (2020).

Um ciclo é um caminho com $n_1 = n_k$, já um circuito é uma trilha com $n_1 = n_k$ (LATORA; NICOSIA; RUSSO, 2017). Na Figura 6, a sequência $afdefbca$ é um circuito (a) e $fbcadef$ é um ciclo (b).

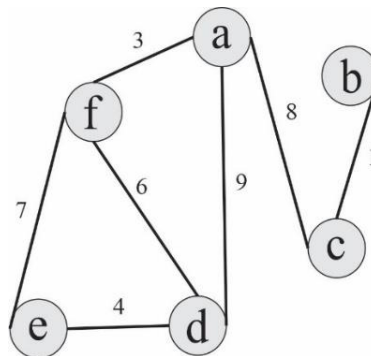
Figura 6 – Circuito afdefbca (a) e ciclo fbcadef (b)



Fonte: O autor (2020).

Um grafo valorado ou grafo ponderado é um grafo que possui funções ou valores relacionando o conjunto de vértices ou o conjunto de arestas (LATORA; NICOSIA; RUSSO, 2017). A Figura 7 mostra um grafo ponderado em suas arestas.

Figura 7 – Grafo ponderado

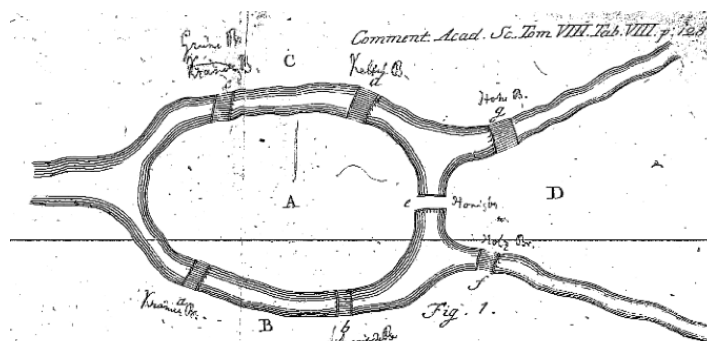


Fonte: O autor (2020).

2.1.2 Grafos eulerianos

A teoria dos grafos teve suas origens em 1736 com os trabalhos de Leonhard Euler referentes ao Problema das Pontes de Königsberg na Prússia, atualmente a cidade é conhecida como Kaliningrad na Rússia. Ao atravessar a cidade, o rio Pregel forma duas ilhas, que foram conectadas ao resto da cidade por sete pontes, conforme a Figura 8 (WEST, 2018). O Problema das Pontes de Königsberg consiste em encontrar um trajeto que percorra todas as sete pontes ao menos uma vez (GROSS, YELLEN, ZHANG, 2013).

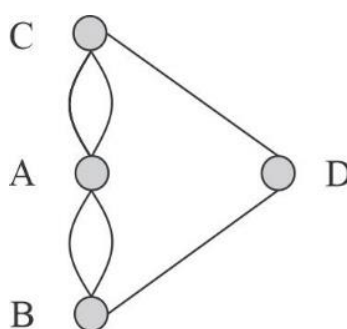
Figura 8 – Pontes de Königsberg



Fonte: University of Pacific (2019).

Euler observou que o Problema das Pontes de Königsberg poderia ser representado por um grafo não direcionado no qual o conjunto de arestas correspondem as sete pontes do problema e os vértices representam as margens do rio e as ilhas. O matemático considerou que a questão seria análoga a encontrar um caminho que percorresse cada aresta do grafo exatamente uma vez, o que passou a ser conhecido como caminho de Euler ou caminho Euleriano, Euler reparou que, para atravessar cada vértice são consumidas exatamente duas arestas, uma para entrar no vértice e outra sair, logo cada nó deve ter um par de linhas, no entanto, no grafo das pontes de Königsberg só existe vértices de grau ímpar e, portanto, o problema não possui solução (JUNGNIKE, 2005). A Figura 9 mostra o grafo associado ao Problema das Pontes de Königsberg.

Figura 9 - Grafo não direcionado do problema das pontes de Königsberg



Fonte: Adaptação Dror (2012).

Embora Euler não tenha resolvido o problema das sete pontes na época, ele ofereceu as condições necessárias para a existência de um grafo Euleriano. Assim, para que um circuito de Euler exista em um grafo, todos os vértices devem ter grau par (Teorema 1) (GROSS, YELLEN, ANDERSON, 2018).

O teorema 1 foi proposto por Euler (1736) para definir a existência de um grafo Euleriano.

Teorema 1 - Um grafo G , fortemente conexo, contém um circuito Euleriano, se, e somente se, o grafo tem apenas vértices de grau par.

Prova: O entendimento do Teorema 1 sobre a existência de um circuito euleriano em um grafo fortemente conectado, refere-se ao fato de poder-se percorrer todos os nós e arestas do grafo partindo de um vértice inicial, a isso adiciona-se o fato de todo circuito Euleriano apresentar cada aresta sendo percorrida exatamente uma vez, isso implica, que haverá uma aresta entrando em um nó i e uma aresta saindo deste nó, logo a incidência das arestas nos nós forma um conjunto de números pares. Ao tratar-se de um grafo direcionado, o *indegree* do nó deve ser igual ao seu *outdegree*, ou seja, a soma dos arcos que entram e saem de um vértice é par. ■

2.1.3 Estruturas de dados para grafos

Segundo Netto (2003) matrizes e listas configuram-se como as formas mais convenientes para a entrada e tratamento de dados, devido a economia e simplicidade em sua representação. Grafos por sua vez, podem ser representados por listas e matrizes para um posterior tratamento ou implementação de algoritmos e modelos matemáticos.

Tanto as matrizes, quanto as listas podem ser classificadas como de adjacência e incidência, dentre as estruturas de dados existentes, as mais utilizadas para representação de dados são a matriz de adjacência, a matriz de incidência e a lista de incidência (GOODRICH; TAMASSI, 2013).

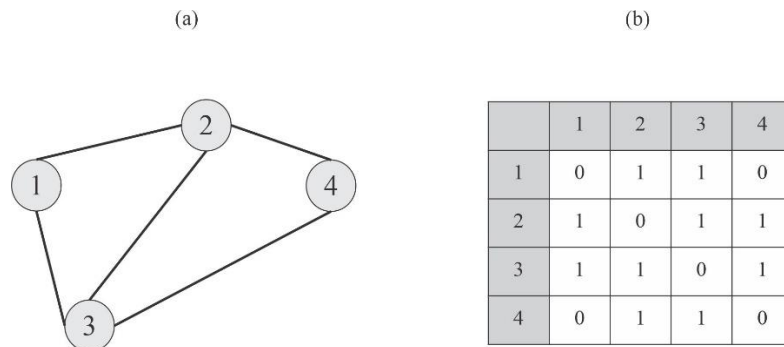
2.1.3.1 Matriz de adjacência

Dado um grafo $G = (V, E)$, uma matriz de adjacência M é formada por n linhas e n colunas, sendo n o número de vértices do grafo e (i, j) os pares ordenados dos vértices da matriz (ASCENCIO; ARAÚJO, 2010). A matriz $M_{n \times n}$ é dada pela Equação 2.1:

$$M_{ij} = \begin{cases} 1, & \text{se } (i,j) \in E(G) \\ 0, & \text{se } (i,j) \notin E(G) \end{cases} \quad (2.1)$$

A Figura 10 exibe um grafo não direcionado (a) e sua matriz de adjacências (b).

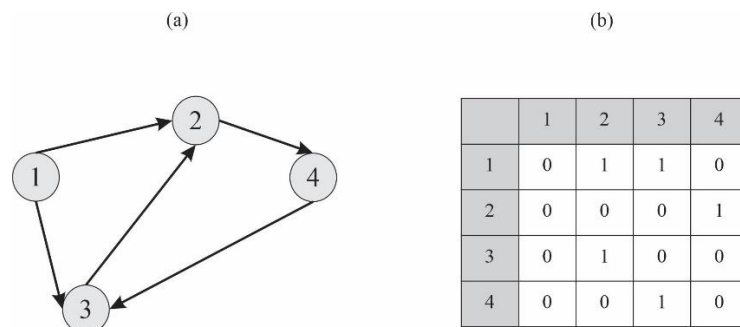
Figura 10 – Grafo não direcionado (a) e matriz de adjacências (b)



Fonte: O autor (2020).

A Figura 11 , por sua vez, representa um grafo direcionado (a) e a matriz de adjacências associada a este grafo (b).

Figura 11 – Grafo direcionado (a) e matriz de adjacências (b)



Fonte: O autor (2020).

2.1.3.2 Matriz de incidência

Segundo Skiena (2008), dado um grafo $G = (V, E)$ de n vértices e m arestas. O conjunto de vértices é expresso por $E(G)$, sendo (i, j) pares ordenados de vértices no grafo G não direcionado, a matriz de incidência de G é denotada por $M_{n \times m}$, e definida por 2.2:

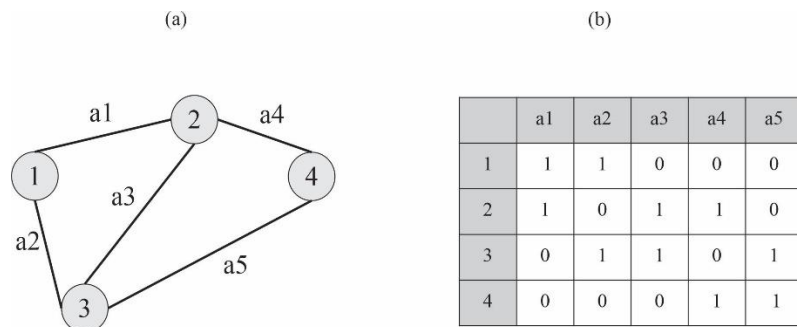
$$M_{ij} = \begin{cases} 1, & \forall i, \text{ se } (i, j) \in E(G) \\ 0, & \text{se } (i, j) \notin E(G) \\ 0, & \text{se } (i, i) \in E(G) \end{cases} \quad (2.2)$$

No caso do grafo direcionado, a matriz de incidência é definida por 2.3:

$$M_{ij} = \begin{cases} 1, & \forall i, \text{ se } (i, j) \in E(G) \\ -1, & \forall j, \text{ se } (i, j) \in E(G) \\ 0, & \text{se } (i, j) \notin E(G) \\ 0, & \text{se } (i, i) \in E(G) \end{cases} \quad (2.3)$$

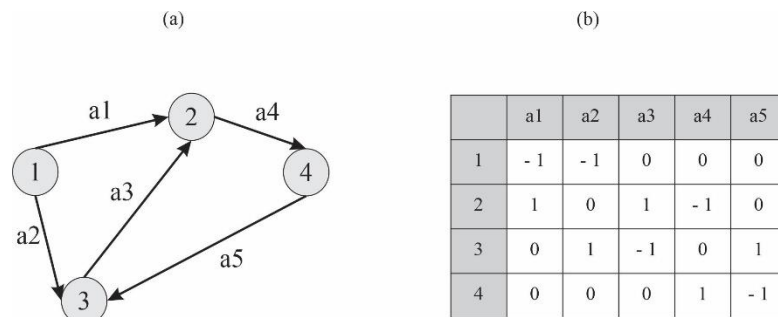
As Figuras 12 e 13 mostram um grafo não direcionado e outro direcionado e as suas respectivas matrizes de incidência associadas.

Figura 12 – Grafo não direcionado (a) e matriz de incidência (b)



Fonte: O autor (2020).

Figura 13 – Grafo direcionado (a) e matriz de incidência (b)



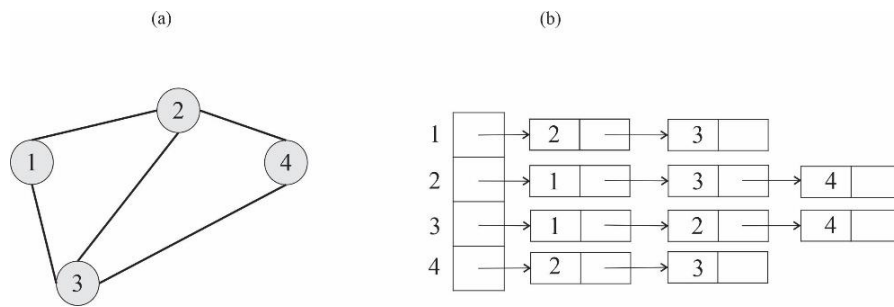
Fonte: O autor (2020).

2.1.3.3 Lista de adjacência

Uma lista de adjacências para um grafo $G = (V, E)$ consiste em um vetor Adj com $n = |V|$ entradas, uma entrada para cada vértice do grafo. Cada entrada, $Adj[v]$ possui listas encadeadas de vértices adjacentes a v em G (GOODRICH; TAMASSIA; 2013).

A Figura 14 mostra um exemplo de grafo não direcionado (a) e sua representação por lista de adjacências (b).

Figura 14 – Grafo orientado (a) e lista de adjacência (b)



Fonte: O autor (2020).

2.2 PROBLEMAS DE ROTEAMENTO

Problemas de roteamento referem-se a problemas de otimização combinatória que buscam encontrar rotas em um grafo (SHERAFAT, 2013). Eles podem ser divididos em duas classes: Problemas de Roteamento em Nós (PRNs) ou Node Routing Problems (NRPs) e Problemas de Roteamento em Arcos (PRAs) ou Arc Routing Problems (ARPs). Na primeira classe, os problemas consistem em determinar uma, ou mais rotas, em que alguns, ou todos os nós de um grafo devem ser visitados. O segundo caso tem por objetivo determinar uma, ou mais rotas, em que alguns, ou todos os arcos ou arestas de um grafo devem ser atendidos (MONROY; AMAYA; LANGEVIN, 2013).

Em Problemas de Roteamento em Arcos e Problemas de Roteamento em Nós, grafos podem ser classificados como eulerianos e hamiltonianos, respectivamente. Os grafos Eulerianos requerem que as todas as arestas sejam percorridas no máximo uma vez sem repetição, os grafos Hamiltonianos por sua vez que todas os vértices sejam percorridos no máximo uma vez sem repetição (BIGGS; LLOYD; WILSON, 1986).

Para a resolução de problemas de roteamento Bodin et al. (1983) e Cunha (2000) classificam e apresentam estratégias de resolução. Bodin et al. (1983) classifica os métodos de resolução como: *cluster first - route second*, *route first - cluster second*, procedimentos de inserção, heurísticas de melhoria, programação matemática, otimização iterativa e procedimentos exatos. Cunha (2000) divide os métodos de solução dos problemas como exatos, heurísticos e metaheurísticos.

Conforme Comert et. al. (2018) uma abordagem de resolução *cluster first - route second* é composta por uma etapa de agrupamento seguida por outra de roteamento, um agrupamento consiste em uma partição de itens em subconjuntos, onde cada subconjunto

corresponde a uma rota, o roteamento representa o sequenciamento de itens a serem visitados em uma rota.

2.2.1 Complexidade computacional

Problemas de Roteamento podem ser modelados e resolvidos computacionalmente por meio de algoritmos e programação matemática de forma exata e heurística (CUNHA, 2000).

Segundo Ziviani (2019) um algoritmo pode ser definido como uma sequência de ações executáveis para a obtenção da solução de um determinado problema. Um método de programação matemática consiste em um sistema de equações com função objetivo, variáveis e restrições que modelam o mundo real (KREYSZIG, 2011).

Para Problemas de Roteamento dá-se o nome de instância ao conjunto de características do problema modelado a ser resolvido, como por exemplo o número de vértices e arcos de um grafo. Conforme Ziviani (2019), o tamanho e demais características ligadas às instâncias impactam diretamente na complexidade computacional do problema.

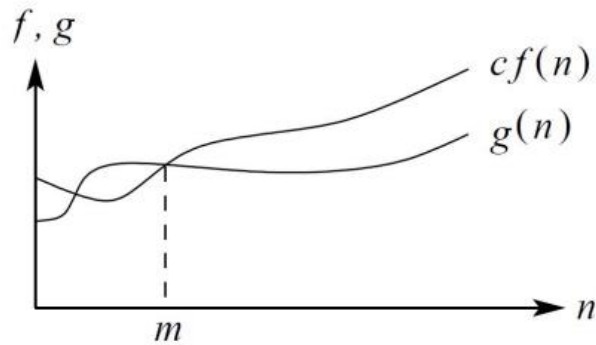
Segundo Cook (1983) o termo complexidade refere-se em geral, aos requerimentos de recursos necessários para que um algoritmo possa resolver um problema computacional, várias medidas de complexidade de um algoritmo surgiram historicamente, no entanto, as mais importantes estão ligadas a medida do tempo e da memória consumida para a resolução do problema.

A complexidade computacional de um algoritmo pode ser medida mediante a análise de seu comportamento assintótico. O comportamento assintótico, por sua vez, pode ser entendido como a curva de crescimento dos recursos consumidos para a resolução de um algoritmo (BOVET; CRESCENZI, 1993). A complexidade assintótica de algoritmos é chamada notação *big-Oh* (ou simplesmente O). A notação *big-Oh* é uma notação universal para se falar de desempenho de algoritmos, que fornece um limite superior para o tempo de execução desse algoritmo.

Segundo Ziviani (2019) a análise do comportamento assintótico inicia-se pelo estabelecimento das funções de custo $f(n)$ e $g(n)$, onde $f(n)$ expressa a medida de espaço de memória ou a medida de tempo necessário para executar um algoritmo para uma instância de tamanho n . Quando queremos medir o tamanho da memória consumida, f é chamada de função de complexidade de espaço do algoritmo, enquanto que, quando deseja-se medir o tempo de execução do algoritmo, f é conhecida como função de complexidade de tempo do algoritmo. Por sua vez, $g(n) = c \times n$, expressa o limite superior para o tempo de execução de um algoritmo,

onde c é uma constante positiva e n o tamanho da instância. A Figura 15 mostra o comportamento das funções f e g durante o estudo da complexidade assintótica das mesmas.

Figura 15 – Estudo do comportamento assintótico



Fonte: Ziviani (2019).

Diz-se que $f(n) = O(g(n))$ caso exista uma constante positiva c tal que $f(n) \leq c \times g(n)$, para qualquer n maior que m . O ponto m é o ponto para o maior valor de n em que $f(n)$ intercepta a curva da função $g(n)$. Caso $g(n)$ não cresça mais rapidamente que $f(n)$, então, $f(n)$ domina assintoticamente $g(n)$ (ABREU, 1987; ZIVIANI, 2019). Como medidas de complexidade temos os valores abaixo:

- a) $f(n) = O(1)$ (complexidade constante). O uso do algoritmo não depende do tamanho de n ; suas instruções são executadas um número fixo de vezes. A complexidade constante é encontrada em algoritmos utilizados para descobrir se um número é par ou ímpar, para verificar se um item em um vetor é nulo ou para imprimir o primeiro elemento de uma lista (CORMEN, et al., 2012).
- b) $f(n) = O(\log n)$ (complexidade logarítmica). A complexidade logarítmica, geralmente, se aplica a algoritmos que dividem os problemas pela metade todas as vezes, como ocorre na busca binária. A busca binária tem por objetivo encontrar o índice de um dado elemento em um vetor ordenado (SKIENA, 2012).
- c) $f(n) = O(n)$ (complexidade linear). Esse tipo de comportamento aparece em algoritmos que realizam um pequeno trabalho em cada elemento de entrada. A complexidade linear está presente em algoritmos que obtêm o valor máximo ou mínimo de um vetor, encontram determinado elemento em uma coleção ou imprimem todos os elementos de uma lista (KOPEC, 2019).
- d) $f(n) = O(n \log n)$ (complexidade $n \log n$). Esse tipo de comportamento ocorre em algoritmos que atacam um problema dividindo-o em problemas menores, resolvendo

cada um deles independentemente, ajuntando em seguida as soluções. Como exemplo de algoritmo com complexidade $(n \log n)$ tem-se o merge sort (SEGEWICK; WAYNE, 2011).

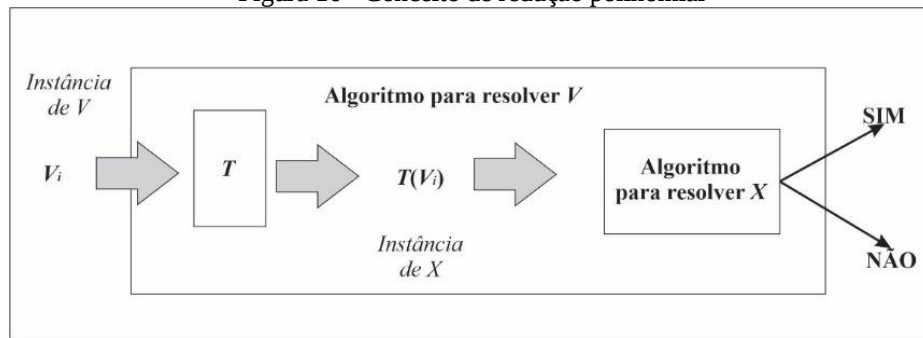
- e) $f(n) = O(n^2)$ (complexidade quadrática). Esse tipo de função ocorre em algoritmos em que os itens de dados são processados aos pares, muitas vezes em um laço dentro do outro. Algoritmos utilizados para verificar se uma coleção possui valores duplicados, algoritmos para ordenar uma coleção de itens por meio de técnicas como o bubble sort, insertion sort ou selection sort, ou algoritmos para encontrar pares ordenados em um vetor tem complexidade quadrática (SKIENA, 2012).
- f) $f(n) = O(n^3)$ (complexidade cúbica). Algoritmos desta ordem de complexidade são utilizados apenas para resolver pequenos problemas. A complexidade cúbica é encontrada na multiplicação de duas matrizes (CORMEN, et al., 2012).
- g) $f(n) = O(2^n)$ (complexidade exponencial). Esse tipo de função ocorre em algoritmos que utilizam força bruta para resolver problemas. A sequência de Fibonacci e a resolução do Problema do Caixeiro Viajante usando programação dinâmica tem complexidade exponencial (SEGEWICK; WAYNE, 2011).
- h) $f(n) = O(n!)$ (complexidade fatorial). Esse tipo de comportamento aparece em algoritmos que listam todas as possíveis combinações de elementos de uma coleção para escolher a melhor, como o método de enumeração completa utilizado no Problema do Caixeiro Viajante.

Um problema pode ser classificado como fácil ou difícil, caso ele possua ou não, um algoritmo polinomial capaz de resolvê-lo (ZIVIANI, 2019). Um algoritmo, por sua vez, pode ser classificado como bom ou ruim, dependendo se ele possui ou não função de complexidade polinomial.

Dada a complexidade computacional que exigem, os problemas podem ser divididos em problemas de otimização e de decisão (VAN BEVERN et al., 2013). Os problemas de decisão são aqueles cujas soluções requerem apenas uma resposta “sim” e “não”. No entanto, muitos problemas de otimização podem ser expressos como problemas de decisão, logo, se existir um algoritmo em tempo polinomial para o problema de decisão, então também existirá um algoritmo em tempo polinomial para o problema de otimização. Os problemas de decisão são, geralmente, difíceis de resolver (BLUM, L. et al., 1996; WILLIAMSON; SHMOYS, 2011).

Lewis e Papadimitriou (2000) apresentam um conceito de redução polinomial necessária para o entendimento da classificação dos problemas, ver a Figura 16.

Figura 16 - Conceito de redução polinomial



Fonte: Lewis e Papadimitriou (2000).

Pode-se dizer que T é uma redução polinomial do problema V para o problema X quando T transforma, em tempo polinomial, instâncias do problema V em instâncias do problema X , de maneira que V_i é uma instância do problema V com resposta “sim”, se e somente se, $T(V_i)$ for uma instância do problema X que fornece a resposta “sim”. Quando existe uma redução polinomial de V para X , podemos dizer que X é no mínimo mais complexo que V . Se existir um algoritmo polinomial para X , esse método também resolverá V em tempo polinomial. Se V exigir um tempo exponencial, então X exigirá um tempo pelo menos exponencial.

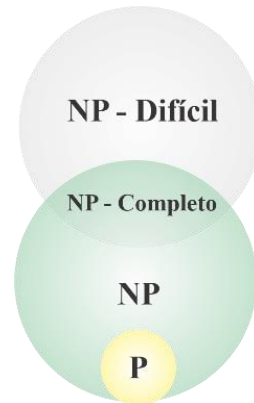
A maioria dos problemas de decisão associados a problemas de interesse prático, derivados ou não de problemas de otimização, pertencem às seguintes classes:

- a) **classe P** : Um problema X pertence à classe P se ele pode ser resolvido por um algoritmo em tempo polinomial. Os Problemas do Carteiro Chinês não Direcionado e do Carteiro Chinês Direcionado são da classe P (PAPADIMITRIOU, 2003).
- b) **classe NP** : Um problema X pertence à classe NP quando sua solução é não verificável em tempo polinomial. O Problema de Isomorfismo de grafos que consiste em determinar se dois grafos finitos são isomórficos pertence a classe NP (SCHÖNING, 1988).
- c) **classe NP -completo**: Um problema X é NP -completo quando está em NP , e para qualquer V em NP , um algoritmo para resolver X pode ser utilizado em tempo polinomial para resolver esse problema. Exemplos de problemas da classe NP -completo são o Problema de Roteamento de Veículos, o Problema da Mochila e o Problema do Caminho Hamiltoniano em um grafo (GOETSCHALCKX; JACOBS-BLECHA, 1989; GUREVICH; SHELAH, 1987).
- d) **classe NP -difícil (NP -árduo ou NP -hard)**: Um problema X pertence à classe NP -difícil se qualquer outro problema V pertencente à classe NP puder ser reduzido polinomialmente a X , X não necessariamente pertence à classe NP . Um problema NP -difícil é pelo menos tão difícil quanto qualquer problema em NP . O Problema

das p -Medianas e o Problema do Carteiro Chinês Misto são NP – *difícil* (PAPADIMITRIOU, 2003; KARIV; HAKIMI, 1979).

A Figura 17 mostra como as classes dos problemas estão associadas:

Figura 17 - Conceito de complexidade



Fonte: Esta pesquisa (2020)

2.2.2 Problema de roteamento de arcos

Os problemas de roteamento de arcos têm atraído a atenção de muitos pesquisadores e profissionais nos últimos 60 anos, devido ao seu grande impacto econômico e aos desafios matemáticos envolvidos em seu estudo e solução (CORBERÁN, et al. 2020).

Apesar da expressão arco referir-se a arestas dirigidas, a literatura convencionou o termo Problema de Roteamento de Arcos para designar um conjunto amplo de problemas onde deseja-se percorrer as arestas de um grafo dirigido, não dirigido ou misto ao menos uma vez, sob as mais diversas condições. Segundo Van Bevern et al. (2015) os Problemas de Roteamento em Arcos podem ser de três tipos:

- a) **Problema do Carteiro Chinês:** onde requer-se que todas as arestas ou arcos sejam percorridos pelo menos uma vez por uma entidade;
- b) **Problema do Carteiro Chinês Rural:** este problema generaliza o Problema do Carteiro Chinês e requer que apenas um conjunto de arestas/arcos seja percorrido dentre todas/os arestas/arcos do grafo;
- c) **Problema de Roteamento de Arcos Capacitados:** é um problema mais geral, além de permitir que mais de um veículo seja utilizado para atravessar as arestas, as entidades que realizam esse atravessamento possuem alguma restrição de capacidade.

O PRA iniciou com os primeiros trabalhos referentes a grafos publicados por Euler no século XVIII, e vem evoluindo com a aplicação de novas modelagens e a adição de avanços tecnológicos relevantes como identificação por radiofrequência (RFID), a disponibilidade de dados em tempo real, a geolocalização e o fluxo de tráfego.

2.2.3 Problema do carteiro chinês

Apesar de Euler demonstrar que, para existir um circuito fechado que percorra todas as arestas de um grafo não direcionado, ao menos uma vez, todos os vértices devem ter grau par, ele não se preocupou com o custo associado à distância percorrida no percurso. No entanto, na década de 1960 o matemático chinês Meigu Guan (Mei-Ko Kwan) introduziu o que ficou conhecido como Problema do Carteiro Chinês (PCC). No problema introduzido por Kwan (1962) um carteiro tem um conjunto de segmentos de ruas associados a ele. O PCC consiste em encontrar um trajeto que permita que um carteiro saindo de uma agência percorra todas as ruas designadas ao menos uma vez e retorne à agência de origem.

Conforme Ahuja, Magnant e Orlin (1993) na Teoria dos Grafos, o PCC é um problema no qual, dado um grafo $G = (V, A)$ cujos arcos (i, j) têm um comprimento não negativo c_{ij} , deseja-se identificar um caminho de comprimento mínimo que inicie em um dado vértice, passe por todos os arcos da rede pelo menos uma vez e retorne ao nó inicial.

A depender das características do problema a ser modelado o PCC pode possuir as características de um grafo não direcionado, direcionado e misto. Segundo Papadimitriou (2003), enquanto o PCC não direcionado e o PCC direcionado podem ser resolvidos em tempo polinomial, o PCC misto é *NP - difícil*.

Para grafos não direcionados, direcionados e mistos certas condições necessárias e suficientes foram estabelecidas para que um grafo euleriano ocorra, embora condições necessárias tenham sido propostas por Euler, condições suficientes foram expostas em estudos posteriores (EISELT; GENDREAU; LAPORTE, 1995; GODINHO FILHO; JUNQUEIRA, 2006).

2.2.3.1 Problema do carteiro chinês para grafos não direcionados

Seja $G = (V, E)$ um grafo não direcionado cujas arestas tenham pesos conhecidos para todo $(i, j) \in E(G)$. Um grafo não direcionado conexo G é Euleriano se e somente se todo vértice em G tem o mesmo grau e G for a união disjunta de ciclos.

Uma vez determinado que um grafo é Euleriano, a solução para se encontrar os circuitos Eulerianos neste grafo é trivial, como pode-se ver com o algoritmo de Fleury, mostrado em Kauffman (1967), o algoritmo de Larson e Odoni (1981) e o algoritmo de Edmonds e Johnson (1973). Segundo Konowalenko et al. (2012) o algoritmo para a resolução do Problema do Carteiro Chinês não Direcionado, e descoberta de um circuito Euleriano em grafos não direcionados, consiste em percorrer todas as arestas, partindo de um vértice qualquer, apagando cada aresta percorrido e nunca percorrendo uma aresta que divida o grafo em dois subgrafos conexos separados. Porém, por vezes um grafo não apresenta inicialmente um circuito de Euler. Para estes casos, Kwan (1962) e Larson e Odoni (1981) observam que um grafo não direcionado e conectado $G = (V, E)$ tem sempre um número par de nós de grau ímpar e que, portanto, o grafo $G(V, E)$ pode ser transformado em um grafo $G'(V, E')$ através da inclusão de arcos artificiais duplicados, que transformem $G(V, E)$ em um grafo Euleriano.

Ahuja, Magnant e Orlin (1993) oferece um modelo matemático para a resolução do Problema do Carteiro Chinês Direcionado (PCCND) expresso entre as Equações (2.4) – (2.7):

$$\text{Minimize } \sum_{(i,j) \in E} c_{ij} x_{ij} \quad (2.4)$$

Sujeito a

$$\sum_{(i,j) \in E} x_{ij} - \sum_{(i,j) \in E} x_{ji} = 0, (i = 1, \dots, n) \quad (2.5)$$

$$x_{ij} + x_{ji} \geq 1, \forall (i,j) \in E \quad (2.6)$$

$$x_{ij} \text{ inteiro}, \forall (i,j) \in E \quad (2.7)$$

A variável x_{ij} representa o número de vezes que a aresta (i, j) deverá ser percorrida do vértice i para o vértice j durante o *tour*. O peso da aresta (i, j) é representado por c_{ij} , que representa o custo de deslocamento. A função objetivo (2.4) minimiza o custo total do *tour* do carteiro. As restrições (2.5) asseguram a conservação do fluxo, ou seja, que o número de arestas que entram no vértice i será igual ao número de arestas que irá deixá-lo; (2.6) garantem que nenhuma aresta deixará de ser percorrida, seja de i para j ou vice-versa; e (2.7) indica que x_{ij} pertence ao conjunto dos números inteiros.

2.2.3.2 Problema do carteiro chinês para grafos direcionados

Para o caso direcionado, tem-se um grafo direcionado $G = (V, A)$ que possui arcos com pesos não negativos c_{ij} para todo $(i, j) \in A(G)$. Um grafo G direcionado fortemente conexo é considerado Euleriano se e somente se todo vértice em G forem simétricos, ou seja, para qualquer vértice, o número de arcos direcionados que “*entram*” é igual ao número de arcos que “*saem*”. Segundo Konowalenko et al. (2012) quando em alguns vértices o número de arcos de entrada diverge do número de arcos de saída, o grafo não é unicursal e, para torná-lo é necessário o acréscimo de cópias apropriadas de alguns arcos.

Assim como nos grafos não direcionados, o primeiro passo para a resolução do problema CPP em redes direcionadas é verificar se o grafo possui um circuito Euleriano. Para isto, pode-se utilizar uma versão modificada do teorema de Euler, apresentado em Eiselt, Gendreau e Laporte (1995). Caso tenha-se certeza da existência de um circuito Euleriano no grafo utiliza-se para sua identificação uma adaptação do algoritmo de Fleury para grafos direcionados encontrado em Christofides (1975), o algoritmo sugerido por Van Aardenne-Ehrenfest e Bruijn (1951), ou mesmo o algoritmo mostrado em Ahuja, Magnant e Orlin (1993).

A resolução do Problema do Carteiro Chinês para Grafos Direcionados e a definição de um circuito euleriano de custo mínimo em grafos não direcionados pode ser construído a partir de um simples Problema de Transporte. Nesse caso, os vértices com excesso de entrada serão considerados como suprimento e os com excesso de saída como demanda. A solução do Problema de Transporte indica qual vértice de suprimento deve ser associado em qual demanda. As cópias dos arcos devem ser acrescentadas ao grafo, ao longo dos caminhos mínimos que ligam os vértices de suprimento aos de demanda na solução do Problema de Transporte (BELTRAMI; BODIN, 1974).

Backer (1983) utilizou o modelo matemático (2.8) – (2.10) para resolução exata do Problema do Carteiro Chinês Direcionado (PCCD) por meio de Programação Linear.

$$\text{Minimize } \sum_{(i,j) \in A} c_{ij} x_{ij} \quad (2.8)$$

Sujeito a

$$\sum_{(i,j) \in A} x_{ij} - \sum_{(i,j) \in A} x_{ji} = 0, \quad (i = 1, \dots, n) \quad (2.9)$$

$$x_{ij} \geq 1 \text{ e inteiro, } \forall (i,j) \in A \quad (2.10)$$

No modelo matemático, a função objetivo (2.8) minimiza o custo total do carteiro em seu percurso direcionado. Quanto as restrições, (2.9) garantem a conservação de fluxo, cada vértice tem seu *indegree* igual ao seu *outdegree* e; (2.10) permitem que cada arco (i, j) seja percorrido pelo menos uma vez em G .

2.2.3.3 Problema do carteiro chinês para grafos mistos

Uma rede mista é representada por um grafo $G = (V, E \cup A)$ com um conjunto de arestas não orientadas $E(G)$, um conjunto de arcos orientados $A(G)$ e pesos c_{ij} em todas as arestas e arcos. Um grafo misto e fortemente conexo G é Euleriano se, e somente se, G for balanceado. Um grafo é balanceado, se as condições de balanceamento e unicursalidade são satisfeitas.

Algoritmos publicados por Eiselt, Gendreau e Laporte (1995) podem ser utilizados para identificar um circuito Euleriano em grafos mistos ou transformar grafos mistos não Eulerianos em grafos mistos Eulerianos.

Edmonds e Johnson (1973) resolve o Problema do Carteiro Chinês em Grafos Mistos substituindo cada arco não direcionado (i, j) por dois arcos direcionados (i, j) e (j, i) . Ao atribuir arbitrariamente uma direção a cada arco não direcionado, o grafo misto, inicialmente tratado, torna-se direcionado. Dado o grafo totalmente direcionado, aplica-se o Problema de Fluxo de Custo Mínimo para a resolução do Problema do Carteiro Chinês.

Ao tornar um arco não orientado (i, j) e substituí-la por dois arcos direcionados (i, j) e (j, i) , o problema muda, agora o carteiro deve viajar pelos dois arcos direcionados (i, j) e (j, i) , antes ele poderia caminhar para frente ou para trás ao longo da aresta não direcionada (i, j) (MINIEKA, 1979).

Utilizando o Problema da Satisfabilidade Boleana, Papadimitriou (1976) mostrou que a versão mista do PCC é *NP - completo*, mesmo se restringir a entrada utilizando um grafo misto planar, com cada vértice com grau máximo três e todos os custos dos arcos iguais a um.

Ahuja, Magnanti e Orlin (1993) propõe um modelo matemático para a resolução do Problema do Carteiro Chinês Mistos, conforme as Equações (2.11) – (2.16) . Onde separa-se os componentes do grafo em dois grupos, um incluindo somente os nós com arcos direcionados, tal que $(i, j) \in A$ e outro contendo apenas os nós compondo as arestas não direcionadas em sua incidência, tal que $(i, j) \in E$.

$$\text{Minimize } \sum_{(i,j) \in A} c_{ij} x_{ij} + \sum_{(i,j) \in E} c_{ij} x_{ij} \quad (2.11)$$

Sujeito a

$$\sum_{(i,j) \in A} x_{ij} - \sum_{(i,j) \in E} x_{ij} = \sum_{(i,j) \in E} x_{ji} - \sum_{(i,j) \in A} x_{ji}, \quad (i = 1, \dots, n) \quad (2.12)$$

$$x_{ij} + x_{ji} \geq 1, \quad \forall (i,j) \in E \quad (2.13)$$

$$x_{ij} \geq 1, \quad \forall (i,j) \in A \quad (2.14)$$

$$x_{ij} \text{ inteiro}, \quad \forall (i,j) \in E \quad (2.15)$$

$$x_{ij} \text{ inteiro}, \quad \forall (i,j) \in A \quad (2.16)$$

O modelo matemático é formulado fazendo x_{ij} ser uma variável que recebe o número de vezes que cada aresta/arco (i, j) será caminhada em G para um custo específico c_{ij} . A função objetivo (2.11) minimiza o custo total do circuito. As restrições: (2.12) garantem a conservação de fluxo; (2.13) forçam cada aresta a ser percorrida pelo menos uma vez durante o *tour*; e (2.14) forçam cada arco a ser percorrido no mínimo uma vez durante o *tour*. As restrições (2.15) e (2.16) apenas indicam que todo x_{ij} pertence ao conjunto dos números inteiros, tanto para arestas quanto para arcos.

O PCC Misto é *NP – difícil*, para problemas com um baixo número de instâncias (aresta/arcos), o PCC Misto pode ser resolvido com facilidade, no entanto, com o aumento do número de instâncias o esforço computacional para a resolução do problema aumenta (EISELT; GENDREAU; LAPORTE, 1995).

2.2.4 Roteamento de arcos capacitados

Segundo Eiselt, Gendreau e Laporte (1995) o Problema de Roteamento de Arcos Capacitados (PRAC) ou Capacitated Arc Routing Problem (CARP) é provavelmente o mais importante problema na área de roteamento em arcos.

O PRAC é um problema de otimização combinatória introduzido por Golden e Wong (1981). O problema é caracterizado por possuir uma demanda não negativa associada a cada arco de um grafo, os veículos que realizam o roteamento devem atravessar os arcos coletando e/ou entregando determinadas demandas sem exceder sua capacidade. A capacidade dos veículos pode ser limitada pela quantidade de carga que eles podem carregar, no entanto, a capacidade também pode estar vinculada à máxima capacidade de trabalho da entidade que

realiza o roteamento ou ainda à distância percorrida por ela, para o problema a frota de veículos são idênticos.

O PRAC pode ser modelado por meio de um grafo $G = (V, E \cup A)$, onde $V(G)$ é o conjunto de nós ou vértices, $E(G)$ o conjunto de arestas e $A(G)$ o conjunto de arcos. Em G existe uma demanda $d_{ij} \geq 0$ e um custo $c_{ij} > 0$ para todo $(i, j) \in E \cup A$, neste contexto, a demanda deverá ser coletada (e entregue) por um conjunto de k veículos com capacidade restrita Q cada, arcos que possuem demanda igual a zero não precisam ser percorridos para o modelo. Três condições são estabelecidas para o roteamento de arcos capacitados: 1) cada veículo inicia e termina seu circuito em um mesmo nó; 2) arestas e arcos com demandas positivas devem ser percorridos ao menos uma vez por um único veículo; 3) a demanda total coletada (e entregue) por um circuito não deve exceder a capacidade máxima do veículo (CORBERÁN; PRINS, 2010; FOULDS; LONGO; MARTINS, 2015; YAO et al., 2017).

Na construção da rota para cada veículo, o custo e a demanda de cada aresta ou arco serão computados. Determinado veículo ao atravessar a primeira vez a aresta ou arco terá os valores de custo e demanda associados a ele, como cada veículo pode percorrer as arestas e arcos mais de uma vez, nas demais vezes que a aresta e arco forem percorridos apenas o custo de deslocamento será computado. As arestas percorridas, onde as demandas e custos são computados, recebem o nome de arestas necessárias, quando ao atravessar uma aresta apenas o custo de atravessamento é computado, essa aresta recebe o nome de *deadhead*. As arestas de *deadhead* são arestas fictícias que são atribuídas aos vértices quando as arestas originais já foram percorridas uma única vez, porém ainda é preciso atravessá-las novamente para concluir o circuito (YAO et al., 2017; BELENGUER; BENAVENT, 2003; CAMPBELL et al., 2018).

Quando todas as demandas das arestas/arcos do PRAC precisam ser atendidas, o problema se reduz a versão capacitada do PCC, conhecido como Problema do Carteiro Chinês Capacitado (PCCC), para esse caso, toda aresta/arco têm demanda positiva ($d_{ij} > 0$ para todo $(i, j) \in E \cup A$).

2.2.4.1 Problema do carteiro chinês capacitado

O PCCC foi inicialmente introduzido por Christofides (1976). O PCCC é um caso especial do PRAC onde todas as arestas/arcos apresentando demanda positiva. Nesta situação, o percurso realizado deve ter o mínimo custo possível e as arestas/arcos do grafo devem ser percorridas pelo menos uma vez por um conjunto de k veículos idênticos cada qual dotado de

uma capacidade conhecida Q . Quando a capacidade de um veículo é maior ou igual a soma de todas as demandas o PCCC pode ser tratado como PCC (GOLDEN; WONG, 1981).

Conforme Golden e Wong (1981) e Dror (2012) tanto o PCCC, quanto o PRAC são NP-difíceis tanto para casos não direcionados, como para casos direcionados por derivarem do Problema de Roteamento de Veículos (PRV) clássico.

Golden e Wong (1981) propôs um modelo de Programação Linear Inteira Mista (PLIM) para a resolução do PCCC de forma ótima. Nele dado um grafo $G = (V, E)$, onde $V(G)$ é o conjunto de vértices e $E(G)$ o conjunto de arestas, o objetivo é encontrar um conjunto de circuitos para um conjunto de K veículos em n vértices, cada veículo possui capacidade W e deve atender a demanda existente em G . Este problema é formulado como um problema de fluxo de tal forma que: c_{ij} representa o comprimento do arco (i, j) ; q_{ij} é a demanda a ser atendida no arco (i, j) ; $^p l_{ij}$ é uma variável binária que assume o valor de 1 quando o veículo p atender a demanda presente em (i, j) , e assume o valor de 0, caso contrário; $^p x_{ij}$ é uma variável binária que assume o valor de 1, se a aresta (i, j) for atravessada pelo veículo p , e 0, caso contrário; $^p f_{ij}$ é uma variável de fluxo que apresenta valor positivo quando $^p x_{ij} = 1$. O PLIM proposto por Golden e Wong (1981) pode ser visto a seguir:

$$\text{Minimize } \sum_{i=1}^n \sum_{j=1}^n \sum_{p=1}^K c_{ij} ^p x_{ij} \quad (2.17)$$

Sujeito a

$$\sum_{k=1}^n ^p x_{ki} - \sum_{k=1}^n ^p x_{ik} = 0, \quad (i = 1, \dots, n); (p = 1, \dots, K) \quad (2.18)$$

$$\sum_{p=1}^K (^p l_{ij} - ^p l_{ji}) = \left\lceil \frac{q_{ij}}{W} \right\rceil, \quad (i, j) \in E \quad (2.19)$$

$$^p x_{ij} \geq ^p l_{ij}, \quad (i, j) \in E; (p = 1, \dots, K) \quad (2.20)$$

$$\sum_{i=1}^n \sum_{j=1}^n ^p l_{ij} q_{ij} \leq W, \quad (p = 1, \dots, K) \quad (2.21)$$

$$\begin{cases} \sum_{k=1}^n ^p f_{ik} - \sum_{k=1}^n ^p f_{ki} = \sum_{j=1}^n ^p l_{ij}, & (i = 2, \dots, n); (p = 1, \dots, K) \\ ^p f_{ij} \leq n^2 ^p x_{ij}, & (i, j) \in E; (p = 1, \dots, K) \\ ^p f_{ij} \geq 0, & (i, j) \in E; (p = 1, \dots, K) \end{cases} \quad (2.22)$$

$$^p x_{ij}, ^p l_{ij} \in \{0, 1\}, \quad (i = 1, \dots, n); (j = 1, \dots, n); (p = 1, \dots, K) \quad (2.23)$$

A função objetivo (2.17) minimiza o comprimento total percorrido pelos veículos. A restrição (2.18) assegura a conservação do fluxo para cada veículo. A restrição (2.19) garante que o atendimento dos veículos será considerado em apenas uma de suas passagens pelo arco (i, j) . A restrição (2.20) garante que a aresta (i, j) será percorrida pelo veículo p somente se ele atravessar o arco (i, j) . A restrição (2.21) permitirá que a capacidade máxima de cada veículo não seja ultrapassada. As restrições (2.22) formam um conjunto de restrições que foram empregadas para assegurar a eliminação de sub rotas ilegais, ou seja, rotas desconexas. E por fim, (2.23) definem as classes das variáveis e parâmetros de entrada.

2.2.5 Problema de roteamento de leiturista

O Problema de Roteamento de Leituristas (PRL) ou Meter Reader Problem (MRP) é um subproblema do PRA. A depender das condições sob as quais é modelado, o PRL pode enquadrar-se como PCC ou como PCCC. O PRL é um PCC quando um leiturista pode percorrer todas as arestas de um dado grafo, passando ao menos uma vez por cada aresta, por outro lado, o PRL é um PCCC quando o percurso realizado deve ter o mínimo custo possível e as arestas do grafo devem ser percorridas pelo menos uma vez por um conjunto de k leituristas idênticos cada qual dotado de uma capacidade conhecida Q

O PRL emerge em concessionárias de fornecimento de energia elétrica, água, gás, ou qualquer serviço que necessite de vistoria para verificação do consumo e emissão do faturamento. Szydlowski (1993) e Quadar et al. (2021) descrevem cinco formas de leitura de consumo conforme a tecnologia empregada:

- a) **leitura manual do medidor:** O leitor usa um livro de rotas e registra manualmente o consumo do medidor;
- b) **leitura com computadores de mão:** O leiturista carrega durante a leitura um dispositivo onde digita o consumo;
- c) **leitura eletrônica remota:** O leiturista ainda carrega o dispositivo de medição, no entanto, não precisa digitar o consumo, apenas aproximar o dispositivo de medição do relógio medidor de consumo das residências;
- d) **leitura por radio frequência:** O leiturista precisa apenas estar minimamente próximo do relógio medidor de consumo para que a leitura ocorra via rádio frequência; e

- e) **leitura automática:** O relógio medidor envia o valor do consumo via internet automaticamente para a concessionária, não sendo necessário o envio de leituristas às residências. Este sistema de leitura pode ser encontrado em operação nos Estados Unidos (U.S. ENERGY INFORMATION ADMINISTRATION, 2019) e na União Européia (EUROPEAN COMMISSION ENERGY, 2019).

Eglese, Golden e Wasil (2015) destacam as principais tecnologias e técnicas utilizadas no Roteamento de Leituristas desde a década de 1970, quando Stern e Dror (1979) publicaram o primeiro trabalho sobre Roteamento Leituristas, nessa década os primeiros esforços concentraram-se no desenvolvimento de algoritmos para encontrar ciclos de Euler para pequenos problemas Eulerianos com menos de 50 vértices e 100 arestas.

Na década de 1990, os dados geográficos ficaram disponíveis na forma de arquivos GBF/DIME. Houve diversos desafios na modelagem, devido aos dados dos arquivos nem sempre estarem compatíveis com os dados geográficos reais. Algoritmos de otimização e recursos interativos tornaram-se disponíveis pela primeira vez em sistemas de informação para roteamento de leituristas. No início dos anos 2000, Sistemas de Informação Geográfica (SIG) foram combinados com algoritmos de roteamento (heurísticos) gerando sistemas computadorizados (EGLESE; GOLDEN; WASIL, 2015).

No final dos anos 2000, com a ampla disponibilidade da Tecnologia de Identificação por Radiofrequência ou Radio Frequency Identification (RFID), uma etiqueta RFID foi anexada ao medidor e a leitura pode ser realizada à distância, dessa forma, um caminhão equipado com um dispositivo de leitura consegue atravessar ruas e coletar dados dos medidores automaticamente. Não é necessário visitar cada medidor individualmente porque um medidor pode ser lido a uma distância pré-definida (EGLESE; GOLDEN; WASIL, 2015). Atualmente, já existem tecnologias que permitem o monitoramento do consumo de energia elétrica via *web*.

Apesar dos avanços tecnológicos e de automação presentes nos processos de leitura de consumo, o Problema de Roteamento de Leituristas continua atual na rotina de concessionárias em diversos países, principalmente em nações subdesenvolvidas ou em desenvolvimento onde a mão-de-obra humana empregada na leitura é relativamente barata comparada à implementação de tecnologias mais eficientes de leitura.

2.3 PROBLEMA DAS p –MEDIANAS

Problemas de localização tratam da decisão de onde alocar recursos. Geralmente, os recursos a serem alocados são chamados de instalações e as entidades beneficiadas com a

alocação das instalações recebem o nome de clientes (DREZNER, 1995; DASKIN, 1995). O termo instalação pode ser usado para designar fábricas, depósitos, escolas etc., enquanto que a expressão clientes refere-se a depósitos, unidades de vendas, estudantes, etc. (LORENA et al., 2001). Segundo Daskin (1995), os problemas de localização de instalações tratam de decisões sobre onde devem ser localizadas instalações, considerando os clientes que podem ser servidos de forma a otimizar um certo critério estabelecido.

Aplicações de problemas de localização de instalações ocorrem nos setores privado e público. No setor privado, o objetivo principal é a minimização de custos logísticos. No setor público, por sua vez, há uma preocupação maior em maximizar a satisfação dos clientes, em detrimento dos custos necessários para o alcance de tal (MAPA; LIMA; MENDES, 2006).

O problema de localização pode ser modelado por meio de um grafo $G = (V, E)$, sendo que, $V(G)$ representa o conjunto dos vértices do grafo e $E(G)$ o conjunto de arestas de G . Dessa forma, a localização das instalações pode ocorrer mediante a abertura e atribuição de novos vértices, ou mediante a localização das instalações nos vértices existentes. Nos problemas de localização, pode ocorrer de, além da atribuição das instalações aos nós, o modelo ofereça o particionamento do conjunto de vértices em agrupamentos.

Conforme Owen e Daskin (1998) os problemas de localização podem ser divididos em três tipos:

- a) **Problema das p -Medianas:** consiste em localizar p instalações nos vértices de um grafo e alocar a demanda a estas instalações, de tal forma que minimize as distâncias percorridas entre os vértices. O problema das p -Medianas pode ser capacitado ou não.
- b) **Problema de Cobertura de Conjuntos:** baseia-se na distância ou tempo de viagem máximos aceitáveis estabelecidos inicialmente, desta forma, busca-se minimizar o número de instalações necessárias para garantir certo nível de cobertura de clientes. Para o problema de cobertura assume-se um conjunto finito de localizações que estão associadas a um orçamento fixo.
- c) **Problema dos Centros:** é um problema *minimax* cujo objetivo é minimizar a máxima distância entre os pontos de demanda e a instalação mais próxima. Deseja-se cobrir toda a demanda procurando localizar certo número de instalações, desde que minimize a distância coberta.

O Problema das p -Medianas ou *p-Median Problem* (pMP) é considerado capacitado quando as instalações que atendem aos clientes possuem certa capacidade que limita a

quantidade de clientes e demanda atendida. O p MP é considerado não capacitado, caso contrário.

No Problema das p -Medianas Capacitado ou *Capacitated p -Median Problem (CpMP)*, p corresponde ao número de instalações a serem alocadas e o termo mediana corresponde ao vértice candidato a alocar a instalação dentre todos os vértices existentes. Se em um vértice não for alocado uma instalação, ele será atendido por uma, ou seja, será um cliente.

As primeiras formulações do CpMP foram apresentadas por Hakimi (1965) e Mulvey e Beck (1984). O problema é bem conhecido como sendo *NP - difícil* (GAREY & JOHNSON, 1979). Problemas de localização podem ser conhecidos dadas as características do modelo como problemas de agrupamento ou problemas de clusterização (NEGREIROS; PALHANO, 2006). Um *cluster* ou agrupamento pode ser definido como um conjunto de entidades geograficamente concentradas em uma área, essas entidades possuem características em comum que foram levadas em conta para o agrupamento (PORTER, 1999).

Os problemas de localização e agrupamento podem ocorrer em dois ambientes de referência: o plano e a rede, visto que, a solução do problema requer a especificação das distâncias entre cada par de vértices presentes em um grafo. No caso do plano, o modelo vale-se das distâncias euclidianas entre os vértices. No caso da localização em rede, leva-se em conta os caminhos possíveis entre cada par de vértices, eventualmente afetados por elementos topográficos e barreiras de todo tipo.

No CpMP proposto por Negreiros, Batista e Rodrigues (2017) e Mulvey e Beck (1984) dado um grafo $G = (V, E)$, com um conjunto de vértices $G(E)$ em que todos os vértices possuem uma demanda de atendimento e todos os vértices são candidatos a alocarem as instalações presentes em um conjunto de cardinalidade p . Cada instalação tem uma capacidade de atendimento. O problema consiste em quais vértices alocar esse conjunto de instalações de modo a minimizar o custo de atendimento entre os vértices onde estão localizadas as instalações e os vértices que serão atendidos por essas instalações, nesse caso, os vértices clientes, a medida que a instalação também atende a demanda do vértice onde ela pode ser instalada. No CpMP a soma das demandas de todos os vértices que a instalação atende excluindo a demanda do vértice onde a instalação está instalada não deve ultrapassar a capacidade de atendimento da instalação. Modificações propostas no CpMP culminaram em um outro modelo conhecido como CpMP genérico ou simplesmente *generic Capacitated p -Median Problem (gCpMP)*. No modelo matemático gCpMP, a soma das demandas de todos os vértices que a instalação atende incluindo a demanda do vértice onde a instalação está instalada não deve ultrapassar a capacidade de atendimento da instalação.

Considerando as noções de clientes e instalações o $gCpMP$, segundo Negreiros, Batista e Rodrigues (2017), pode ser representado pelos conjuntos, parâmetros e variáveis a seguir: dado um grafo não direcionado $G = (V, E)$, V representa o conjunto de vértices G ; J representa o conjunto de vértices candidatos a serem locais onde possam ser localizadas instalações em G , as instalações absorvem as demandas dos vértices clientes, ou seja, vértices em que as instalações não estão localizadas, e dos vértices onde as instalações estão localizadas. Dessa forma, J representa um subconjunto dos vértices G . Os vértices candidatos a receberem a localização de instalações em J são chamados de medianas. K é um conjunto de grupos, $|K| = p$, p é o número de *clusters* ou medianas; d_{ij} é a distância entre um cliente i e a mediana j ; q_i é a demanda do cliente i ; Q_k representa a demanda máxima do grupo k ; x_{ij} é a variável binária que representa a alocação do cliente i a mediana j , ela pode assumir o valor 1, caso a associação exista e 0, caso a associação não exista; g_k é uma variável binária que representa a existência de um grupo k , a x_{ii} representam um vértice que será selecionado como a mediana do conjunto V .

Para o $gCpMP$, considere:

$$\text{Minimize } \sum_{i \in V} \sum_{j \in V} q_i d_{ij} x_{ij} \quad (2.24)$$

Sujeito a

$$\sum_{j \in V} x_{ij} = 1, \quad \forall i \in V \quad (2.25)$$

$$x_{ij} \leq x_{ii}, \quad (\forall i \in V); (\forall j \in V) \quad (2.26)$$

$$\sum_{i \in V} x_{ii} = \sum_{k \in K} g_k, \quad (2.27)$$

$$\sum_{i \in V} q_i x_{ij} \leq Q_k g_k, \quad \forall j \in J \quad (2.28)$$

$$\sum_{i \in V} x_{ii} = p, \quad (2.29)$$

$$g_k \in \{0,1\}, \quad \forall k \in K \quad (2.30)$$

$$x_{ij} \in \{0,1\}, \quad (\forall i \in V); (\forall j \in J) \quad (2.31)$$

A função objetivo (2.24) minimiza o custo de deslocamento entre clientes e medianas, que são os vértices candidatos a receberem as instalações. A restrição (2.25) indica que cada cliente pode ser atribuído a apenas uma mediana. A restrição (2.26) define que uma vez que uma mediana é usada, um cliente pode ser atribuído a instalação presente nessa mediana. A

restrição (2.27) indica que o número de medianas é igual ao número de grupos. A restrição (2.28) consideram que a soma da demanda dos clientes atribuídos a uma mediana não pode ultrapassar a capacidade da instalação presente nessa mediana. A restrição (2.29) limita o número de medianas utilizadas. As restrições (2.30) e (2.31) referem-se às variáveis de decisão binárias do problema.

O problema do $gCpMP$ é NP - *difícil*. O modelo pode ser resolvido rapidamente para instâncias em que as capacidades dos *clusters* são homogêneas. O modelo é $O(n^2)$, no número de variáveis binárias e restrições, onde n é a cardinalidade de I . Esta é uma formulação melhor do que a formulação $CpMP$ que é $O((n + m)^2)$ no número de variáveis e restrições, em que m é a cardinalidade de J (NEGREIROS; BATISTA; RODRIGUES, 2017).

Quanto a aplicação do modelo do Problema das p -Medianas Capacitado proposto por Mulvey e Beck (1984); Stefanello, Araújo e Müller (2014) empregam a modelagem em 6 cenários com a quantidade de vértices variando entre 100 e 402 e o número de medianas variando entre 10 e 40; Yaghini, Karimi e Rahbar (2013) aplicam o modelo em 4 cenários com a quantidade de vértices variando entre 5 e 200 e o número de medianas variando entre 5 e 20; Church (2008) estabeleceu 6 cenários com o número de vértices variando entre 100 e 500 e o número de medianas variando entre 5 e 50.

2.4 FÓRMULA DE HAVERSINE

Segundo Sinnott (1984) a Fórmula de Haversine é uma importante equação utilizada em navegação para encontrar a distância entre dois pontos na superfície de uma esfera a partir da informação da latitude e da longitude dos pontos.

Conforme Valsamis et al. (2017) apesar de o planeta Terra não possuir forma esférica, mas ser achatada nos polos, a Fórmula de Haversine caracteriza-se como uma simples e poderosa ferramenta para calcular a menor distância aproximada entre dois pontos na superfície da Terra.

A Fórmula de Haversine é dada pela Equação 2.32, extraída de Seo et al. (2017). Nela d é a distância entre dois pontos em quilômetros na superfície da Terra; R é o raio da Terra; φ_1 , φ_2 são a latitude do ponto 1 e a latitude do ponto 2 (em radianos) e λ_1 , λ_2 são a longitude do ponto 1 e a longitude do ponto 2 (em radianos).

$$d = 2.R.\sin^{-1}\left(\sqrt{\sin^2\left(\frac{(\varphi_2 - \varphi_1)}{2}\right) + \cos\varphi_1\cos\varphi_2\sin^2\left(\frac{\lambda_2 - \lambda_1}{2}\right)}\right) \quad (2.32)$$

3 REVISÃO SISTEMÁTICA DE LITERATURA

O presente capítulo traz 3 revisões de literatura distintas. A primeira Revisão Sistemática de Literatura identifica os trabalhos publicados na literatura referentes a técnicas utilizadas para o Problema de Roteamento de Leituristas. A segunda Revisão Sistemática de Literatura busca identificar abordagens *cluster first - route second* utilizadas na literatura para a resolução do Problema do Carteiro Chinês Capacitado. A terceira Revisão Sistemática de Literatura procura verificar que trabalhos na literatura utilizam o Problema das p -Medianas para a resolução do Problema de Roteamento de Arcos. As três Revisões de Literatura foram realizadas mediante a metodologia proposta por Gonçalves et al. (2012) e Sampaio e Mancini (2007) explorada abaixo.

Entende-se por revisão da literatura a análise e síntese de informações disponíveis em estudos relevantes publicados sobre um determinado tema, de forma a resumir o corpo de conhecimento existente e conduzir a conclusões sobre um determinado tema de interesse (SAMPAIO; MANCINI, 2007). A revisão da literatura pode ser classificada como narrativa, sistêmica e integrativa.

Alvo desse trabalho, a revisão sistemática revisa trabalhos publicados anteriormente, buscando responder determinadas questões levantadas. Essa revisão ocorre mediante a aplicação de métodos explícitos e sistematizados de busca, apreciação crítica e síntese da informação selecionada. Conforme Galvão e Pereira (2014) as revisões sistemáticas são estudos secundários, que tem nos estudos primários sua fonte de dados. Entende-se por estudos primários os artigos científicos que relatam os resultados da pesquisa originalmente.

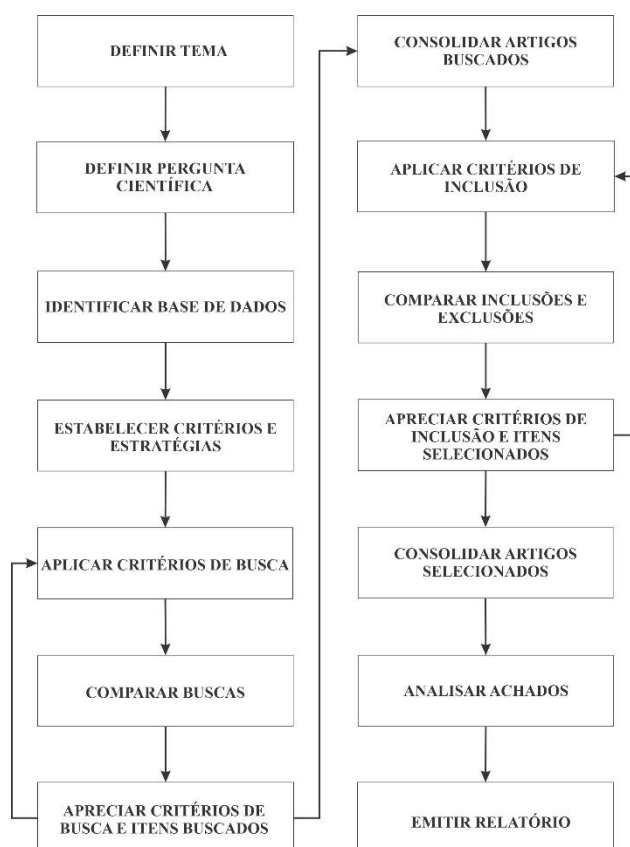
A Revisão Sistêmica da Literatura (RSL) também pode ser conhecida como Estudo da Arte. Segundo Romanowski e Ens (2006), o termo Estado da Arte é usado para definir estudos de diversos portes que pretendem captar trabalhos publicados em determinadas áreas do conhecimento e analisá-los. De acordo com Ferreira e Matos (2012), a execução de estudos como a RSL ou o levantamento do Estado da Arte advêm da necessidade de identificar avanços e lacunas nas pesquisas científicas realizadas ao redor do mundo e frequentemente publicadas virtualmente.

Conforme Drummond, Silva e Coutinho, (2004) a revisão de literatura caracteriza-se por ser replicável, reduzir o viés dos autores da pesquisa, ser facilmente atualizada, identificar lacunas da literatura que possam ser exploradas e fornecer uma base confiável para a tomada de decisão.

Existem na literatura as mais diversas metodologias para a execução da Revisão Sistemática da Literatura, em sua maioria essas metodologias preveem segundo Galvão e Pereira (2014): elaboração da pergunta de pesquisa; busca na literatura; seleção dos artigos; extração dos dados; avaliação da qualidade metodológica; síntese dos dados; avaliação da qualidade das evidências; redação e publicação dos resultados.

A partir das metodologias de Revisão Sistemática de Literatura propostas por Gonçalves et al. (2012) e Sampaio e Mancini (2007) construiu-se o fluxograma presente na Figura 18. A Figura 18 mostra as 14 etapas do processo de RSL utilizado nesse trabalho.

Figura 18 – Fluxograma do processo de revisão sistemática da literatura



Fonte: Adaptado Sampaio e Mancini (2007).

As 14 etapas do processo de RSL utilizado nesse trabalho serão descritas nas alíneas abaixo:

- a) **definir tema:** estabelecer o tema ou objeto de estudo a ser tratado.
- b) **definir pergunta científica:** inicialmente define-se a pergunta ou questão-chave a ser respondida ligada ao tema.

- c) **identificar base de dados:** escolha e estabelecimento da base de dados ou buscadores na web onde será realizada a pesquisa
- d) **estabelecer critérios e estratégias:** nessa etapa estabelecem-se os critérios e estratégias ligadas às atividades de busca e inclusão de trabalhos na revisão. Os critérios devem estar relacionados ao tema e expressar o desejo de responder à pergunta estabelecida ou questões-chave. A estratégia diz respeito ao planejamento e coordenação de todos os elementos presentes na revisão de literatura com o objetivo de responder a pergunta formulada. Os critérios de busca podem ser palavras-chaves, data de publicação, idioma, ou qualquer elemento que limite o espaço de busca. Os critérios de inclusão dos trabalhos podem estar ligados a aspectos como a abrangência da pesquisa encontrada em cada trabalho, ao tipo de pesquisa desenvolvido nas publicações, fator de impacto da publicação, ou qualquer aspecto que não possa ser filtrado diretamente por meio de uma busca em uma base de dados
- e) **aplicar critérios de busca:** com o estabelecimento da base de dados e da estratégia e critérios de busca, dois pesquisadores independentemente irão realizar o mesmo processo de procura, sob as mesmas condições
- f) **comparar buscas:** após a condução das buscas independentes, os trabalhos serão comparados e confrontados para o estabelecimento de uma lista única de publicações.
- g) **apreciar critérios de busca:** nessa etapa cabe discutir o impacto dos critérios estabelecidos para a pesquisa e se modificações nos critérios de busca afetariam o resultado encontrado. Caso modificações nos critérios sejam cabíveis retorna-se a etapa da alínea e.
- h) **consolidar artigos buscados:** novas discussões sobre a legitimidade da presença dos artigos buscados na lista serão levantadas, tendo em vista os critérios estabelecidos. Posteriormente, a lista será consolidada.
- i) **aplicar critérios de inclusão:** os critérios de inclusão serão aplicados a lista por dois pesquisadores independentemente. Inicialmente verifica-se se as idéias presentes no título e resumos em português e inglês estão de acordo com os critérios de inclusão. Caso o resultado for inconclusivo, cabe a leitura do trabalho por completo.
- j) **comparar inclusões e exclusões:** os resultados da aplicação dos critérios de inclusão de ambos os pesquisadores serão confrontados. E uma nova lista com os trabalhos inclusos para a revisão será estabelecida.

- k) **apreciar critérios inclusão:** os critérios de inclusão e seus impactos na formação da lista de trabalhos serão discutidos. Caso seja pertinente, os critérios podem ser modificados, logo retorna-se a alínea i.
- l) **consolidar artigos incluídos:** discussões concernentes a justificativa da inclusão dos trabalhos na revisão será estabelecida, e uma lista única com os trabalhos incluídos consolidada.
- m) **analisar achados:** todos os trabalhos presentes na lista única serão analisados, iniciando pelo título, resumos, palavras-chaves e conteúdo
- n) **emitir relatório:** um relatório com as conclusões e a revisão do estado da arte será emitido.

3.1 ABORDAGENS PARA RESOLUÇÃO DO PROBLEMA DE ROTEAMENTO DE LEITURISTAS

Mediante o uso da metodologia proposta na Figura 18, o tema a ser tratado foi o roteamento de leituristas ou *routing meter readers*. A pergunta a ser respondida estabelecida é: Quais artigos científicos publicados em revistas, livros ou congressos tratam do problema de roteamento de leituristas e que técnicas foram aplicadas ou propostas para a resolução desse problema em língua portuguesa e inglesa?

A pesquisa utilizou como banco de dados o Google Scholar e os Periódicos da Capes que possuem um acervo com as revistas mais importantes e relevantes do mundo. Como critérios de busca foram utilizadas as palavras-chave: *routing meter reader*, *routing meter readers*, roteamento de leiturista e roteamento de leituristas, bem como a definição da língua dos trabalhos pesquisados: português e inglês.

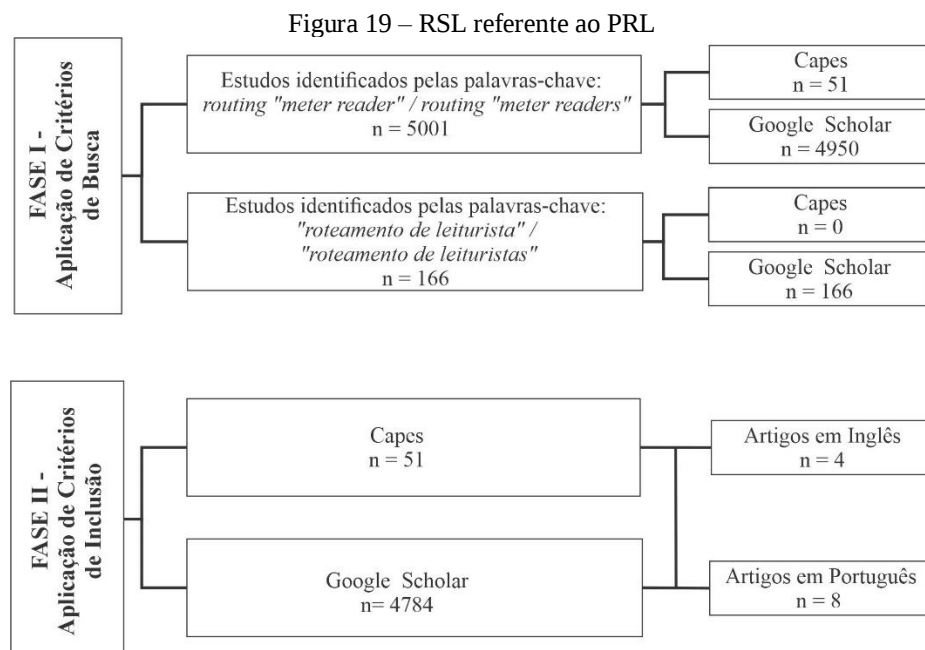
Não estabeleceu-se datas mínimas ou máximas para as publicações buscadas, então as palavras-chaves utilizadas podem vasculhar os trabalhos de todos os anos disponíveis na base de dados. A pesquisa ocorreu entre janeiro e fevereiro de 2020, logo os trabalhos buscados foram publicados até estes meses. Como critério de inclusão definiu-se que a pesquisa deveria tratar ou propor algoritmos e técnicas para a resolução do PRL, o trabalho para ser incluso deveria ser um artigo científico publicado em livro, revista ou congresso. A RSL utilizada para identificar técnicas propostas ou utilizadas para o PRL estruturou-se em 2 Fases conforme a Figura 19, estas 2 Fases estão de acordo com a metodologia da Figura 18.

Na Fase I utilizou-se os termos *routing “meter reader”* e *routing “meter readers”*, bem como as palavras - chaves “roteamento de leiturista” e “roteamento de leituristas” para a

pesquisa dos trabalhos presentes nas bases de dados Capes e Google Scholar que fazem referência aos termos. A presença de aspas significa que foi utilizado o termo exato para procura, esse procedimento é usado quando a palavra-chave buscada é composta.

Na Fase I os termos *routing “meter reader”* e *routing “meter readers”* revelaram 5001 trabalhos, desses trabalhos, 51 estavam presentes nos Periódicos da Capes e 4950 no Google Scholar; os termos “roteamento de leitorista” e “roteamento de leitoristas” expuseram 166 trabalhos no Google Scholar.

Na Fase II os trabalhos dos Periódicos da Capes e do Google Scholar pesquisados pelos termos *routing “meter reader”*, *routing “meter readers”*, “roteamento de leitorista” e “roteamento de leitoristas” na Fase I foram consolidados em duas listas uma apenas com trabalhos publicados na Capes, nesse caso, 51 pesquisas, e outra com os trabalhos do Google Scholar, 4784 pesquisas. Dentre os trabalhos listados nos dois bancos de dados, os trabalhos que tratam efetivamente da aplicação ou proposição de técnicas para o roteamento de leitoristas em língua portuguesa e inglesa foram consolidados. Dessa forma, 4 trabalhos em língua inglesa foram encontrados e 25 em língua portuguesa. Os trabalhos da Fase II em língua inglesa e portuguesa foram revisados e trabalhos duplicados retirados, restaram 4 trabalhos em língua inglesa e 8 em língua portuguesa.



Fonte: O autor (2020).

Na Fase II os trabalhos publicados em inglês identificados foram: Stern e Dror (1979), Wunderlich et al. (1992), Usberti, França e França (2011b) e Cunha et al. (2017). Já os trabalhos

em português identificados foram: Smiderle, Steiner e Carnieri (2003), Usberti, França e França (2008), Usberti, França e França (2012a), Usberti, França e França (2011a), Usberti, França e França (2011c), Arakaki e Usberti (2018) e Arakaki e Usberti (2019).

Stern e Dror (1979) propõem um procedimento heurístico *Route First – Cluster Second* para o roteamento de leituristas em um bairro da cidade de Beersheva, Israel. A heurística proposta é um procedimento de dois estágios. Na primeira fase, o limite máximo do turno de trabalho é relaxado, então realiza-se um circuito euleriano que cobre todas as ruas ao menos uma vez. No segundo estágio, o único circuito é dividido em segmentos de trabalho, cada um com comprimento não superior ao limite máximo do turno de trabalho estabelecido.

Wunderlich et al. (1992) utiliza o algoritmo de partição de arcos proposto por Bodin e Levy (1991) para a programação de leituristas na companhia Southern California Gas Company (SOCAL). O algoritmo utilizado possui uma abordagem *Cluster First – Route Second*, nesse caso as ruas da localidade estudada são tratadas como arcos, as ruas podem ser dotadas de tempo de leitura e percurso. Quando apenas um dos lados da rua exige leitura, cria-se um arco virtual com tempo de vistoria zero e tempo de percurso igual à de sua aresta paralela. Dada uma rede conectada com um conjunto R de arcos, onde cada arco (i, j) tem um tempo de serviço t associado, arcos virtuais são adicionados a R de modo que a rede resultante seja um conjunto R' que tem exatamente dois arcos entre cada par de nós. O problema de particionamento de arco é dividir R' em partições de modo que a carga de trabalho em cada partição seja aproximadamente a mesma, cada arco em R' é atribuído a uma partição. A existencia de arcos paralelos e com sentido contrário entre dois nós adjacentes garante sempre um número par de arestas incidentes para qualquer nó, que consiste na condição necessária e suficiente para o grafo não direcionado ser Euleriano.

O trabalho de Smiderle, Steiner e Carnieri (2003) utiliza um procedimento *Cluster First – Route Second* dividido em duas fases distintas: na primeira fase são estabelecidos agrupamentos que não ultrapassam a capacidade de atendimento de cada leiturista, na segunda etapa realiza-se o roteamento por meio do Problema do Carteiro Chinês. A primeira fase utiliza um algoritmo genético para encontrar 12 medianas, essas medianas formarão grupos que respeitarão a capacidade de atendimento de cada leiturista. A segunda fase consisten na aplicação do Problema do Carteiro Chinês a cada *cluster*.

Usberti, França e França (2008) demostram em seu trabalho que o Problema de Roteamento de Leituristas ou Meter Reader Problem (MRP) é *NP - difícil* e pode ser modelado como o Problema de Roteamento de Arcos Capacitados de Golden e Wong (1981).

Usberti, França e França (2011a) e Usberti, França e França (2011c) realizam modificações na programação linear do Capacitated Arc Routing Problem (CARP) e introduzem a formulação matemática do Problema de Roteamento de Arcos Capacitados Abertos (PRACA). O PRACA difere do PRAC, uma vez que, o PRACA não considera um depósito e as rotas não estão restritas a formar circuitos fechados. O autor sugere a aplicação do PRACA ao Problema de Roteamento de Leituristas. No trabalho também é proposta uma heurística de varredura para a resolução do PRACA.

Usberti, França e França (2011b) e Usberti, França e França (2012a) propõe a aplicação de uma metaheurística Greedy Randomized Adaptive Search Procedure (GRASP) com Path – Relinking (PR) baseado na abordagem PRACA para o roteamento de leituristas em um grafo conexo e não direcionado.

Em Arakaki e Usberti (2018) um Algoritmo Genético Híbrido ou Hybrid Genetic Algorithm (HGA) com Busca Local foi proposto para resolução do PRACA. Experimentos computacionais para um conjunto de instâncias de *benchmark* revelaram que o HGA proposto alcançou os melhores limites superiores para quase todas as instâncias. Os resultados foram comparados com outros métodos heurísticos da literatura.

Cunha et al. (2017) utilizou a metaheurística Variable Neighborhood Descent (VND) para a resolução do Problema de Roteamento de Leituristas e buscou melhorar a solução através de técnicas de *insert*, *2-opt* e *interchange*. A heurística foi incluída a probabilidade de falha do equipamento que faz a medição, foram analisados 50 cenários possíveis.

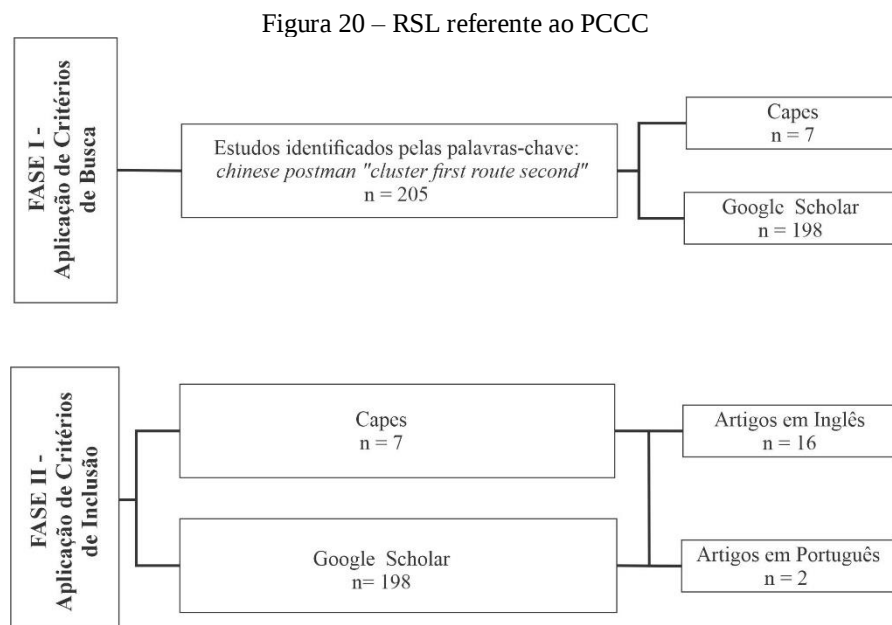
Arakaki e Usberti (2019) apresentam uma Formulação Relaxada Baseada em Fluxo ou Relaxed Flow-Based Formulation para roteamento de leituristas. Esta formulação não garante encontrar uma solução viável para o problema, mas fornece limites inferiores válidos para o CARP não direcionado, que pode ser usado para medir a qualidade de soluções viáveis atualmente conhecidas.

Os trabalhos de Stern e Dror (1979), Wunderlich et al. (1992), Smiderle, Steiner e Carnieri (2003) e Cunha et al. (2017) são os únicos que oferecem uma aplicação efetiva do Problema de Roteamento de Leituristas, os demais oferecem técnicas que podem ser utilizadas no roteamento.

3.2 ABORDAGEM CLUSTER FIRST - ROUTE SECOND E PROBLEMA DO CARTEIRO CHINÊS

Ainda utilizando-se a metodologia proposta na Figura 18, o tema a ser tratado foi quais técnicas *cluster first - route second* foram utilizadas para a resolução do Problema do Carteiro Chinês Capacitado na literatura. A pergunta a ser respondida estabelecida é: Quais artigos científicos publicados em revistas, livros ou congressos tratam da aplicação de técnicas *cluster first - route second* para a Resolução do Problema do Carteiro Chinês Capacitado e que técnicas foram aplicadas ou propostas para a resolução desse problema em língua inglesa e portuguesa?

A pesquisa utilizou como banco de dados o Google Scholar e os periódicos da Capes. Como critérios de busca estabeleceu-se as palavras-chave combinadas: “*chinese postman*” e “*cluster first - route second*”. Não estabeleceu-se datas mínimas ou máximas para as publicações buscadas. A pesquisa ocorreu entre janeiro e fevereiro de 2020, dessa forma, captou-se os trabalhos até os meses citados. A RSL aplicada para identificar técnicas propostas ou utilizadas para resolver o Problema do Carteiro Chinês Capacitado por meio de uma abordagem *cluster first - route second* estruturou-se em 2 Fases, ver Figura 20, em consoante com a metodologia da Figura 18.



Fonte: O autor (2020).

Na Fase I utilizou-se os termos *chinese postman* “*cluster first - route second*” para a pesquisa dos trabalhos presentes nas bases de dados Capes e Google Scholar que fazem

referencia aos termos. Ao todo foram encontrados 205 trabalhos, desses 7 estavam presentes nos Periodicos da Capes e 198 no Google Scholar.

Na Fase II dentre os trabalhos pesquisados, os trabalhos que tratam efetivamente da aplicação ou proposição de técnicas *cluster first - route second* para o Problema do Carteiro Chinês Capacitado em língua inglesa e portuguesa foram consolidados em duas listas, uma com 16 artigos em inglês e outra com 2 trabalhos em português.

Na Fase II os artigos publicados em língua inglesa identificados foram: Moss (1970), Marks e Stricker (1971), Liebling (1973), Transport Canada (1975), Bodin e Kursh (1978), Bodin e Kursh (1979), Benavent (1990), Atkins, Dierckman e O'bryant (1990), Gelders e Cattrysse (1991), Wunderlich et al. (1992), Ahr e Reinelt (2002), Ghiani et al. (2005), Perrier, Langevin e Amaya (2008), Monroy, Amaya e Langevin (2013), Cruz, Chirva e Santana (2015), Holmberg (2018). Os artigos em língua portuguesa foram: Smiderle, Steiner e Carnieri (2003), Usberti, França e França (2012a).

Dos artigos identificados na Fase II, alguns dedicam-se ao estudo do roteamento de veículos de neve, ao roteamento de veículos de transporte de lixo, ao roteamento de veículos em geral, ou mesmo ao roteamento de leiturista, conforme pode-se ver a seguir:

- a) roteamento de veículos de remoção de neve: Moss (1970), Marks e Stricker (1971), Liebling (1973), Transport Canada (1975), Bodin e Kursh (1978), Bodin e Kursh (1979), Atkins, Dierckman e O'bryant (1990), Holmberg (2018).
- b) roteamento de veículos em geral: Benavent et al. (1990), Ahr e Reinelt (2002), Perrier, Langevin e Amaya (2008), Monroy, Amaya e Langevin (2013).
- c) roteamento de veículos de transporte de lixo urbano: Gelders e Cattrysse (1991), Ghiani et al. (2005), Cruz, Chirva e Santana (2015).
- d) roteamento de leiturista: Wunderlich et al. (1992), Smiderle, Steiner e Carnieri (2003), Usberti, França e França (2012a).

O trabalho de Moss (1970) discute como a teoria dos grafos pode ser utilizada para o agrupamento de arestas em *clusters* e como cada rota atribuída a cada *cluster* pode ser tratada como um circuito Euleriano.

No trabalho de Marks e Stricker (1971) o problema de roteamento é modelado como um Problema do Carteiro Chinês não Dirigido de m -veículos, onde a rede de transporte é particionada em m sub-redes, posteriormente, um Problema do Carteiro Chinês é resolvido para cada agrupamento usando uma heurística de decomposição.

Liebling (1973) utiliza a teoria dos grafos para dividir uma cidade em setores, onde cada setor é designado a um veículo. Em seguida, um Problema do Carteiro Chinês é resolvido em cada setor.

No trabalho do Ministério de Transportes Canadense ou Transport Canada (1975) a fase de cluster divide o grafo original em pequenos subgrafos, de acordo com regras estabelecidas. Em particular, cada subgrafo deve conter um número par de nós de graus ímpares, tornando o local o mais compacto e centralizado possível. A fase de roteamento resolve o Problema do Carteiro Chinês não Direcionado em cada subgrafo e o algoritmo de Fleury é usado para determinar um circuito Euleriano no subgrafo resultante.

Bodin e Kursh (1978) oferecem uma modelagem genérica para a resolução de problemas de roteamento por meio do conceito *cluster first - route second*.

Bodin e Kursh (1979) destrincham e esclarecem a modelagem proposta por Bodin e Kursh (1978).

Benavent et al. (1990) propõem a clusterização inicial por meio do Generalized Assignment Problem para a formação de k centros, respeitando-se a capacidade de cada veículo. Para a roteirização nos centros utiliza-se o Problema do Carteiro Chinês Rural ou Rural Postman Problem.

Em Atkins, Dierckman e O'bryant (1990) um grafo G é inicialmente dividido em dois subgrafos de aproximadamente o mesmo tamanho. Um circuito Euleriano é então construído para cada subgrafo traçando um caminho fechado para cada *cluster* por meio de Spanning Tree.

Gelders e Cattrysse (1991) propõem a clusterização por meio do Path Scanning Algorithm respeitando a capacidade dos veículos e uso do Problema do Carteiro Chinês para a roteamento em cada *cluster*.

Ahr e Reinelt (2002) propõem a formação de k -cluster por meio do Farthest-Point Clustering Algorithm e roteamento por meio do Shortest Path Tour Problem.

No trabalho de Ghiani et al. (2005) a clusterização ocorre por meio de uma heurística construtivista para o problema das p -medianas, o roteamento dos veículos acontece por meio da aplicação do Problema do Carteiro Chinês Rural.

O algoritmo *cluster first – route second* proposto por Perrier, Langevin e Amaya (2008) determina primeiro uma partição dos arcos a serem atendidos em *clusters* compactos, cada *cluster* tendo aproximadamente a mesma carga de trabalho. Como os veículos saem em velocidades diferentes, a carga total de trabalho é medida em unidades de tempo (por exemplo, minutos). Uma rota de veículo é então construída em cada *cluster* por meio do Carteiro Chinês

Rural Hierárquico. O Carteiro Chinês Rural Hierárquico foi introduzido por Ghiani e Improta (2000) e nele a visita de cada arco segue uma relação de precedência definida anteriormente.

No trabalho de Monroy, Amaya e Langevin (2013) na primeira fase, arcos próximos são atribuídos a cada rota sem violar a capacidade da rota; na segunda fase, um problema de roteamento de arco de veículo único é resolvido para cada rota. O primeiro passo das heurísticas é inspirado no trabalho desenvolvido por Perrier et al. (2008). Em seguida, um modelo de programação inteiro misto é usado na segunda fase para resolver o roteamento para cada rota.

Cruz, Chirva e Santana (2015) realizam testes de otimização e utilizam três algoritmos para agrupar: Centroid-based Heuristic Algorithm, Sweep algorithm e Generalized Assignment Problem (GAP) e para os três casos fazem a roteirização por meio de um algoritmo Hybrid Mixed-Integer.

Holmberg (2018) apresenta uma abordagem *cluster first – route second* para resolução de roteamento de veículos, inicialmente utiliza-se uma metaheurísticas que combina conceitos de *k-means clustering* e algoritmos de busca local para o agrupamento de arestas, posteriormente usa-se o Problema do Carteiro Chinês Rural Hierárquico para o roteamento.

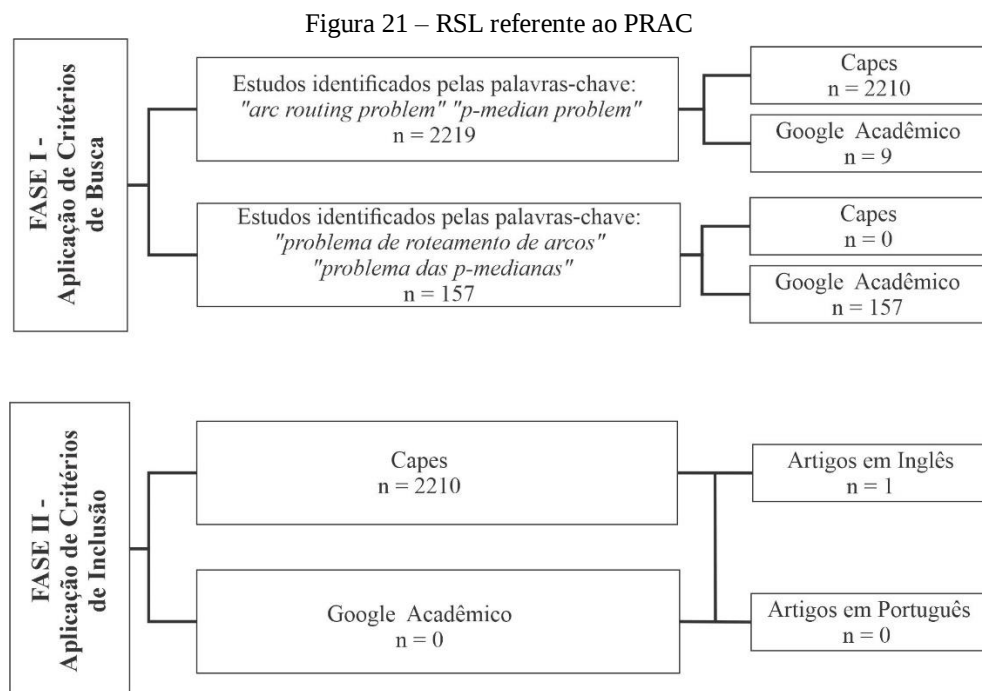
3.3 PROBLEMA DE ROTEAMENTO DE ARCOS E PROBLEMA DAS p -MEDIANAS

A partir da metodologia proposta na Figura 18, o tema a ser tratado agora é que trabalhos utilizam o Problema das p -Medianas na fase de agrupamento para a resolução do Problema de Roteamento de Arcos Capacitados. A pergunta a ser respondida estabelecida é: Quais artigos científicos publicados em revistas, livros ou congressos utilizam o Problema das p -Medianas na fase de agrupamento para a resolução do Problema de Roteamento de Arcos Capacitados e que técnicas foram aplicadas ou propostas para a resolução desse problema em língua inglesa e portuguesa?

A pesquisa utilizou como banco de dados o Google Scholar e os periódicos da Capes. Como critérios de busca estabeleceu-se as palavras-chave combinadas: “*arc routing problem*” “*p-median problem*” em inglês, já em português as palavras selecionadas foram “problema de roteamento de arcos” “problema das p -medianas”. A RSL aplicada para identificar que trabalhos utilizam o Problema das p -Medianas na fase de agrupamento para a resolução do Problema de Roteamento de Arcos Capacitados estruturou-se em 3 Fases, ver Figura 21, em acordo com a metodologia da Figura 18.

Na Fase I utilizou-se os termos “*arc routing problem*” “*p-median problem*” e “problema de roteamento de arcos” “problema das *p*-medianas” para a pesquisa dos trabalhos presentes nas bases de dados Capes e Google Scholar que fazem referência aos termos.

Na Fase II dentre os trabalhos pesquisados, os trabalhos que tratam efetivamente da aplicação do Problema das *p*-Medianas para a resolução do Problema de Roteamento de Arcos Capacitados em língua inglesa e portuguesa foram consolidados. Na Fase III, os trabalhos da Fase II foram revisados e artigos duplicados retirados.



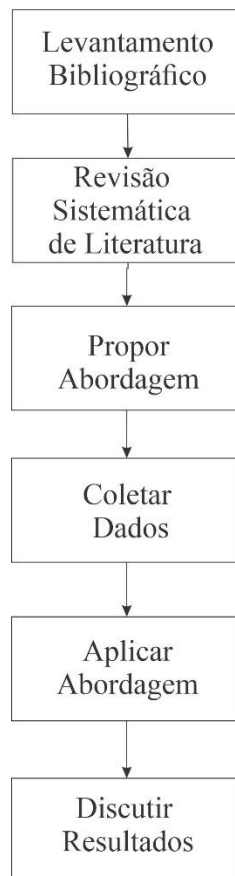
Fonte: O autor (2020).

Na Fase III apenas um artigo em inglês foi identificado a saber: Vecchi et al. (2016). Vecchi et al. (2016) utiliza o Problema das *p*-Medianas para o agrupamento de arcos. A segunda desenvolve-se o Problema de Roteamento de Arcos Capacitados (CARP). A terceira fase realiza a aplicação de um algoritmo adaptado de Hierholzer para o sequenciamento dos arcos obtidos na fase anterior.

4 METODOLOGIA

A metodologia desse trabalho estrutura-se em seis etapas. Essas etapas podem ser observadas na Figura 22. Inicialmente realizou-se o levantamento bibliográfico dos assuntos referentes a Teoria dos Grafos, ao Problema do Carteiro Chinês, ao Problema do Carteiro Chinês Capacitado e demais temas cabíveis. A Revisão Sistemática de Literatura possibilitou o estudo de lacunas presentes na literatura a serem estudadas. Após a Revisão da Literatura, propôs-se uma abordagem para o roteamento de leituristas baseada em um procedimento *cluster first – route second*, coletou-se os dados reais e aplicou-se a abordagem proposta.

Figura 22 - Etapas de execução da metodologia desta pesquisa



Fonte: O autor (2020).

Abaixo serão discutidos aspectos ligados a classificação da pesquisa, bem como a abordagem *cluster first – route second* proposta para o solucionamento do Problema de Roteamento de Leituristas.

4.1 CLASSIFICAÇÃO DA PESQUISA

As motivações para a condução da pesquisa partem da identificação de lacunas presentes na teoria e da existência de demandas práticas.

O presente estudo possui características de natureza aplicada e buscará encontrar soluções para o Problema de Roteamento de Leituristas ao propor rotas a serem percorridas. Segundo Saunders, Lewis e Tornhill (2007) uma pesquisa é classificada como aplicada, quando o trabalho busca encontrar resultados a serem utilizados na solução de impasses.

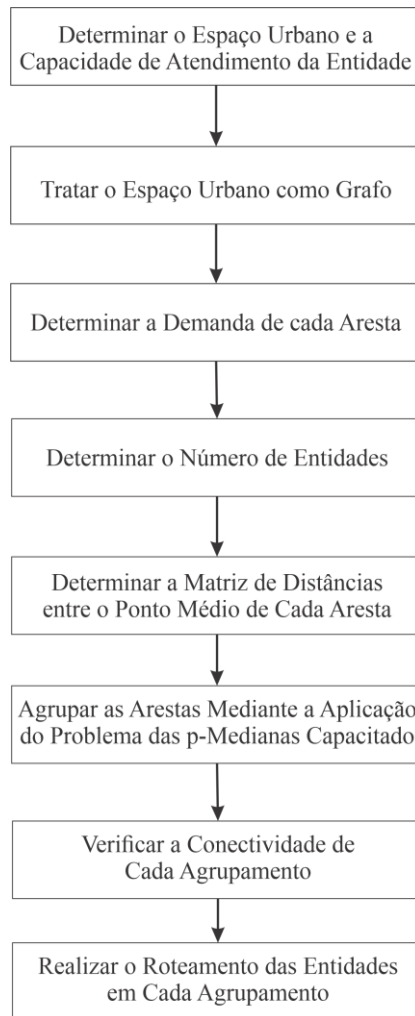
Quanto a abordagem, a pesquisa é caracterizada como quantitativa, conforme Gil (2008), uma pesquisa caracteriza-se como quantitativa quando é centrada na objetividade do assunto tratado e utiliza dados brutos para a compreensão da realidade do estudo, recorrendo à linguagem matemática na descrição de seus resultados.

Quanto ao delineamento da pesquisa, este trabalho pode ser classificado como modelagem e simulação, uma vez que utiliza técnicas matemáticas e computacionais para aprimorar o funcionamento de sistemas reais ou desenvolver representações de situações, com o objetivo de aperfeiçoar o processo de tomada de decisões (MIGUEL; FLEURY; MELLO, 2012).

4.2 ABORDAGEM PARA ROTEAMENTO PROPOSTA

A abordagem de roteamento proposta neste trabalho possui 8 passos sintetizados em duas fases, na primeira fase ocorre o agrupamento em *clusters* dos segmentos de rua da rede urbana estudada por meio do *gCpMP*, os *clusters* resultantes correspondem a subgrafos do grafo original. Na segunda fase, a conectividade de cada subgrafo é testada. Caso o subgrafo não seja conexo, as arestas ou segmentos de rua do grafo original, que possuem o menor custo de atravessamento e tornam o subgrafo conexo são adicionadas ao agrupamento. Depois de tornar o subgrafo ou cluster conexo, realiza-se o roteamento por meio do Problema do Carteiro Chinês. Os oito passos da abordagem são apresentados no fluxograma da Figura 23.

Figura 23 - Etapas de execução da abordagem cluster first – route second



Fonte: O autor (2020).

4.2.1 Determinar o espaço urbano e carga de trabalho

Delimitar o espaço urbano onde será realizado o roteamento e a carga de trabalho de cada leitorista.

4.2.2 Tratar o espaço urbano como grafo

Tratar as ruas do espaço urbano delimitado como um grafo G não direcionado composto por um conjunto de arestas e vértices. Os vértices do grafo G serão os pontos na malha urbana que possibilitem o leitorista atravessar de uma rua para outra rua, ou de um lado de determinada

rua para o outro lado desta mesma rua. Logo, é trivial considerar, que as arestas correspondem aos segmentos de rua entre os pontos estabelecidos como vértices.

No espaço urbano tratado caso todos os arcos ou arestas do grafo do espaço urbano possuam demanda maior que zero, o problema tratado será um PCCC.

4.2.3 Determinar demanda de cada aresta

Cada aresta (i, j) do grafo G do espaço urbano estudado possui uma demanda de atravessamento w_{ij} e um custo de atravessamento c_{ij} . Caso o segmento de rua precise ser percorrido em ambos os lados, w_{ij} corresponderá à soma da demanda de atravessamento dos vértices nas direções (i, j) e (j, i) . Cabe salientar que o Problema das p -Medianas aplicado para o agrupamento irá considerar o peso de cada aresta para a clusterização, em alguns casos o peso de cada aresta, corresponde a percorrer o segmento de rua em ambos os lados.

O custo de atravessamento c_{ij} pode ser dado em metros e representa o custo de deslocamento entre os vértices (i, j) . Para o roteamento e aplicação de um Problema do Carteiro Chinês posteriormente, segmentos de ruas que precisem ser percorridas em ambos os lados podem ser tratados com arestas direcionadas, com um arco saindo e outro entrando em cada vértice do segmento. Caso uma rua precise ser percorrida em apenas um lado a aresta pode ser considerada não direcionada.

Tanto o PRAC, quanto o PCCC consideram que ao atravessar uma aresta ou arco pela primeira vez a demanda do arco ou aresta são captadas, no entanto, uma aresta ou arco podem ser percorridos mais de uma vez para a construção de um circuito euleriano, contudo, nessas vezes adicionais a demanda de atendimento não é computada. Fato semelhante ocorre com a abordagem proposta ao agrupar as arestas considerando a demanda de percorrer cada aresta do agrupamento apenas uma vez e aplicar o Problema do Carteiro Chinês ao cluster resultante.

No Problema de Roteamento de Leituristas a demanda de atravessamento w_{ij} de cada aresta é dada pela Equação (4.1):

$$w_{ij} = l_{ij} + v_{ij} \quad (4.1)$$

em que w_{ij} representa a soma em minutos do tempo de passagem que o leiturista utiliza para percorrer a aresta entre os vértices (i, j) expresso por l_{ij} e o tempo para realizar a vistoria entre os vértices (i, j) dado por v_{ij} . Quando w_{ij} representa a demanda de atravessamento de um

segmento de rua em ambos os lados, w_{ij} é a soma do tempo de passagem l_{ij} e do tempo de vistoria v_{ij} de ambos os lados.

O tempo de vistoria de um segmento de rua qualquer é a quantidade de medidores presentes em um lado ou em ambos os lados de um segmento multiplicado pelo tempo médio que um leitorista leva para realizar a leitura de um medidor. O tempo de vistoria pode ser calculado conforme a Equação (4.2), em que v_{ij} representa o tempo de vistoria de determinado segmento (i, j) ; $QtPontos_{ij}$ representa a quantidade de pontos de leitura de um segmento (i, j) e $TpLeitura$ o tempo de leitura de cada ponto.

$$v_{ij} = QtPontos_{ij} * TpLeitura \quad (4.2)$$

4.2.4 Definir o número de leitoristas

O número de leitoristas, expresso por p , deve ser maior que a soma do tempo de atravessamento, w_{ij} , de todas as arestas do grafo G dividido pela carga de trabalho máxima de um leitorista, dada por Q . Como a divisão pode oferecer como resultado um número decimal, arredonda-se esse número para o próximo inteiro com valor superior.

4.2.5 Determinar matriz de distâncias

Para cada segmento de rua ou aresta será identificado o ponto médio. O ponto médio de um segmento de reta é o ponto que separa o segmento em duas partes com medidas iguais. Serão, então, identificadas as coordenadas geográficas, latitude e longitude, para cada ponto médio de cada aresta do grafo G ; a partir das coordenadas geográficas dos pontos médios, uma matriz de distâncias entre os pontos médios das arestas será estabelecida, por meio da Fórmula de Haversine.

4.2.6 Aplicar o problema da p -medianas

O $gCpMP$ será modelado utilizando-se as informações referentes ao tempo de atravessamento, w_{ij} , de cada aresta; a matriz de distâncias entre os pontos médios; a capacidade de atendimento ou demanda máxima que um leitorista poderia atender e o número de leitoristas.

No $gCpMP$, o número de leitoristas é igual ao número de agrupamentos ou *clusters*, logo cada agrupamento possui uma demanda máxima de atendimento, essa demanda máxima

de atendimento limitará o quanto de arestas dado o tempo de atravessamento, w_{ij} , estão associadas a um *cluster* na fase de agrupamento.

A decisão de designar leituristas e atribuir segmentos de ruas a eles é semelhante a decisão de onde localizar instalações e atribuir a demanda dos clientes a essas localidades.

O $gCpMP$ pode ser aplicado conforme os seguintes parâmetros e variáveis: dado um grafo não direcionado $G = (V, E)$ que representa uma rede urbana. Em G , V representa o conjunto de pontos médios dos segmentos de rua; E representa o conjunto de arestas da matriz de distâncias entre os pontos médios; J representa o conjunto de pontos médios candidatos a serem locais onde possam ser localizados leituristas em G , os leituristas absorvem as demandas dos pontos médios considerados clientes e dos pontos médios onde estão instalados. Os vértices candidatos a receberem leituristas são chamados de medianas. K é a quantidade de leituristas necessárias para realizar o roteamento; $|K| = p$, p é o número de *clusters*, medianas, rotas ou leituristas a serem alocados; d_{ij} é a distância entre um ponto médio i e a mediana candidata a receber o leiturista j ; q_i é a demanda de tempo necessária para atender ao segmento de rua associado ao ponto médio i ; Q_k representa a capacidade máxima de atendimento de um leiturista k ; x_{ij} é a variável binária que representa a alocação do ponto médio i ao leiturista presente na mediana j , ela pode assumir o valor 1, caso a associação exista e 0, caso a associação não exista; g_k é uma variável binária que representa a existência de um conjunto de pontos médios associados ao leiturista k , a x_{ii} representam um ponto médio que será selecionado como a mediana do conjunto V .

Para o $gCpMP$, considere:

$$\text{Minimize } \sum_{i \in V} \sum_{j \in V} q_i d_{ij} x_{ij} \quad (4.3)$$

Sujeito a

$$\sum_{j \in V} x_{ij} = 1, \quad \forall i \in V \quad (4.4)$$

$$x_{ij} \leq x_{ii}, \quad (\forall i \in V); (\forall j \in V) \quad (4.5)$$

$$\sum_{i \in V} x_{ii} = \sum_{k \in K} g_k, \quad (4.6)$$

$$\sum_{i \in V} q_i x_{ij} \leq Q_k g_k, \quad \forall j \in J \quad (4.7)$$

$$\sum_{i \in V} x_{ii} = p, \quad (4.8)$$

$$g_k \in \{0,1\}, \forall k \in K \quad (4.9)$$

$$x_{ij} \in \{0,1\}, (\forall i \in I); (\forall j \in J) \quad (4.10)$$

A função objetivo (4.3) minimiza o custo de deslocamento entre pontos médios e medianas, que são os vértices candidatos a receberem os leituristas. A restrição (4.4) indica que cada ponto médio pode ser atribuído a apenas um leiturista. A restrição (4.5) define que uma vez que um leiturista é alocado, um ponto médio pode ser atribuído ao leiturista presente nessa mediana. A restrição (4.6) indica que o número de medianas é igual ao número de leituristas. A restrição (4.7) considera que a soma da demanda dos pontos médios atribuídos a um leiturista não pode ultrapassar a capacidade do leiturista presente nessa mediana. A restrição (2.28) limita o número de medianas utilizadas ao número de leituristas. As restrições (2.9) e (2.10) referem-se às variáveis de decisão binárias do problema.

Como resultado da aplicação do *gCpMP* para a área estudada tem-se como resultado um determinado número de *clusters*, cada *cluster* é um subgrafo não direcionado do grafo original, cada agrupamento absorve certa demanda de tempo de atravessamento do grafo original, cada agrupamento também possui uma distância total a ser percorrida pelo leiturista, que representa o custo de atravessamento de todas as arestas do subgrafo.

4.2.7 Verificar conectividade

Cada *cluster* possui um determinado número de arestas e a cada aresta estão associados vértices. A conectividade do subgrafo de cada agrupamento será verificada por meio de um Algoritmo de Busca em Profundidade Modificado presente na Figura 24.

Para a aplicação de um Algoritmo de Busca em Profundidade Modificado, tem-se como entrada a lista de adjacências dos vértices de um *cluster*. Caso o *cluster* seja conexo obtém-se como resultado uma lista com apenas um conjunto de vértices. Caso o *cluster* seja desconexo tem-se como resultado uma lista com mais de um conjunto disjunto de vértices. No presente trabalho, todos os *clusters* desconexos obtidos após a aplicação do Problema das *p*-Medianas apresentavam apenas dois conjuntos disjuntos de vértices.

O Algoritmo de Busca em Profundidade Modificado utilizado nesta pesquisa deriva do Algoritmo de Busca em Profundidade. O Algoritmo de Busca em Profundidade também é chamado de Depth-First Search (DFS), conforme Carmen et al. (2009), é usado para realizar

buscas em grafos, o algoritmo começa em um nó raiz e explora tanto quanto possível cada um dos seus ramos, antes de retroceder.

Segundo Even (2011) dado um grafo não direcionado $G = (V, E)$, onde n é o número de vértices e m é o número de arestas, a complexidade do DFS é dada por $O(n + m)$.

Figura 24 – Algoritmo de busca em profundidade modificado

- 1) Marque todos os vértices como não visitados
- 2) Inicialize uma lista para guardar os possíveis componentes desconexos do subgrafo
- 3) Para cada vértice v , se v não for visitado, inicialize uma lista para o conjunto de vértices conexos que inclui v e chame DFS(v)
- 4) DFS(v)
 - 4.a) Marque v como visitado
 - 4.b) Adicione v a lista do conjunto de vértices conexos
 - 4.c) Para cada vértice u adjacente de v , se u não for visitado, adicione u a lista do conjunto de vértices conexos que inclui v e chame recursivamente DFS(u)
- 5) Adicione a lista com o conjunto de vértices conexos a lista de componentes

Fonte: Adaptado de Even (2011).

Caso no subgrafo existam conjuntos de itens desconexos, para a aplicação de um Algoritmo de Dijkstra Modificado, Figura 25, tem-se como entrada a lista de adjacências com os vértices do grafo original e custo de atravessamento c_{ij} , com a distância necessária para percorrer a aresta entre os vértices (i, j) em uma direção.

Figura 25 – Algoritmo de Dijkstra modificado

- 1) Marque todos os vértices como não visitados e armazene-os.
- 2) Defina a distância como zero para o vértice inicial e infinito para os outros vértices.
- 3) Selecione o vértice não visitado com a menor distância, esse será o vértice atual.
- 4) Encontre vizinhos não visitados para o vértice atual e calcule suas distâncias através do vértice atual. Compare a distância recém-calculada com a atribuída e salve a menor.
- 5) Marque o vértice atual como visitado e remova-o do conjunto não visitado.
- 6) Pare, se o vértice de destino foi visitado ou se a menor distância entre os nós não visitados é infinito. Caso contrário, repita a etapa 3.

Fonte: Adaptado de Dijkstra (1959).

Dados dois conjuntos de vértices disjuntos em um *cluster*, a aplicação do Algoritmo de Dijkstra Modificado define todos os vértices de um conjunto como vértices iniciais e todos os vértices de um outro conjunto como vértices finais. Logo, executa-se o Algoritmo de Dijkstra Modificado para todas as combinações possíveis de vértices iniciais e finais, para todas as combinações obtém-se o conjunto de vértices e arestas que compõem o caminho mínimo que

tornam o subgrafo do *cluster* conexo. Do conjunto de caminhos mínimos possíveis, seleciona-se o caminho com a aresta ou as arestas que oferecem o menor custo de atravessamento c_{ij} e tornam o subgrafo do *cluster* conexo.

No Problema de Roteamento de Leituristas como às arestas existem tempos de passagem l_{ij} associados, ao tornar um *cluster* desconexo em um *cluster* conexo, o tempo de passagem de atravessar as arestas em ambos os lados, ou em um lado expresso por l_{ij} é adicionado à demanda do agrupamento.

O Algoritmo de Dijkstra Modificado difere do Algoritmo de Dijkstra por retornar o caminho mínimo entre vértices inicial e final previamente especificados, dessa forma, não é necessário percorrer todos os vértices de um grafo. Conforme Brassard e Bratley (1988) o Algoritmo de Dijkstra soluciona o problema do caminho mais curto em um grafo dirigido ou não dirigido com arestas de peso não negativo, a complexidade do Algoritmo de Dijkstra leva em consideração dois fatores: o tempo da procura por vértices não visitados com a menor distância e o tempo de atualização da distância. A implementação dessas operações requerem complexidade de $O(n)$ e $O(1)$, respectivamente. Logo, executa-se a primeira operação $O(n)$ vezes, e a segunda $O(m)$ vezes, obtendo a complexidade $O(n^2+m)$ para a algoritmo.

4.2.8 Realizar roteamento

Para cada subgrafo conexo é aplicado o Problema do Carteiro Chinês, dado o grafo associado o Problema do Carteiro Chinês pode ser Dirigido, Dirigido ou Misto.

4.3 MATERIAIS

O computador pessoal utilizado para a resolução dos problemas presentes nesse trabalho possui sistema operacional Windows 10 Home de 64 bits, com velocidade de processamento de 2.7 GHz, Memória RAM 8 GB, Processador Intel Core i5 do modelo INTEL CORE I5 5200U.

A linguagem de programação utilizada para a modelagem e solução dos problemas foi a *Python 3*, onde destaca-se o uso de dois pacotes: o *scipy.spatial* e o *PuLP*.

O pacote *scipy.spatial* é uma biblioteca Python voltado para aplicações matemáticas, científicas e de engenharia, o pacote é utilizado para calcular a distância entre cada par de duas coordenadas geográficas (SCIPY, 2020). O *PuLP* é uma biblioteca do Python para estruturação e resolução de problemas de Programação Linear (PULP, 2020).

O pacote `scipy.spatial` teve como entrada as coordenadas geográficas (latitude e longitude) de todos os pontos médios das arestas e como saída a matriz de distâncias euclidianas entre todos os pontos. O PuLP foi utilizado para modelar e aplicar o generic Capacitated p -Median Problem.

Na implementação do Algoritmo de Busca em Profundidade Modificado e do Algoritmo de Dijkstra Modificado fez-se uso da linguagem *Python 3*. O *Python 3* também foi utilizado para escrever o Problema do Carteiro Chinês de cada *cluster* em lp. Posteriormente, o arquivo lp foi resolvido por meio do software Soplex/SCIP.

A ferramenta SoPlex (Sequential object-oriented simPlex) é um pacote de otimização para resolver Problemas de Programação Linear (PPLs) com base em uma implementação avançada do algoritmo simplex oferecendo o primal e o dual. O SoPlex foi desenvolvido no Zuse-Institute Berlin (SCIP, 2020).

Os problemas podem ser escrito em arquivos no formato MPS ou LP, posteriormente esses arquivos são lidos e interpretados pela ferramenta SoPlex, que realiza os cálculos e oferece as soluções.

Dada uma aresta (i, j) , o *Google Maps* foi utilizado para estabelecer o tempo de passagem l_{ij} em minutos, o custo de atravessamento c_{ij} dado em metros e a quantidade de pontos de leitura $QtPontos$. O *Google Maps* foi utilizado para contar o número de pontos de leitura presentes em cada segmento de rua. A partir do valor da quantidade de pontos e do tempo médio de vistoria, estabelece-se o valor de v_{ij} e o valor de w_{ij} é expresso pela Equação 4.1.

5 ABORDAGEM PROPOSTA

Neste capítulo será exemplificada a abordagem para roteamento de leituristas baseada no procedimento *cluster first - route second* proposto. Inicialmente foi decalcado uma parte da rede urbana da cidade de Recife para a aplicação da abordagem. O decalque pode ser visto na Figura 26. Nela as linhas retas são ruas e as partes em cinza são o espaço urbano dedicado a construção civil.

Figura 26 – Decalque do espaço urbano



Fonte: Google Maps (2020).

À parte decalcada foram atribuídos 15 vértices e 19 arestas, os vértices foram numerados de 1 a 15 e as arestas da letra *a* até a letra *t*, conforme mostra a Figura 27.

Figura 27 – Decalque do espaço urbano com vértices e arestas (a) e pontos médios (b)



Fonte: Google Maps (2020)

Na Figura 26, os vértices estão destacados em pontos vermelhos sobre a malha urbana. O ponto médio de cada aresta foi identificado e marcado na cor azul. A partir da localização

foram retiradas as coordenadas geográficas (latitude e longitude) dos pontos médios por meio do *Google Maps*.

A Tabela 1 mostra as arestas que formam o grafo do território estudado, os vértices associados as arestas e as coordenadas geográficas dos pontos médios de cada aresta. Para cada segmento de vértice (i, j) , o tempo de passagem de ambos os lados do segmento é expresso por l_{ij} , o tempo de vistoria de ambos os lados é v_{ij} , o tempo de atravessamento é w_{ij} . Nesse caso, c_{ij} é o custo de atravessamento na direção (i, j) , como no problema ambos os lados das arestas devem ser visitados, a aresta não dirigida (i, j) pode ser tratada como dirigida com uma aresta saindo e outra entrando em cada vértice.

O valor de l_{ij} para cada aresta foi captado por meio do *Google Maps* e é dado em minutos, ainda com o auxílio do *Google Maps* foi estabelecido o valor de $QtPontos$, que é a quantidade de residências em ambos os lados dos segmentos de rua que compõem as arestas do grafo. Posteriormente, estabeleceu-se o valor de $TpLeitura$ como 1 minuto, $TpLeitura$ representa o tempo que o leitorista consome para realizar a leitura de um ponto de consumo. Por meio da Equação (4.2) estabeleceu-se o valor de v_{ij} em minutos para cada aresta entre qualquer vértice (i, j) , w_{ij} corresponde a soma dos valores de l_{ij} e v_{ij} para cada par ordenado (i, j) em ambos os lados, conforme a equação (4.1). O valor de c_{ij} em metros para cada aresta, também foi extraído no *Google Maps*.

Foi estabelecido o valor arbitrário de 90 minutos como a capacidade trabalho de cada leitorista, para fins de exemplificação. Somou-se o tempo de atravessamento w_{ij} para cada aresta e obteve-se o valor w_{ij} total de 294 minutos.

Tabela 1 - Atributos do espaço urbano estudado

Arestas	Vértices (i, j)	Coordenadas	c_{ij} (m.)	l_{ij} (min.)	v_{ij} (min.)	w_{ij} (min.)
{ a }	{ 1, 2 }	-8.060828, -34.941134	64	2	13	15
{ b }	{ 2, 3 }	-8.061303, -34.941248	51	2	7	9
{ c }	{ 3, 4 }	-8.061742, -34.941342	46	2	7	9
{ d }	{ 4, 5 }	-8.061855, -34.941690	69	2	7	9
{ e }	{ 5, 6 }	-8.061684, -34.942287	80	2	8	10
{ f }	{ 6, 7 }	-8.061318, -34.942667	47	2	7	9
{ g }	{ 7, 8 }	-8.060880, -34.942604	49	2	3	5
{ h }	{ 3, 7 }	-8.061340, -34.941955	160	4	14	18
{ i }	{ 2, 8 }	-8.060901, -34.941830	160	4	16	20
{ j }	{ 1, 9 }	-8.060359, -34.941658	160	4	34	38
{ k }	{ 8, 9 }	-8.060355, -34.942546	64	4	4	8
{ l }	{ 6, 15 }	-8.061475, -34.942928	58	2	11	13

(continua)

Tabela 1 - Atributos do espaço urbano estudado

(conclusão)

Arestas	Vertices (i, j)	Coordenadas	c_{ij} (m.)	l_{ij} (min.)	c_{ij} (min.)	w_{ij} (min.)
{ m }	{ 14, 15 }	-8.061475, -34.942928	56	2	13	15
{ n }	{ 13, 14 }	-8.061207, -34.943889	55	2	7	9
{ o }	{ 12, 13 }	-8.060881, -34.944131	50	2	9	11
{ p }	{ 7, 12 }	-8.060905, -34.943320	170	4	19	23
{ q }	{ 11, 12 }	-8.060434, -34.944067	51	2	8	10
{ r }	{ 8, 11 }	-8.059910, -34.943116	170	4	11	15
{ s }	{ 10, 11 }	-8.059921, -34.944000	63	2	7	9
{ t }	{ 9, 10 }	-8.059884, -34.943222	170	2	35	39
Total						294

Fonte: O autor (2020).

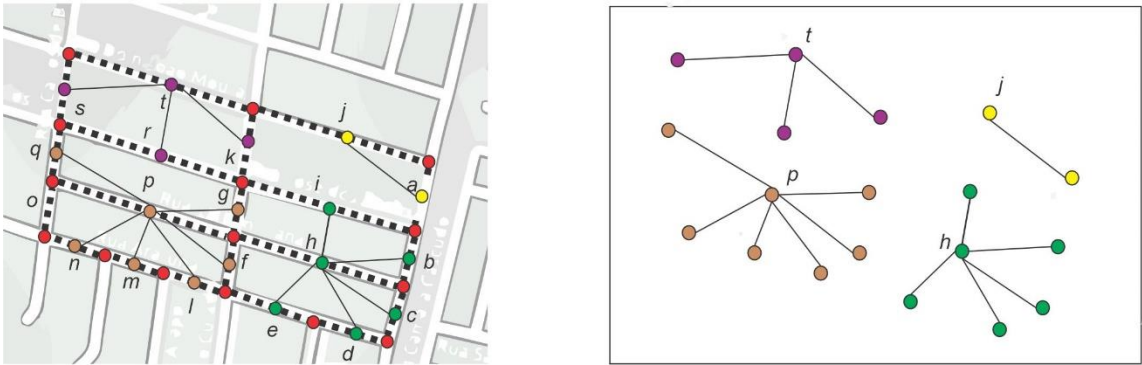
A partir das coordenadas geográficas (latitude e longitude) dos pontos médios de cada aresta utilizou-se a Fórmula de Haversine (Equação 2.32) para construir a matriz de distâncias euclidianas entre cada ponto médio das arestas, a matriz de distância foi estabelecida por meio pacote `scipy.spatial` do Python.

A quantidade de agrupamentos, leituristas ou *clusters* corresponde ao tempo de atravessamento total, dado pela soma do valor w_{ij} de cada aresta, dividido pela capacidade de trabalho do leiturista, essa divisão obteve como resultado o valor decimal aproximado de 3,267, esse valor foi aproximado para o próximo inteiro positivo. Dessa forma, a quantidade de leituristas foi estabelecido como 4.

De posse da matriz de distância entre as arestas, do número de leituristas a serem alocados, da capacidade de cada leiturista e da necessidade de atendimento de cada aresta, expressa por w_{ij} aplicou-se o problema das p -medianas conhecido como *gCpMP*. Para o *gCpMP* como resultado foram gerados 4 *clusters*, para cada *cluster* uma aresta foi escolhida como mediana.

A Figura 28 (a) parte a mostra a área decalcada para o exemplo, nela temos os vértices do grafo em vermelho e os pontos médios de cada aresta destacados em cores diferentes. Existem 4 agrupamentos, nas cores violeta, marrom, verde e amarela. Na Figura 28 (b) ver-se que cada agrupamento possui um conjunto com arestas disjuntas, mas podem compartilhar vértices semelhantes.

Figura 28 – Clusters na rede urbana (a) e clusters (b)



Fonte: O autor (2020).

A cada mediana foram atribuídos um conjunto de arestas respeitando a máxima capacidade que cada leitorista possui. A Tabela 2 mostra os *clusters*, as arestas escolhidas como mediana, as arestas associadas a esse mediana, os vértices de cada aresta associada a mediana, bem como os valores w_{ij} .

Tabela 2 – Clusters e demandas

Cluster	Mediana	Arestas	Vértices (i,j)	w_{ij}
1	{ h }	{ b }	{ 2,3 }	9
		{ c }	{ 3,4 }	9
		{ d }	{ 4,5 }	9
		{ e }	{ 5,6 }	10
		{ f }	{ 6,7 }	9
		{ h }	{ 3,7 }	18
		{ i }	{ 2,8 }	20
Total				84
2	{ j }	{ j }	{ 1,9 }	15
		{ a }	{ 1,2 }	38
Total				53
3	{ p }	{ g }	{ 7,8 }	5
		{ l }	{ 6,15 }	13
		{ m }	{ 14,15 }	15
		{ n }	{ 13,14 }	9
		{ o }	{ 12,13 }	11
		{ p }	{ 7,12 }	23
		{ q }	{ 11,12 }	10
Total				86

(continua)

Tabela 2 – Clusters e demandas

(conclusão)

Cluster	Mediana	Arestas	Vértices (i,j)	w_{ij}
4	{ t }	{ k }	{ 8,9 }	8
		{ r }	{ 8,11 }	15
		{ s }	{ 10,11 }	9
		{ t }	{ 9,10 }	39
Total				71

Fonte: O autor (2020).

Conforme pode-se ver na Tabela 2, para cada *cluster* a demanda total de atendimento de cada agrupamento não superou a capacidade de atendimento de 90 minutos estabelecida para cada leitorista. Para o estabelecimento dos valores do tempo de atravessamento w_{ij} em minutos foi considerado o percurso em ambos os lados de um segmento, ou aresta (i, j), dessa forma os subgrafos de cada agrupamento podem ser considerados como um grafo direcionado. O roteamento em cada subgrafo pode ser semelhante a encontrar um circuito Euleriano. O custo de atravessamento em metros c_{ij} , por sua vez, corresponde a distância de percorrer entre os pontos (i, j).

Após a aplicação do *gCpMP* e antes de realizar o roteamento é necessário verificar se o subgrafo de cada agrupamento é conexo. Caso o subgrafo não seja conexo é necessário adicionar as arestas necessárias para tornar o subgrafo conexo. A aresta ou as arestas a serem adicionadas são aquelas que tornam os componentes do subgrafo conexos e oferecem o caminho com os menores valores de c_{ij} . Para testar a conectividade dos subgrafos do problema utilizou-se o Algoritmo de Busca em Profundidade Modificado, e logo identificou-se que todos os 4 subgrafos do problema são conexos. Com a conectividade verificada, os 4 clusters ou agrupamentos foram tratados como subgrafos com arestas direcionadas, o Problema do Carteiro Chinês Direcionado foi aplicado e obteve os resultados presentes na Tabela 3. A Tabela 3 mostra os clusters obtidos, a demanda de cada agrupamento e a distância mínima obtida após a aplicação do PCC.

Tabela 3 – Clusters

Cluster	Demanda	Distância
1	84	1226
2	53	448
3	86	978
4	71	934

Fonte: O autor (2020).

6 RESULTADOS E DISCUSSÕES

Este capítulo 6 apresenta a aplicação da abordagem *cluster first – route second* proposta para o roteamento de leituristas em um bairro da cidade de Recife e em uma cidade do interior do Pernambuco, Flores, e apresenta discussões sobre as aplicações.

No bairro do Engenho do Meio em Recife e em uma primeira aplicação na cidade de Flores considerou-se que o leiturista deve percorrer toda a malha urbana indo e vindo por todos os segmentos de ruas que formam as arestas. Caso um segmento de rua não possua casas para vistoria em ambos os lados, apenas a demanda de passagem l_{ij} de cada lado em minutos é considerada. No caso em que existem casas em um lado da rua e não em outro, do lado em que existem casas é computado a demanda de passagem l_{ij} e o tempo de vistoria v_{ij} , já do outro apenas a demanda de passagem l_{ij} .

Dadas as condições acima, caso o subgrafo de um *cluster* seja desconexo, e determinada aresta torna o *cluster* conexo, o tempo de passagem l_{ij} de ir e vir em ambos os lados da aresta é adicionado a demanda total do *cluster*.

No Engenho do Meio e na primeira aplicação em Flores em cada *cluster* o roteamento ocorre por meio da aplicação do Problema do Carteiro Chinês Direcionado, visto que uma aresta não direcionada pode ser modelada por dois arcos.

Na segunda aplicação para a cidade de Flores, dada a malha urbana, caso uma rua não apresente pontos de vistoria em nenhum dos lados, o tempo de passagem não é computado. Caso existam casa em apenas um lado da rua, apenas desse lado é computado a demanda de passagem l_{ij} e o tempo de vistoria v_{ij} .

Dada essas características, caso o subgrafo de um *cluster* seja desconexo, e determinada aresta torna o *cluster* conexo, o tempo de passagem l_{ij} de passagem em apenas um lado da rua é computado e adicionado ao *cluster*.

Para o Engenho do Meio e o primeiro caso de Flores, ao ter de percorrer todas as arestas indo e vindo, todos os vértices dos *clusters* terão grau par, assim como os vértices das arestas que podem tornar um *cluster* desconexo em conexo, mesmo que algumas arestas não tenham pontos de vistoria e apenas demanda de passagem l_{ij} .

Para o segundo caso de Flores, em alguns casos apenas um lado da rua terá a demanda captada, nesse caso a aresta é não direcionada. Ao tornar um *cluster* conexo, como não é necessário percorrer ambos os lados, o tempo de passagem de apenas um dos lados das arestas que tornam um *cluster* conexo é computado.

Para o estudo foram criados dois cenários em que os leituristas tem carga de trabalho de 6 horas ou 360 minutos e 8 horas ou 480 minutos. Para ambos os cenários, criaram-se situações em que variou-se o tempo de leitura de cada ponto de consumo em 0.5 segundos, 0.7 segundos, 1 minuto, e em uma quarta situação para ambos os cenários considerou-se o tempo de 1 minuto para leitura e para cada aresta foi adicionado o valor extra de 1 minuto para cobrir alguma dificuldade de passagem. Os dois cenários e as quatro situações foram combinados, o que gerou oito casos possíveis presentes na Tabela 4:

Tabela 4 – Relação de casos para o problema do Engenho do Meio

Casos	Carga de trabalho do leiturista	Tempo de vistoria por relógio
1	480	0.5 s.
2	360	
3	480	0.7 s.
4	360	
5	480	1 min.
6	360	
7	480	1 min. + 1 min. para a travessamento de cada via
8	360	

Fonte: O autor (2020).

Para cada segmento de rua ou aresta (i, j) foi computado o número de residências $QtPontos$ e aplicou-se a Equação 4.2 para estabelecer o valor de v_{ij} em cada aresta.

Os valores v_{ij} e, conseqüentemente, w_{ij} são influenciados diretamente pelo tempo médio que o leiturista leva para fazer a leitura, visto que os valores de v_{ij} para cada aresta entre os pares ordenados (i, j) é a multiplicação entre o tempo de vistoria de um relógio e a quantidade de relógios presentes em ambos os locais do segmento de rua. Os valores de w_{ij} , por sua vez, são a soma dos valores de passagem, l_{ij} e o valor de v_{ij} de cada par ordenado (i, j) .

A Tabela 5 mostra os dois cenários propostos com tempos de trabalho de 480 e 360 minutos, para cada cenário quatro situações são idealizadas, onde os tempos de vistoria dos relógios que medem o consumo podem diferir.

Tabela 5 – Engenho do meio: cenários e situações

Tempo por vistoria	Cenário			
	480 min.		360 min	
	$\sum w_{ij}$	P	$\sum w_{ij}$	p
0.5 s.	2227	5	2227	7
0.7 s.	3519	8	3519	10
1 min.	3703	8	3703	11
1 min. + 1 min. para a travessamento de cada via	3910	9	3910	12

Fonte: O autor (2020).

Para cada cenário e situação a quantidade de *clusters* ou leituristas podem variar, para cada caso, a quantidade de agrupamentos é dado pela soma do valor de w_{ij} de todas as arestas do grafo dividida pela capacidade de atendimento de cada leiturista, como a divisão gera um número fracionário, arredonda-se esse número, conforme a proposta de abordagem do Capítulo 4, logo a Tabela 5 mostra o número de medianas, expresso pela letra p , para cada cenário e situação.

Para cada aresta foram identificados os pontos médios, de cada ponto médio foram retiradas a latitude e a longitude por meio do *Google Maps*. As coordenadas geográficas foram tratadas por meio da biblioteca *scipy.spatial* do *Python*, essa ferramenta aplica a Fórmula de Haversine as coordenadas e permite encontrar a matriz quadrada das distâncias euclidianas dos pontos médio.

Após a obtenção da matriz de distâncias, do número de medianas ou leituristas a serem alocados, da definição das capacidades de cada leiturista e do tempo de atravessamento de cada vértice, w_{ij} , e do custo atravessamento c_{ij} , aplica-se o *gCpMP*.

Para os dois cenários e as quatro situações obtiveram-se os resultados das Tabelas 6, 7, 8 e 9. As Tabelas 6, 7, 8 e 9 tratam respectivamente das situações em que o tempo de vistoria de cada relógio medidor de consumo é de 0.5 segundos, 0.7 segundos, 1 minuto e 1 minuto, com a adição de mais 1 minuto por aresta. Para os cenários com carga de trabalho de 360 minutos e 480 minutos em cada situação observa-se nas tabelas o número de *clusters* ou agrupamentos, as demandas de atendimento de todas as arestas ou segmentos de rua presentes em um *cluster*. As Tabelas 6, 7, 8 e 9 também expõem o tempo de processamento consumido para a obtenção dos *clusters* ou agrupamentos mediante a resolução do *gCpMP* por meio do *Python*.

Tabela 6 – Engenho do Meio: gCpMP para os casos 1 e 2

Cluster	480 min.	360 min.
	Demanda (min.)	Demanda (min.)
1	480	300
2	478	265
3	479	323
4	392	356
5	398	360
6		312
7		311
Tempo de Resolução (s.)	271.26	26.04

Fonte: O autor (2020).

Tabela 7 – Engenho do Meio: gCpMP para os casos 3 e 4

Cluster	480 min.	360 min.
	Demanda (min.)	Demanda (min.)
1	479	346
2	477	353
3	478	357
4	460	355
5	344	359
6	325	357
7	478	354
8	478	339
9		343
10		356
Tempo de Resolução (s.)	157.13	1289.83

Fonte: O autor (2020).

Tabela 8 – Engenho do Meio: gCpMP para os casos 5 e 6

Cluster	480 min.	360 min.
	Demanda (min.)	Demanda (min.)
1	479	307
2	480	357
3	479	309
4	480	352
5	445	354
6	441	355
7	419	291
8	480	341
9		352
10		333

(continua)

Tabela 8 – Engenho do Meio: gCpMP para os casos 5 e 6

(conclusão)

Cluster	480 min. Demanda (min.)	360 min. Demanda (min.)
11		352
Tempo de Resolução (s.)	523.52	1051.08

Fonte: O autor (2020).

Tabela 9 – Engenho do Meio: gCpMP para os casos 7 e 8

Cluster	480 min. Demanda (min.)	360 min. Demanda (min.)
1	479	308
2	476	356
3	418	301
4	353	215
5	450	352
6	447	350
7	433	352
8	375	261
9	479	343
10		353
11		359
12		360
Tempo de Resolução (s.)	89.32	582.79

Fonte: O autor (2020).

Para cada cenário e situação estabelecida, após o agrupamento das arestas, a conectividade de cada agrupamento foi verificada por meio do Algoritmo de Busca em Profundidade Modificado, conforme mostram as Tabelas 10, 12, 15 e 16. Caso o subgrafo do agrupamento seja conexo, o algoritmo gera um conjunto com todos os vértices do subgrafo, caso contrário gera mais de um conjunto de vértices, onde cada conjunto de vértices é disjunto um do outro.

Para os agrupamentos com componentes desconexos aplica-se o Algoritmo de Dijkstra Modificado, que oferece o menor caminho entre dois nós de conjuntos disjuntos de vértices. O objetivo dessa aplicação seria encontrar as arestas que oferecem um caminho com o menor custo de passagem total, l_{ij} , que conecte os dois conjuntos desconexos e torne o subgrafo não direcionado de cada agrupamento conexo. O custo utilizado no Algoritmo de Dijkstra

Modificado para cada aresta (i, j) será o custo l_{ij} , uma vez que esse custo é apenas o custo de passar em ambos os lados da via e não o de verificar os pontos de leitura.

Dado um subgrafo com dois conjuntos de vértices desconexos. Para cada vértice de um primeiro componente desconexo, dado o grafo original, todos os vértices serão testados como vértices iniciais, no segundo componente desconexo todos os vértices serão estabelecidos como vértices finais. Aplicando-se o Algoritmo de Dijkstra Modificado para testar todas as possíveis combinações de vértices iniciais e finais, e para todas as combinações será encontrada a combinação que oferece o caminho com o menor tempo de atravessamento em ambos os lados de uma aresta (i, j) .

A Tabela 10 mostra o tempo gasto em segundos para verificar a conectividade de cada agrupamento individualmente utilizando o Algoritmo de Busca em Profundidade Modificado para a situação em que cada ponto de leitura leva 0.5 segundos para ser lido nos cenários em que o leitor tem capacidade de atendimento de 480 e 360 minutos.

Tabela 10 – Engenho do Meio: DFS modificado para os casos 1 e 2

Cluster	480 min.	360 min.
	Tempo de Resolução (s.)	
1	0.995	0.829
2	0.849	0.717
3	0.699	0.777
4	0.723	0.787
5	0.816	0.793
6		0.713
7		0.844

Fonte: O autor (2020).

Para essa situação em que o tempo de vistoria de um relógio é 0.5 segundos e a capacidade de trabalho de 480 minutos apenas o *cluster* 1 mostrou-se desconexo. O agrupamento é composto por dois conjuntos de vértices sem ligação entre si, um conjunto formado pelos vértices {27, 28, 37, 38, 39, 40, 41, 44, 45, 46, 49, 50, 51, 34, 35, 36, 33, 52, 48, 68, 80, 67, 69, 70, 71, 83, 95, 82, 94, 81, 42, 43, 47} e outro pelos vértices {32, 54}.

Para tornar o *cluster* 1 conexo é necessário adicionar a aresta entre os vértices {33,32}, essa constatação foi verificada a partir da aplicação do Algoritmo de Dijkstra Modificado a todos os vértices dos dois conjuntos disjuntos de nós. Dado o tempo de passagem l_{ij} para todas as arestas do grafo original, a aresta entre os vértices oferece o tempo de passagem de 2 minutos.

A Tabela 11 mostra o custo de atendimento em minutos e a distância percorrida em cada *cluster* antes da adição da aresta entre os vértices {33,32}, e o custo de atendimento e a distância

percorrida com a adição da aresta, além do tempo computacional que o Algoritmo de Dijkstra Modificado consumiu para chegar a conclusão de que aresta deveria ser adicionada.

Tabela 11 – Engenho do Meio: cluster 1 para o caso 1

Cluster	480 min.		
	Demanda (min.)	Demanda (min.)	Tempo de Resolução (s.)
1	480	482	1.737

Fonte: O autor (2020).

Tabela 12 – Engenho do Meio: DFS modificado para os casos 3 e 4

Cluster	480 min.	360 min.
	Tempo de Resolução (s.)	
1	0.73	0.71
2	0.736	0.67
3	0.74	0.889
4	0.698	0.879
5	0.739	0.731
6	0.781	0.793
7	0.718	0.761
8	0.681	0.768
9		0.775
10		0,759

Fonte: O autor (2020).

Para a situação com vistoria de 0.7 segundos e tempo de trabalho de 480 minutos, o *cluster* 2 tem dois conjuntos disjuntos de vértices, um formado pelos vértices {58, 72, 84, 96, 97, 85, 73, 59, 60, 61, 74, 86, 98, 99, 87, 75, 62, 63, 64, 76, 88, 100} e outro pelos vértices {111, 112}, ao adicionar as arestas entre os pares de vértices {(100, 101), (101, 108), (108, 112)} o *cluster* 2 torna-se conexo, conforme o Algoritmo de Dijkstra Modificado executado. A Tabela 13 mostra o custo de atendimento em minutos no *cluster* 2 antes da adição das arestas destacadas e depois, assim como o tempo de execução do Algoritmo de Dijkstra Modificado.

Tabela 13 – Engenho do Meio: cluster 2 para o caso 3

Cluster	480 min.		
	Demanda (min.)	Demanda (min.)	Tempo de Resolução (s.)
2	477	489	1.595

Fonte: O autor (2020).

Para a situação com vistoria de 0.7 segundos e tempo de trabalho de 360 minutos, dois *clusters* são desconexos, os *clusters* 2 e 4. O *cluster* número 2 tem dois conjuntos disjuntos de vértices, um com os vértices {1, 2, 5, 6, 23, 25, 4, 26, 24, 9, 10, 7, 3, 11, 13, 8} e outro com os vértices {12, 14}, ao adicionar a aresta entre os vértices {9, 14}, o subgrafo torna-se conexo, a demanda total do *cluster* é adicionado o valor de 2 minutos.

O *cluster* número 4 tem dois componentes desconexos, um formado pelos vértices {56, 58, 72, 84, 96, 97, 85, 73, 59, 60, 61, 74, 86, 98, 99, 87, 75, 62, 63, 64, 76, 88, 100, 101, 89, 77, 65, 90} e outro pelos vértices {66, 78}, respectivamente, ao adicionar a aresta entre os vértices {77, 78}, a demanda de tempo total para percorrer todas as arestas do *cluster* aumenta em 2 minutos.

A Tabela 14 mostra o custo de atendimento em minutos dos *clusters* 2 e 4 antes da adição das arestas sugeridas pelo Algoritmo de Dijkstra Modificado e depois da adição, assim como o tempo que o Algoritmo de Dijkstra Modificado consumiu para oferecer a resposta em cada caso.

Tabela 14 – Engenho do Meio: cluster 2 e 4 para o caso 4

Cluster	360 min.		
	Demanda (min.)	Demanda (min.)	Tempo de Resolução (s.)
2	353	355	1.558
4	355	357	1.53

Fonte: O autor (2020).

A Tabela 15 mostra o tempo que o Algoritmo de Busca em Profundidade Modificado consumiu para identificar se os *clusters* para as situações em que o tempo de vistoria era de 1 minuto para os cenários onde o tempo de trabalho de cada leiturista era de 480 minutos e 360 minutos. Para esse caso, nenhum *cluster* foi identificado como desconexo.

Tabela 15 – Engenho do Meio: DFS modificado para os casos 5 e 6

Cluster	480 min.	360 min.
	Tempo de Resolução (s.)	
1	0.784	0.75
2	0.698	0.758
3	0.822	0.719
4	0.712	0.743
5	0.959	0.759
6	0.736	0.697

(continua)

Tabela 15 – Engenho do Meio: DFS modificado para os casos 5 e 6

(conclusão)

Cluster	480 min.	360 min.
	Tempo de Resolução (s.)	
7	0.92	0.873
8	0.745	0.737
9		0.758
10		0.729
11		0.698

Fonte: O autor (2020).

A Tabela 16 mostra o tempo que o Algoritmo de Busca em Profundidade Modificado consumiu para identificar se os *clusters* para as situações em que o tempo de vistoria era de 1 minuto para os cenários com 1 minuto de acréscimo para cada aresta (i, j) onde o tempo de trabalho de cada leitorista era de 480 minutos e 360 minutos.

Tabela 16 – Engenho do Meio: DFS modificado para os casos 7 e 8

Cluster	480 min.	360 min.
	Tempo de Resolução (s.)	
1	0.678	0.75
2	0.79	0.758
3	0.718	0.719
4	0.945	0.743
5	0.90	0.759
6	0.652	0.697
7	0.612	0.873
8	0.747	0.737
9	0.756	0.758
10		0.729
11		0.754
12		0.698

Fonte: O autor (2020).

Para a situação com vistoria de 1 minuto por relógio de medição adicionado de 1 minuto por aresta e tempo de trabalho de 480 minutos, um *cluster* é desconexo, nesse caso, o *cluster* número 2. Esse *cluster* tem dois componentes, um formado pelo conjunto de vértices {58, 72, 84, 96, 97, 85, 73, 59, 60, 61, 74, 86, 98, 99, 87, 75, 62, 63, 64, 76, 88, 100, 101, 89, 77, 65, 90, 102} e outro pelo conjunto de vértice {66, 78}, ao adicionar a aresta entre os vértices {77, 78} o subgrafo do *cluster* torna-se conexo, dessa forma, na demanda são adicionados 2 minutos, conforme pode-se ver na Tabela 17.

Tabela 17 – Engenho do Meio: cluster 2 para o caso 7

Cluster	480 min.		
	Demanda (min.)	Demanda (min.)	Tempo de Resolução (s.)
2	476	478	1.771

Fonte: O autor (2020).

Para a situação onde o tempo de vistoria é de 1 minuto por relógio adicionado de 1 minuto por aresta e a carga de trabalho do leiturista é de 360 minutos, apenas um *cluster* é desconexo, o *cluster* número 6 tem dois conjuntos de vértices desconexos, um contendo os vértices {104, 109, 108, 110, 120, 121, 118, 117} e outro contendo os nós {123, 132}, para tornar o subgrafo conexo é necessário adicionar a aresta entre os vértices {121, 132}, essa aresta tem um tempo de percurso de 2 minutos, conforme pode-se verificar na Tabela 18.

Tabela 18 – Engenho do Meio: cluster 2 para o caso 8

Cluster	360 min.		
	Demanda (min.)	Demanda (min.)	Tempo de Resolução (s.)
6	350	352	2.505

Fonte: O autor (2020).

Após testada a conectividade para cada *cluster* presente em cada cenário e situação, as arestas necessárias para tornar os subgrafos conexos em cada *cluster* foram adicionadas. Posteriormente, cada subgrafo de cada *cluster* foi tratado como um grafo direcionado, então aplicou-se o Problema do Carteiro Chinês Direcionado. Para cada *cluster* a resolução foi instantânea mediante o uso do SCIP. As Tabelas 19, 20, 21 e 22 mostram o Problema do Carteiro Chinês Direcionado resolvido para cada *cluster*.

Tabela 19 – Engenho do Meio: PCC para os casos 1 e 2

Cluster	480 min.	360 min.
	Distância (m.)	Distância (m.)
1	8620	5688
2	11394	6724
3	8692	8580
4	5910	3442
5	4272	4652
6		5284
7		5088

Fonte: O autor (2020).

Tabela 20 – Engenho do Meio: PCC para os casos 3 e 4

Cluster	480 min.	360 min.
	Distância (m.)	Distância (m.)
1	6986	4680
2	6810	4934
3	5774	5798
4	4466	4282
5	1684	1964
6	2960	2784
7	5080	3040
8	5024	3782
9		3996
10		3762

Fonte: O autor (2020).

Tabela 21 – Engenho do Meio: PCC para os casos 5 e 6

Cluster	480 min.	360 min.
	Distância (m.)	Distância (m.)
1	7242	4128
2	7830	5586
3	5392	5272
4	4614	3378
5	4106	3754
6	1684	3054
7	3320	2498
8	4596	2820
9		1124
10		3790
11		3380

Fonte: O autor (2020).

Tabela 22 – Engenho do Meio: PCC para os casos 7 e 8

Cluster	480 min.	360 min.
	Distância (m.)	Distância (m.)
1	7160	4198
2	6744	5040
3	4366	4746
4	3790	2198
5	3634	3434
6	1684	2964
7	3320	2694
8	3760	2136
9	4452	844

(continua)

Tabela 22 – Engenho do Meio: PCC para os casos 7 e 8

(conclusão)

Cluster	480 min.	360 min.
	Distância (m.)	Distância (m.)
10		3790
11		3270
12		3558

Fonte: O autor (2020).

6.2 FLORES 1

A cidade de Flores é um município de pequeno porte localizada no interior de Pernambuco, essa cidade possui 63 ruas. A Figura 30 mostra a vista aérea da cidade de Flores.

Figura 30 – Vista aérea da cidade de Flores



Fonte: Google Maps (2020).

No grafo $G = (V, A)$ associado a malha urbana da cidade, os vértices representam locais onde o leitorista pode atravessar de um segmento de rua para o mesmo lado do mesmo segmento ou para um outro segmento adjacente, as arestas são os segmentos de reta entre os vértices. Para o grafo G , que representa a cidade, o número de vértices é dado por $|V| = 160$, a quantidade de arestas $|A| = 104$.

Para o estudo foi criado um cenário em que os leitoristas tem carga de trabalho de 8 horas ou 480 minutos. Para esse cenário variou-se o tempo de leitura de cada ponto de consumo em 1 minuto, 0.7 segundos e 0.5 segundos, em uma quarta situação considerou-se o tempo de

1 minuto para leitura e para cada aresta foi adicionado o valor extra de 1 minuto para cobrir alguma possível dificuldade de passagem.

Para cada segmento de rua ou aresta (i, j) foi computado o número de residências $QtPontos$ e aplicou-se a Equação 4.2, para estabelecer o valor de v_{ij} em cada aresta.

Os valores v_{ij} e, consequentemente, w_{ij} são influenciados diretamente pelo tempo médio que o leitorista leva para fazer a leitura, visto que os valores de v_{ij} para cada aresta entre os pares ordenados (i, j) é a multiplicação entre o tempo de vistoria de um relógio e a quantidade de relógios presentes em ambos os locais do segmento de rua. Os valores de w_{ij} , por sua vez, são a soma dos valores de passagem, l_{ij} e o valor de v_{ij} de cada par ordenado (i, j) , Equação 4.1.

Para cada aresta foram identificados os pontos médios, de cada ponto médio foram retiradas a latitude e a longitude por meio do *Google Maps*. As coordenadas geográficas foram tratadas por meio da biblioteca *scipy.spatial* do *Python*, essa ferramenta aplica a Fórmula de Haversine aos dados e permite encontrar a matriz quadrada das distâncias euclidianas para todas as coordenadas.

Combinou-se o cenário proposto e as situações estabelecidas e compõe-se quatro casos, conforme pode-se ver na Tabela 23.

Tabela 23 – Relação de casos para o problema de Flores

Casos	Carga de trabalho do leitorista	Tempo de vistoria por relógio
1	480	0.5 s.
2		0.7 s.
3		1 min.
4		1 min. + 1 min. para a travessamento de cada via

Fonte: O autor (2020).

A Tabela 24 mostra o cenário proposto para o tempo de trabalho de 480 minutos, para o cenário quatro situações são idealizadas, onde os tempos de vistoria dos relógios que medem o consumo podem variar. Para cada cenário e situação a quantidade de *clusters* ou leitoristas podem variar, para cada caso, a quantidade de agrupamentos é definido pela soma do valor de w_{ij} de todas as arestas do grafo dividida pela capacidade de atendimento de cada leitorista, como a divisão gera um número fracionário, arredonda-se esse número.

Tabela 24 – Flores 1: cenário e situações

Tempo por vistoria	480 min.	
	$\sum w_{ij}$	p
0.5 s.	1754	4
0.7 s.	2268	5
1 min.	2984	7
1 min. + 1 min. para a travessamento de cada via	3144	7

Fonte: O autor (2020).

Após a obtenção da matriz de distâncias, do tempo de atravessamento de cada vértice, w_{ij} , do custo atravessamento c_{ij} , do número de medianas ou leituristas a serem alocados, da definição das capacidades de cada leiturista, aplica-se o $gCpMP$.

As Tabelas 25, 26, 27 e 28 tratam respectivamente das situações em que o tempo de vistoria de cada relógio medidor é de 0.5 segundos, 0.7 segundos, 1 minuto e 1 minutos, com a adição de mais 1 minuto por aresta. Para o cenário com carga de trabalho de 480 minutos em cada situação observa-se nas Tabelas 25, 26, 27 e 28 o número de *clusters* ou agrupamentos, as demandas de atendimento de todas as arestas ou segmentos de rua presentes em um *cluster*. As Tabelas 25, 26, 27 e 28 também expõem o tempo de processamento consumido para a obtenção dos *clusters* ou agrupamentos por meio do $gCpMP$ modelado e resolvido mediante a biblioteca PuLP do *Python*.

Tabela 25 – Flores 1: $gCpMP$ para o caso 1

Cluster	480 min.
	Demanda (min.)
1	479
2	349
3	446
4	480
Tempo de Resolução (s.)	25.78

Fonte: O autor (2020).

Tabela 26 – Flores 1: gCpMP para o caso 2

Cluster	480 min.
	Demanda (min.)
1	448
2	456
3	478
4	410
5	476
Tempo de Resolução (s.)	42.64

Fonte: O autor (2020).

Tabela 27 – Flores 1: gCpMP para o caso 3

Cluster	480 min.
	Demanda (min.)
1	444
2	412
3	425
4	475
5	479
6	353
7	396
Tempo de Resolução (s.)	29.57

Fonte: O autor (2020).

Tabela 28 – Flores 1: gCpMP para o caso 4

Cluster	480 min.
	Demanda (min.)
1	466
2	479
3	467
4	479
5	477
6	403
7	373
Tempo de Resolução (s.)	92.72

Fonte: O autor (2020).

Para o cenário estabelecido e as situações propostas, após o agrupamento das arestas, a conectividade de cada *cluster* foi verificada por meio do Algoritmo de Busca em Profundidade Modificado. As Tabelas 29, 30, 31 e 33 abaixo exibem o tempo de execução do Algoritmo de Busca em Profundidade Modificado para os *clusters* obtidos por meio da aplicação do *gCpMP*.

Tabela 29 – Flores 1: DFS modificado para o caso 1

Cluster	480 min.
	Tempo de Resolução (s.)
1	0.96
2	0.768
3	0.643
4	0.713

Fonte: O autor (2020).

Tabela 30 – Flores 1: DFS modificado para o caso 2

Cluster	480 min.
	Tempo de Resolução (s.)
1	0.756
2	0.783
3	0.713
4	0.617
5	0.749

Fonte: O autor (2020).

Tabela 31 – Flores 1: DFS modificado para o caso 3

Cluster	480 min.
	Tempo de Resolução (s.)
1	0.756
2	0.762
3	0.834
4	0.876
5	0.954
6	0.856
7	0.713

Fonte: O autor (2020).

Para a situação com vistoria de 1 minuto por relógio de medição e tempo de trabalho de 480 minutos, um *cluster* é desconexo, nesse caso, o *cluster* número 7. Esse *cluster* tem dois componentes um formado pelo conjunto de vértices {20, 53, 51, 44, 41, 40, 39, 38, 37, 46, 45, 43, 42, 74, 35, 34, 32, 23, 22, 21, 50, 52, 56} e outro pelo conjunto de vértice {50, 49}, ao adicionar a aresta entre os vértices {48, 49} o subgrafo do *cluster* torna-se conexo, dessa forma, na demanda são adicionados 2 minutos, conforme pode-se ver na Tabela 32 abaixo.

Tabela 32 – Flores 1: cluster 7 para o caso 3

Cluster	480 min.		
	Demanda (min.)	Demanda (min.)	Tempo de Resolução (s.)
7	396	398	1.112

Fonte: O autor (2020).

Tabela 33 – Flores 1: DFS modificado para o caso 4

Cluster	480 min.
	Tempo de Resolução (s.)
1	0.691
2	0.657
3	0.981
4	0.723
5	0.70
6	0.756
7	0.645

Fonte: O autor (2020).

Para a situação com vistoria de 1 minuto acrescido de 1 minuto por aresta e tempo de trabalho de 480 minutos, o *cluster* 2 tem dois conjuntos de vértices um formado pelos vértices {7, 16, 80, 83, 82, 15, 10, 14, 13, 12, 11, 84, 85, 86, 89, 81, 78, 17, 55, 79} e outro pelos vértices {87, 88}. Mediante a aplicação do Algoritmo de Dijkstra Modificado, ao adicionar a aresta entre os vértices {85, 87} o *cluster* 2 torna-se conexo, a demanda de atendimento do *cluster* aumenta em 2 minutos, ver Tabela 34.

Tabela 34 – Flores 1: cluster 2 para o caso 4

Cluster	480 min.		
	Demanda (min.)	Demanda (min.)	Tempo de Resolução (s.)
2	479	481	1.504

Fonte: O autor (2020).

Após testada a conectividade para cada *cluster* presente em cada cenário e situação, as arestas necessárias para tornar os subgrafos conexos em cada *cluster* foram adicionadas. As Tabelas 35, 36, 37 e 38 mostram o Problema do Carteiro Chinês Direcionado aplicado a cada *cluster*.

Tabela 35 – Flores 1: PCC para o caso 1

Cluster	480 min.
	Distância (m.)
1	7322
2	6412
3	8250
4	9158

Fonte: O autor (2020).

Tabela 36 – Flores 1: PCC para o caso 2

Cluster	480 min.
	Distância (m.)
1	6032
2	6764
3	5738
4	5814
5	6794

Fonte: O autor (2020).

Tabela 37 – Flores 1: PCC para o caso 3

Cluster	480 min.
	Distância (m.)
1	4750
2	3812
3	3540
4	5114
5	4924
6	3950
7	5176

Fonte: O autor (2020).

Tabela 38 – Flores 1: PCC para o caso 4

Cluster	480 min.
	Distância (m.)
1	4524
2	4282
3	3778
4	5050
5	5366
6	4350
7	3950

Fonte: O autor (2020).

6.3 FLORES 2

Ainda utilizando-se os dados e o grafo associado a cidade de Flores. Para o estudo foi criado um cenário em que os leituristas tem carga de trabalho de 8 horas ou 480 minutos. Para esse cenário variou-se o tempo de leitura de cada ponto de consumo em 1 minuto, 0.7 segundos e 0.5 segundos, em uma quarta situação considerou-se o tempo de 1 minuto para leitura e para cada aresta foi adicionado o valor extra de 1 minuto para cobrir alguma possível dificuldade de passagem.

Para cada segmento de rua ou aresta (i, j) foi computado o número de residências $QtPontos$ e aplicou-se a Equação 4.2, para estabelecer o valor de v_{ij} em cada aresta.

Para cada aresta foram identificados os pontos médios, de cada ponto médio foram retiradas a latitude e a longitude por meio do *Google Maps*. As coordenadas geográficas foram tratadas por meio da biblioteca *scipy.spatial* do *Python*, essa ferramenta aplica a Fórmula de Haversine aos dados e permite encontrar a matriz quadrada das distâncias euclidianas para todas as coordenadas.

Combinou-se o cenário proposto e as situações estabelecidas e compõe-se os quatro casos, vistos na Tabela 24.

A Tabela 39 abaixo mostra o cenário proposto para o tempo de trabalho de 480 minutos, para o cenário quatro situações são idealizadas, onde os tempos de vistoria dos relógios que medem o consumo podem variar. Para o cenário e a situação a quantidade de *clusters* ou leituristas podem variar, para cada caso, a quantidade de agrupamentos é definido pela soma do valor de w_{ij} de todas as arestas do grafo dividida pela capacidade de atendimento de cada leiturista, como a divisão gera um número fracionário, arredonda-se esse número.

Tabela 39 – Flores 2: cenário e situações		
Tempo por vistoria	480 min.	
	$\sum w_{ij}$	p
0.5 s.	1705	4
0.7 s.	2219	5
1 min.	2935	7
1 min. + 1 min. para a travessamento de cada via	3095	7

Fonte: O autor (2020).

Após a obtenção da matriz de distâncias, do tempo de atravessamento de cada vértice, w_{ij} , do custo de atravessamento c_{ij} , do número de medianas ou leituristas a serem alocados, da definição das capacidades de cada leiturista, aplica-se o *gCpMP*.

As Tabelas 40, 41, 42 e 43 tratam respectivamente das situações em que o tempo de vistoria de cada relógio medidor é de 0.5 segundos, 0.7 segundos, 1 minuto e 1 minutos, com a adição de mais 1 minuto por aresta. Para o cenário com carga de trabalho de 480 minutos em cada situação observa-se nas Tabelas 40, 41, 42 e 43 o número de *clusters* ou agrupamentos, as demandas de atendimento de todas as arestas ou segmentos de rua presentes em um *cluster* e a distância a ser percorrida para atender todas as ruas presentes nesse agrupamento. As Tabelas

40, 41, 42 e 43 também expõem o tempo de processamento consumido para a obtenção dos *clusters* ou agrupamentos por meio do *gCpMP* modelado e resolvido mediante a biblioteca PuLP do *Python*.

Tabela 40 – Flores 2: gCpMP para o caso 1	
	480 min.
Cluster	Demanda (min.)
1	480
2	479
3	331
4	415
Tempo de Resolução (s.)	58.28
Fonte: O autor (2020).	

Tabela 41 – Flores 2: gCpMP para o caso 2	
	480 min.
Cluster	Demanda (min.)
1	440
2	444
3	472
4	398
5	465
Tempo de Resolução (s.)	40.43
Fonte: O autor (2020).	

Tabela 42 – Flores 2: gCpMP para o caso 3	
	480 min.
Cluster	Demanda (min.)
1	477
2	480
3	440
4	479
5	362
6	417
7	440
Tempo de Resolução (s.)	53.62
Fonte: O autor (2020).	

Tabela 43 – Flores 2: gCpMP para o caso 4

Cluster	480 min.
	Demanda (min.)
1	445
2	398
3	475
4	422
5	373
6	342
7	480
Tempo de Resolução (s.)	10.57

Fonte: O autor (2020).

Para o cenário estabelecido e as situações propostas, após o agrupamento das arestas, a conectividade de cada *cluster* foi verificada por meio do Algoritmo de Busca em Profundidade Modificado. As Tabelas 44, 46, 47 e 48 abaixo exibem o tempo de execução do Algoritmo de Busca em Profundidade Modificado para os *clusters* obtidos por meio da aplicação do *gCpMP*.

Tabela 44 – Flores 2: DFS modificado para o caso 1

Cluster	480 min.
	Tempo de Resolução (s.)
1	0.89
2	0.75
3	0.729
4	0.7

Fonte: O autor (2020).

Para a situação com vistoria de 0.5 segundos e tempo de trabalho de 480 minutos, um *cluster* é desconexo, nesse caso, o *cluster* número 3, observe a Tabela 45. Esse *cluster* tem dois componentes um formado pelo conjunto de vértices {15, 82, 95, 94, 93, 92, 91, 90, 88, 87, 85, 86, 89, 84, 83, 98, 97, 96, 100, 101, 105, 103, 99} e outro pelo conjunto de vértice {13, 80}.

Tabela 45 – Flores 2: Cluster 2 para o caso 4

Cluster	480 min.		Tempo de Resolução (s.)
	Demanda (min.)	Demanda (min.)	
3	331	333	1.112

Fonte: O autor (2020).

Ao adicionar a aresta entre os vértices $\{(15,14), (14,13)\}$ o subgrafo do *cluster* torna-se conexo, dessa forma, na demanda são adicionados 2 minutos, conforme pode-se ver na Tabela 45.

Tabela 46 – Flores 2: DFS modificado para o caso 2

Cluster	480 min.
	Tempo de Resolução (s.)
1	0.756
2	0.783
3	0.713
4	0.617
5	0.749

Fonte: O autor (2020).

Tabela 47 – Flores 2: DFS modificado para o caso 3

Cluster	480 min.
	Tempo de Resolução (s.)
1	0.756
2	0.762
3	0.834
4	0.876
5	0.954
6	0.856
7	0.713

Fonte: O autor (2020).

Tabela 48 – Flores 2: DFS modificado para o caso 4

Cluster	480 min.
	Tempo de Resolução (s.)
1	0.691
2	0.657
3	0.981
4	0.723
5	0.70
6	0.756
7	0.645

Fonte: O autor (2020).

Para a situação com vistoria de 1 minuto acrescido de 1 minuto por aresta e tempo de trabalho de 480 minutos, o *cluster* 5 tem dois conjuntos de vértices um formado pelos vértices $\{20, 53, 51, 44, 41, 40, 39, 38, 37, 46, 45, 43, 42, 35, 34, 32, 23, 22, 21, 50, 52, 56\}$ e outro pelos vértices $\{48, 49\}$. Mediante a aplicação do Algoritmo de Dijkstra Modificado, ao adicionar a aresta entre os vértices $\{50, 49\}$ o *cluster* 5 torna-se conexo, a demanda de atendimento do cluster aumenta em 1 minutos, ver Tabela 49.

Tabela 49 – Flores 2: cluster 2 para o caso 4

Cluster	480 min.		
	Demanda (min.)	Demanda (min.)	Tempo de Resolução (s.)
5	331	332	1.112

Fonte: O autor (2020).

Após testada a conectividade para cada *cluster* presente em cada cenário e situação, as arestas necessárias para tornar os subgrafos conexos em cada *cluster* foram adicionadas. As Tabelas 50, 51, 52 e 53 mostram o Problema do Carteiro Chinês Misto aplicado a cada *cluster*.

Tabela 50 – Flores 2: PCC para o caso 1

Cluster	480 min.
	Distância (m.)
1	9458
2	7316
3	5902
4	7692

Fonte: O autor (2020).

Tabela 51 – Flores 2: PCC para o caso 2

Cluster	480 min.
	Distância (m.)
1	5826
2	6481
3	5716
4	5083
5	6663

Fonte: O autor (2020).

Tabela 52 – Flores 2: PCC para o caso 3

Cluster	480 min.
	Distância (m.)
1	4767
2	311
3	5154
4	3540
5	4324
6	3801
7	5155

Fonte: O autor (2020).

Tabela 53 – Flores 2: PCC para o caso 4

Cluster	480 min.
	Distância (m.)
1	4413
2	5188
3	3828
4	4956
5	3801
6	4432
7	3540

Fonte: O autor (2020).

6.4 DISCUSSÕES

Este capítulo trata da discussão dos resultados e da revisão de literatura realizada para dar suporte ao trabalho.

6.4.1 Discussão dos resultados

O PRL é um problema de otimização combinatória que consiste em encontrar percursos com os menores custos de atravessamento associados para que leituristas capacitados realizem o atravessamento e a vistoria de pontos de consumo de uma rede urbana. No PRL o leiturista inicia a rota de determinado ponto de origem, percorre todos os caminhos da área previamente estabelecida, ao menos uma vez, realizando a leitura do consumo, e posteriormente, retorna ao ponto de origem. Nesse sentido, a abordagem *cluster first – route second* proposta procura atender os requisitos de um PRL, a medida que divide o espaço em *clusters* atribuídos a leituristas capacitados e identifica o circuito Euleriano de cada agrupamento, por meio do Problema do Carteiro Chinês.

Usberti et al. (2008) demonstra que o PRL pode ser tratado como um PCCC. No modelo matemático proposto por Golden e Wong (1981) para a resolução do PCCC, a função objetivo do modelo tem como função objetivo minimizar a distância percorrida por k veículos dado uma capacidade Q estabelecida para cada veículo, já na abordagem proposta o agrupamento de arestas ocorre por meio do Problema das p - Medianas que tem como função objetivo minimizar o custo de deslocamento entre os vértices clientes e o local candidato a alocação da instalação, priorizando a formação de *clusters* que possuem arestas o mais próximo possível uma das outras dado uma capacidade Q estabelecida para cada agrupamento. Para a abordagem proposta, sabe-se que o tempo de atravessamento, w_{ij} , é o custo de percorrer e realizar a vistoria dos pontos de

consumo da aresta localizada entre os vértices (i, j) de um lado ou em ambos os lados. Ao clusterizar as arestas e consequentemente os valores de w_{ij} associados a elas, o subgrafo de cada *cluster* poderá ser tratado posteriormente por meio do Problema do Carteiro Chinês.

O PCCC proposto por Golden e Wong (1981) considera a conectividade em sua estrutura axiomática e o *cluster* estabelecido no problema não tem a capacidade do veículo excedida, já na abordagem proposta neste trabalho, a fase de agrupamento, em alguns casos, pode gerar *clusters* desconexos, e para tornar esses *clusters* conexos adiciona-se arestas que podem, eventualmente, exceder a capacidade dos leituristas. No entanto, dado a função objetivo do Problema das p - Medianas aplicado, os conjuntos de vértices desconexos de um *cluster* estão relativamente próximos, dessa forma ocorre apenas a adição do tempo de passagem, l_{ij} , do caminho mínimo que torna o subgrafo não direcionado do *cluster* em um subgrafo conexo.

No modelo de Golden e Wong (1981) existe a eleição de um vértice aleatório como depósito ou depot dentre todos os vértices existentes, no entanto, no modelo proposto esse vértice atribuído ao depósito não existe durante a otimização do roteamento, uma vez que o vértice depósito do modelo de Golden e Wong (1981) é escolhido arbitrariamente e pode não corresponder a localização de um depósito real, logo o modelo proposto foca na otimização do roteamento dentro de uma determinada área, não considerando os custos de deslocamento da garagem para a área estudada, ou da área estudada para outro ponto.

A modelagem proposta também possibilita a aplicação do Problema do Carteiro Chinês Misto em cada agrupamento durante o roteamento após a fase de clusterização. Logo a abordagem pode tratar grafos mistos, o modelo de Golden e Wong (1981) por sua vez trata apenas grafos não direcionados.

O presente trabalho aplica a abordagem proposta a duas áreas urbanas, sendo elas, um bairro de pequeno porte na cidade de Recife e uma cidade do interior de Pernambuco.

A construção de cenários e situações para cada espaço urbano possibilitou a análise da sensibilidade da abordagem. Percebeu-se que o tempo de processamento para o solucionamento do Problema das p -Medianas aumentou com o número de medianas ou leituristas e instâncias (vértices e arcos). Apesar do problema das p -Medianas ser um problema *NP - difícil*, ele oferece uma solução ótima para pequenas instâncias em um espaço de tempo relativamente curto, conforme pode-se ver a aplicação da abordagem na cidade de Flores, o que torna a procedimento proposto adequada para cidades, bairros ou malhas urbanas de pequeno porte. Ao agrupar ruas onde pode-se realizar o movimento de ida e volta a abordagem reduz o espaço de soluções viáveis.

O Algoritmo Busca em Profundidade que testa a conectividade de cada *clusters* para cada cenário e situação no bairro de Engenho do Meio e na cidade de Flores ofereceu um tempo de processamento relativamente curto, variando em cerca de 1 segundo.

Ao testar a conectividade para ambos os territórios, a saber, Engenho do Meio e Flores (1 e 2), o subgafo de alguns *clusters* associados aos territórios mostraram-se desconexos. Ao aplicar Algoritmo de Dijkstra Modificado e adicionar as arestas que tornam os *clusters* conexos, alguns agrupamentos excederam a capacidade de atendimento dos leituristas. O Algoritmo de Dijkstra Modificado também consumiu pouco tempo computacional durante a execução.

A aplicação do Algoritmo Busca em Profundidade e do Algoritmo de Dijkstra Modificado foi mostrado para cada *cluster* individualmente com o propósito de verificar o tempo que cada algoritmo exige para execução.

No Engenho do Meio para os 8 casos simulados 70 *clusters* foram obtidos, destes apenas 6 foram identificados como desconexos, dos 6 o total de 2 excederam a capacidade do leiturista ao tornarem-se conexos. Um deles excedeu a capacidade do leiturista em 2 minutos e outro em 9 minutos.

Para a cidade de Flores 1, 23 *clusters* foram observados, deste total 2 *clusters* foram identificados como desconexos e 1 deles excedeu em 1 minuto a capacidade do leiturista. Na cidade de Flores 2, 23 *clusters* também foram observados, deste total 2 *clusters* foram identificados como desconexos desses nenhum excedeu a capacidade ao tornarem-se conexos.

Dos 8 casos simulados no Engenho do Meio, o caso número 1 teve o *cluster* 1 com a capacidade excedida em 2 minutos, assim como o caso número 2, que teve o *cluster* 3 com a capacidade excedida em 9 minutos. Quando trata-se da cidade de Flores 1, dos 4 casos, o caso número 4 teve o *cluster* 2 com a capacidade do leiturista excedida em 1 minuto.

Na aplicação do Problema do Carteiro Chinês um arquivo em lp foi escrito em Python para cada *cluster* modelando o problema, o arquivo em lp foi introduzido como entrada no software SCIP. O tempo para escrever a modelagem matemática para cada *cluster* em lp foi inferior a 0.01 segundos e a resolução do PCC para cada agrupamento durou em média 1 segundo.

Como resultado do estudo do Problema de Roteamento de Leituristas e aplicação do Problema do Carteiro Chinês, o autor da dissertação realizou a publicação de Lima e Lins (2020), o trabalho tem como objetivo encontrar a menor distância associada a rota de leitura que percorra todas as ruas de uma cidade de pequeno porte ao menos uma vez, considerando-se ambos os lados da via, logo a presente dissertação aprofundou o estudo do Problema do Carteiro Chinês do trabalho de Lima e Lins (2020) ao tratar o Problema do Carteiro Chinês

Capacitado associado ao Problema do Roteamento de Leituristas. No Problema do Carteiro Chinês Capacitado o roteamento é realizado por mais de um veículo ou entidade, o veículo que realiza o roteamento possui uma capacidade e as arestas ou arcos por onde o veículo passa são dotadas de demanda.

6.4.2 Discussão da revisão sistemática de literatura

Após a realização da Revisão Sistemática de Literatura observou-se que dois trabalhos possuem similaridades e diferenças com a abordagem proposta para a resolução do PRL, PCCC e PRAC, os dois trabalhos são: Smiderle, Steiner e Carnieri (2003) e Vecchi et al. (2016). Ambos os trabalhos propõem uma abordagem *cluster first – route second*; Smiderle, Steiner e Carnieri (2003) solucionam o PCCC e o PRL, por outro lado, Vecchi et al. (2016) tratam do PRAC. Na clusterização tanto Smiderle, Steiner e Carnieri (2003), quanto Vecchi et al. (2016) utilizam o Capacitated p -Median Problem no agrupamento, no presente trabalho propõem-se a aplicação do generic Capacitated p -Median Problem. O generic Capacitated p -Median Problem considera a demanda da aresta onde será localizada a mediana ou a instalação e oferece maior acuracidade na solução, algo que não ocorre no Capacitated p -Median Problem.

Nos trabalhos de Smiderle, Steiner e Carnieri (2003) e Vecchi et al. (2016) não ocorre a verificação da conectividade de cada *cluster*, ou a proposição de uma estratégia para tornar um agrupamento conexo, algo que ocorre no presente trabalho.

7 CONCLUSÃO

Aplicações de técnicas de Pesquisa Operacional, como a Programação Linear, podem trazer importantes benefícios para a resolução de problemas em empresas e organizações. Neste trabalho é proposto a utilização de métodos oriundos da Pesquisa Operacional, Teoria dos Grafos e Otimização Combinatória para aprimorar a logística que envolve o percurso dos leituristas de uma companhia de energia elétrica.

O objetivo desta pesquisa consistiu em propor uma abordagem *cluster first – route second* para o Problema de Roteamento de Leituristas e consequentemente o Problema do Carteiro Chinês Capacitado. A abordagem proposta utiliza uma técnica exata consolidada na literatura para o agrupamento, um algoritmo robusto para verificar a conectividade e outro para verificar o conjunto de arestas que tornam o subgrafo desconexo de um *cluster* em um subgrafo conexo.

Para a aplicação da abordagem proposta o espaço urbano é tratado como um grafo, na fase de clusterização as arestas do grafo são agrupadas em subgrafos por meio do Problema das p -Medianas. A fase de roteamento ocorre depois da verificação da conectividade do subgrafo de cada *cluster* mediante a aplicação de um Algoritmo de Busca em Profundidade, caso um subgrafo não seja conexo, acrescenta-se as arestas que tornam o subgrafo conexo, mediante a aplicação do Algoritmo de Dijkstra Modificado. Para o roteamento, o circuito Euleriano é encontrado por meio da aplicação do Problema do Carteiro Chinês.

Aplicou-se a abordagem proposta a dois casos distintos com dados reais e fez-se inferências. Apesar do modelo oferecer *clusters* desconexos que excedem a capacidade estabelecida para o leiturista ao tornarem-se conexos, o tempo acrescido para tornar um *cluster* conexo corresponde ao tempo de passagem de determinado conjunto de arestas, esse conjunto de arestas adicionado ao *cluster* o tornam conexo. Logo, a abordagem proposta tanto pode ser usada para o planejamento de rotas de leitura, quanto oferece uma alternativa ainda não estudada na literatura para o roteamento de leituristas e consequentemente para resolução do Problema do Carteiro Chinês Capacitado.

O Problema do Carteiro Chinês Capacitado deriva do Problema do Caixeiro Viajante ou Travelling Salesman Problem que é de difícil resolução para instâncias de pequeno porte. Os casos reais em que a abordagem *cluster first – route second* proposta foi modelada e executada comprovam a aplicabilidade do procedimento. Apesar do Problema das p - Medianas utilizado para o agrupamento das arestas ser NP - *difícil*, ele oferece uma solução ótima para problemas com instâncias pequenas e de médio porte como acontece com o grafo associado com a cidade

de Flores em um tempo relativamente curto. Segundo Calvo et. al. (2016) a maioria dos municípios brasileiros ou 76,9% deles são considerados de pequeno porte, dessa forma o roteamento de leituristas por meio da abordagem proposta pode ser aplicado sem grandes custos computacionais para o planejamento de rotas de leitura.

Para a aplicação da abordagem com mais acurácia, o tempo médio que um leiturista leva para realizar a vistoria de relógio de consumo poderia ser determinado por meio de cronoanálise, este trabalho estabeleceu tempos arbitrários para o tempo médio de leitura e carga de trabalho. O tempo médio de leitura impacta diretamente no peso de cada aresta e na definição da quantidade de leituristas.

Como dificuldades para a aplicação da abordagem tem-se a não exatidão do tempo médio de vistoria de um relógio de consumo. Utilizou-se o *Google Maps* para contar o número de pontos de leitura em cada segmento de rua estabelecido como aresta, essa contagem requer esforço humano e está sujeita a erros. As imagens das ruas oferecidas pelo *Google Maps* para contagem possuem a possibilidade de estarem desatualizadas e não oferecerem o número real de relógios de leitura, ou estabelecimentos e construções que representam pontos de consumo. Os tempos de passagem l_{ij} em minutos e os custos de atravessamento c_{ij} em metros, são tempos e custos médios oferecidos pelo *Google Maps*.

Para trabalhos futuros sugere-se a aplicação da abordagem proposta a outros problemas que podem ser modelados por meio do Problema do Carteiro Chinês Capacitado como por exemplo, o roteamento de veículos coletores de lixo urbano. Após a aplicação do Problema do Carteiro Chinês para encontrar o caminho mínimo presente em um agrupamento, algoritmos para o sequenciamento podem ser utilizados para a definição de rotas, o algoritmo de Fleury é utilizado no sequenciamento e identificação de um circuito euleriano em grafos não direcionados conexos, enquanto o algoritmo de Heierholzer é utilizado no sequenciamento de um circuito euleriano em grafos direcionados conexos. Neste trabalho, na fase de roteamento utilizou-se o Problema do Carteiro Chinês para encontrar o caminho mínimo que saindo de um vértice de origem percorra todas as arestas de um grafo e retorne ao vértice de origem, em trabalhos futuros pode ser que não seja necessário retornar ao vértice de origem depois de passar por todas as arestas de um grafo, dessa forma um algoritmo de caminho mínimo pode ser utilizado para o roteamento.

REFERÊNCIAS

- ABREU, N. M. M. A. Teoria da complexidade computacional. **Revista militar de ciência e tecnologia**, Rio de Janeiro, v. 4, n. 1, p. 90 - 95, 1987.
- AHR D.; REINELT G. New heuristics and lower bounds for the min-max k -chinese postman problem. In: MÖHRING R.; RAMAN R., ed. **Algorithms – ESA 2002**. Berlin: Springer, 2002. cap 6, p. 64 - 74.
- AHUJA, R. K.; MAGNANT, T. L.; ORLIN; J. B. **Network flows: theory, algorithms, and applications**. New Jersey: Prentice-Hall, 1993.
- ARAKAKI, R.K.; USBERTI, F.L. Hybrid genetic algorithm for the open capacitated arc routing problem. **Computers and operations research**, Amsterdam, v. 90, p. 221-231, 2018.
- ARAKAKI, R.K.; USBERTI, F.L. An efficiency-based path-scanning heuristic for the capacitated arc routing problem. **Computers and operations research**, Amsterdam, v. 103, p. 288-295, 2019.
- ASCENCIO, A. F. G.; ARAÚJO, G. S. **Estruturas de dados: algoritmos, análise da complexidade e implementações em Java e C/C++**. São Paulo: Pearson Prentice Hall, 2010.
- ATKINS J. E.; DIERCKMAN J. S.; O'BRYANT K. A real snow job. **The undergraduate mathematics and its applications**, Lexington, v. 11, p. 231-239, 1990.
- BAKER, E. K. An exact algorithm for the time constrained travelling salesman problem. **Operations research**, Baltimore, v. 31, n. 5, p. 938-945, 1983.
- BALAKRISHNAN, R.; RANGANATHAN, K. **A textbook of graph theory**. New York: Springer-Verlag, 2012.
- BELENGUER, J. M.; BENAVENT, E. A cutting plane algorithm for the capacitated arc routing problem. **Computers and operations research**, Amsterdam, v. 30, n. 5, p. 705–728, 2003.
- BELTRAMI, E. J.; BODIN, L. D. Networks and vehicle routing for municipal waste collection. **Networks**, Hoboken, v. 4, n.1, p.65-94, 1974.
- BENAVENT, E. et al. The capacitated arc routing problem: a heuristic algorithm. **Qüestiió**, Grenoble, v. 14, n. 1, p. 107 - 122, 1990.
- BENJAMIN, A.; CHARTRAND, G.; ZHANG, P. **The fascinating world of graph theory**. Princeton: Princeton University Press, 2015.
- BIGGS, N. L.; LLOYD, E. K.; WILSON, R. J. **Graph theory: 1736-1936**. Oxford: Clarendon Press, 1986.
- BLANCHARD, S. J., ALOISE, D., DESARBO, W. S. The heterogeneous p -median problem for categorization based clustering. **Psychometrika**, Amsterdam, v. 77, n.4, p. 741–762, 2012.

BLUM, L. et al. Complexity and real computation: a manifesto. **International journal of bifurcation and chaos**, Hackensack, v. 6, n. 1, p. 3-26, 1996.

BODIN, L. D. et al. Routing and scheduling of vehicles and crews: the state of the art. **Computers e operations research**, Amsterdam v. 10, p. 63 - 212, 1983.

BODIN, L. D.; KURSH, S. J. A computer-assisted system for the routing and scheduling of street sweepers. **Operations research**, Baltimore, v. 26. n. 4, p. 525–537, 1978.

BODIN, L. D.; KURSH, S. J. A detailed description of a computer system for the routing and scheduling of street sweepers. **Operations research**, Baltimore, v. 6, n. 4, p. 181–198, 1979.

BODIN, L. D.; LEVY, L. The arc partitioning problem. **European journal of operational research**, Amsterdam, v. 53, n. 3, p. 393–401, 1991.

BOVET, D. P.; CRESCENZI, P. **Introduction to the theory of complexity**. New York: Prentice Hall, 1993.

BRASIL. Resolução nº 3, de 26 de agosto de 2019. **Diário Oficial da União**, Brasília, DF, 28 agosto 2019. Seção 1, p. 374.

BRASSARD, G.; BRATLEY, G. P. **Algorithmics, theory e practice**. Upper Saddle River: Prentice-Hall, 1988.

CALVO, M. C. M. et al. Estratificação de municípios brasileiros para avaliação de desempenho em saúde. *Revista epidemiologia e serviços de saúde*, Brasília, v.25, n. 4, 767-776, 2016.

CAMPBELL, J. F. et al. Drone arc routing problems. **Networks**, Hoboken, v. 72, n. 4, p. 543-559, 2018.

CHURCH, R. L. BEAMR: An exact and approximate model for the p-median problem. **Computers and operations research**, Amsterdam, v.35, n 2, p. 417–426, 2008.

CHRISTOFIDES, N. **Graph theory: an algorithmic approach**. London: Academic Press, 1975.

CHRISTOFIDES, N. **Worst-case analysis of a new heuristic for the travelling salesman problem**. Fort Belvoir: Defense Technical Information Center, 1976

COMERT, S. E. et al. A cluster first-route second approach for a capacitated vehicle routing problem: a case study. **International journal of procurement management**, v. 11, n. 4, p. 399-419, 2018.

COOK, S.A. An overview of computational complexity. **Communications of the association for computing machiner**, v. 26, n. 6, p. 401-407, 1983.

CORBERÁN, Á. et al. Arc routing problems: a review of the past, present, and future. **Networks**, Hoboken, n. 1, p. 1 – 27, 2020.

CORBERÁN, A.; PRINS, C. Recent results on arc routing problems: an annotated bibliography. **Networks**, Hoboken, n. 56, p. 50–69, 2010.

CORMEN, T. H. et al. **Introduction to algorithms**. Cambridge: MIT Press, 2009.

CORMEN, T. H. et al. **Algoritmos: teoria e prática**. Rio de Janeiro: Elsevier, 2012.

CRUZ, Y. X. D.; CHIRVA, J. A. P.; SANTANA, E. R. L. A mixed integer optimization model to design a selective collection routing problem for domestic solid waste. *In*: WORKSHOP ON ENGINEERING APPLICATIONS, Bogota, 2015. **Anais [...]**. Bogota: WEA, 2015..

CUNHA, C. B. Aspectos práticos da aplicação de modelos de roteirização de veículos a problemas reais. **Associação nacional de pesquisa e ensino em transportes**, v. 8, n. 2, p. 51-74, 2000.

CUNHA, V. A. M. et al. Dimensionamento de mão de obra e roteamento através de um algoritmo VND: estudo de caso em uma empresa de medição de consumo de energia. *In*: SIMPÓSIO BRASILEIRO DE PESQUISA OPERACIONAL, 49., Blumenau, 2017. **Anais [...]**. Blumenau: SOBRAPO, 2017.

DASKIN, M. **Network and discrete location: models, algorithms, and applications**. New York: Wiley Interscience, 1995.

DIESTEL, R. **Graph Theory**. New York: Springer, 2000.

DIJKSTRA, W. A note on two problems in connection with graphs. **Numerische mathematik**, v. 1, p. 269-271, 1959.

DREZNER, Z. (org.). **Facility location: a survey of applications and methods**. New York: Springer, 1995.

DROR, M. (org.). **Arc routing: theory, solutions and applications**. New York: Springer, 2012.

DRUMMOND, J. P.; SILVA, E. E.; COUTINHO, M. **Medicina baseada em evidências**. São Paulo: Atheneu, 2004.

EDMONDS, J.; JOHNSON, E. Matching, euler tours and the chinese postman problem. **Mathematical programming**, v. 5, p. 88-124, 1973.

EGLESE, R.; GOLDEN, B. E.; WASIL, E. Chapter 13: route optimization for meter reading and salt spreading. *In*: Corberán, Á.; Laporte, G. (org.). **Arc routing: problems, methods, and applications**. Philadelphia: SIAM, 2015. cap. 13. p. 303 – 320.

EISELT, H. A.; GENDREAU, M.; LAPORTE, G. Arc routing problems, part I: the chinese postman problem. **Operations research**, Baltimore, v. 43, n. 2, p. 231-242, 1995.

EULER, L. Solutio problematis ad geometriam situs pertinentes. **Commentarii academiae scientiarum imperialis petropolitanae**, v. 8, p. 128–140, 1736.

EUROPEAN COMMISSION ENERGY. EU 28 smart metering benchmark revised final report, 2019. Disponível em: https://www.vert.lt/SiteAssets/teises_aktai/EU28%20Smart%20Metering%20Benchmark%20Revised%20Final%20Report.pdf. Acesso em: 20 set. 2020.

EVEN, S. **Graph algorithms**. 2 ed. Cambridge: Cambridge University Press, 2011.

FERNANDEZ-VIAGAS, V.; MOLINA-PARIENTE J. M.; FRAMINAN, J. M. Generalised accelerations for insertion-based heuristics in permutation flowshop scheduling. **European journal of operational research**, Amsterdam, v. 282, n. 3, p. 858-872, 2020.

FERREIRA, J. L.; MATOS, E. L. M. Mapeando o conhecimento científico em teses e dissertações na pedagogia hospitalar. In: ENCONTRO NACIONAL DE ATENDIMENTO HOSPITALAR, 7., Natal, 2012. **Anais [...]**. Natal: ENAH, 2012.

FOULDS, L.; LONGO, H.; MARTINS, J. A compact transformation of arc routing problems into node routing problems. **Annals of operations research**, v. 226, n. 1, p. 177-200, 2015.

FOURNIER, J. C. **Graph theory and applications with exercises and problems**. New York: John Wiley & Sons, 2009.

FOURNIER, G.; SCARSINI, M. Location games on networks: existence and efficiency of equilibria. **Mathematics of operations research**, v. 44, n. 1, p. 212-235, 2019.

GALVÃO, T. F. E.; PEREIRA, M. G. Revisões sistemáticas da literatura: passos para sua elaboração. **Epidemiologia e serviços de saúde**, v. 23, p. 183-184, 2014.

GAREY, M.R.; JOHNSON, D.S. **Computers and intractability: a guide to the theory of NP-completeness**. San Francisco: W.H. Freeman and Co., 1979.

GELDERS, L. F.; CATTRYSSE, D. G. Public waste collection: a case study. **Belgian journal of operations research, statistics and computer science**, v. 31, n. 1-2, p.7-18, 1991.

GHIANI, G. et al. Waste collection in southern Italy: solution of a real-life arc routing problem. **International transactions in operational research**, v. 12, n. 2, p.135–144, 2005.

GHIANI, G; IMPROTA, G. An Algorithm for the hierarchical Chinese postman problem. **Operations research letters**, v. 26, n. 1, p. 27-32, 2000.

GIL, A. C. **Métodos e técnicas de pesquisa social**. 6. ed. São Paulo: Atlas, 2008.

GODINHO FILHO, M.; JUNQUEIRA, R. Á. R. Problema do carteiro chinês: escolha de métodos de solução e análise de tempos computacionais. **Produção**, v. 16, n. 3, p. 538-551, 2006.

GOETSCHALCKX, M.; JACOBS-BLECHA, C. The vehicle routing problem with backhauls. **European journal of operational research**, Amsterdam, v. 42, n. 1, p. 39–51, 1989.

GOLDEN, B. L.; WONG, R. T. Capacitated arc routing problems. **Networks**, Hoboken, v. 11, n. 3, p. 305–315, 1981.

GONÇALO, C. S et al. Planejamento e execução de revisões sistemáticas da literatura. **Revista brasílica médica**, v. 49, p. 104-110, 2012.

GOODRICH, M. T.; TAMASSIA, R. **Estruturas de dados e algoritmos em Java**. Porto Alegre: Bookman, 2013.

GOOGLE MAPS. Google, 2010. Disponível em: <https://www.google.com.br/maps/>. Acesso em: 20 set. 2020.

GUREVICH, Y.; SHELAH, S. Expected computation time for hamiltonian path problem. **SIAM journal on computing**, v. 16, n. 3, p. 486–502, 1987.

SCIP.Optimization Suite, 2020. Disponível em : <https://soplex.zib.de/>. Acesso em: 20 set. 2020.

GROSS, J. L.; YELLEN, J.; ZHANG, P. **Handbook of graph theory**. New York: Chapman and Hall, 2013.

HAKIMI, S.L. Optimum location of switching centers and the absolute centers and the medians of a graph. **Operations research**, Baltimore v. 12, p. 450-459, 1965.

HARRIS, J.; HIRST, J. L; MOSSINGHOFF, M. **Combinatorics and graph theory**. New York: Springer, 2008.

HERBERT, F. Algorithms for eulerian trails and cycle decompositions, maze search algorithms. **Annals of discrete mathematics**, v. 50, p. 1-13, 1991.

HOLMBERG, K. Heuristics for the weighted k-rural postman problem with applications to urban snow removal. **Journal on vehicle routing algorithms**, v. 1, p. 105–119, 2018.

JUNGNICKEL, D. **Graphs, networks and algorithms**. New York: Springer, 2005.

KAUFFMAN, A. **Graphs, dynamic programming and finite games**. New York: Academic Press, 1967.

KAUFMAN, L.; ROUSSEEUW, P.J. **Finding groups in data: an introduction to cluster analysis**. New York: Wiley, 2005.

KARIV, O.; HAKIMI, S. L. An algorithmic approach to network location problems. II: the p-medians. **SIAM journal on applied mathematics**, v. 37, n.3, p. 539–560, 1979.

KLASTORIN, T. The p-median problem for cluster analysis: a comparative test using the mixture model approach. **Management science**, v. 31, n.1,p. 84–95, 1985.

KONOWALENKO, F. et al. Aplicação de um algoritmo genético para o problema do carteiro chinês em uma situação real de cobertura de arcos. **Revista ingeniería industrial**, v. 1 p. 27-36, 2012.

KOPEC, D. **Classic computer science problems in python**. New York: Manning publications, 2019.

KREYSZIG, E. **Advanced engineering mathematics**, 10 ed. New York: Wiley, 2011.

KWAN, M. K. Graphic programming using odd or even points. **Chinese mathematics**, v. 1, p. 273–277, 1962.

LARSON, R. C.; ODONI, A. R. **Urban operations research**. New Jersey: Prentice-Hall, 1981.

LATORA, V.; NICOSIA, V.; RUSSO, G. **Complex networks**: principles, methods and applications. Cambridge: Cambridge University Press, 2017.

LEWIS, H. R.; PAPADIMITRIOU, C. **Elementos de teoria da computação**. 2. ed. Porto Alegre: Bookman, 2000.

LIEBLING, T. M. Routing problems for street cleaning and snow removal. *In*: DEININGER, R. (org.). **Models for environmental pollution control**. Ann Arbor: Ann Arbor Science Publishers, 1973. p. 363-374.

LIMA, L. E. S.; LINS, S. L. S. Aplicação do problema do carteiro chinês não direcionado ao roteamento de leitorista em uma cidade do interior da Paraíba. *In*: ENCONTRO NACIONAL DE ENGENHARIA DE PRODUÇÃO, 40., Foz do Iguaçu, 2020. **Anais [...]**. Foz do Iguaçu: ABEPRO, 2020.

LORENA, L. A. N. et al. Integração de modelos de localização a sistemas de informações geográficas. **Gestão e produção**, v.8, n. 2, p.180-195, 2001.

MAPA, S. M. S.; LIMA, R. S.; MENDES, J. F. G. M. Localização de instalações com o auxílio de Sistema de informações Geográficas (SIG) e modelagem matemática. *In*: ENCONTRO NACIONAL DE ENGENHARIA DE PRODUÇÃO, 26., 2006, Fortaleza. **Anais[...]**. Fortaleza: ABEPRO, 2006.

MARINI M. C.; GELDERS, L. F. Repairable item inventory systems: a literature review. **Belgian journal of operations research, statistics and computer science**, v. 30, n. 4, p. 57-69, 1990.

MARKS H. D.; STRICKER R. Routing for public service vehicles. **Journal of the urban planning and development division**, v. 97, p. 165-178, 1971.

MERRIS, R. **Graph theory**. New Jersey: Wiley Interscience, 2001.

MIGUEL, P. A. C.; FLEURY, A.; MELLO, C. H. P.; et al. **Metodologia de pesquisa em engenharia de produção e gestão de operações**. 2ed. Rio de Janeiro: Elsevier, 2012.

- MINIEKA, E. The chinese postman problem for mixed networks. **Management science**, v. 25, n. 7, p. 643-648, 1979.
- MONROY, M. L.; AMAYA, C. A.; LANGEVIN, A. The periodic capacitated arc routing problem with irregular services. **Discrete applied mathematics**, v. 161, p.691-701, 2013.
- MOSS, C. R. A routing methodology for snow plows and cindering trucks. **Journal of mathematical analysis and applications**, v. 23, p.156-179, 1990.
- MULVEY, J.M.; BECK, M.P. Solving capacitated clustering problems. **European journal of operational research**, Amsterdam, v. 18, p. 339-348, 1984.
- NEGREIROS, M.; BATISTA, P.; RODRIGUES, J. A. Optimization Models for Capacitated Clustering Problems. *In: SIMPÓSIO BRASILEIRO DE PESQUISA OPERACIONAL*, 49., Blumenau, 2017. **Anais [...]**. Blumenau: SOBRAPO, 2017. p. 1-11.
- NEGREIROS, M. J.; PALHANO, A. W. The capacitated centred clustering problem. **Computers and Operations Research**, Amsterdam, v. 33, n. 6, p. 1639-1663, 2006.
- NETTO, P. O. B. **Grafos: teoria, modelos e algoritmos**. Massachusetts: Edgard Blücher, 2003. 314 p.
- ORLIS, C. et. al. Distribution with quality of service considerations: the capacitated routing problem with profits and service level requirements. **Omega**, v. 93, p. 1-13, 2020.
- OWEN, S. H.; DASKIN, M. S. Strategic facility location: a review. **European journal of operational research**, Amsterdam, v. 111, n. 3, p. 423-447, 1998.
- PAPADIMITRIOU, C.H. **Computational complexity**. New York: Wiley, 2003.
- PERRIER, N.; LANGEVIN, A.; AMAYA, C. A. Vehicle routing for urban snow plowing operations. **Transportation science**, v.42, n. 1, p. 44-56, 2008.
- PORTER, E. M. **Competição: estratégias competitivas essenciais**, 7. ed. Rio de Janeiro: Editora Campus, 1999.
- PRINS, C.; LABADI, N.; REGHIOUI, M. Tour splitting algorithms for vehicle routing problems. **International journal of production research**, v. 47, p. 507-535, 2009.
- PULP. PuLP 2.3.1. 2020. Disponível em: <https://pypi.org/project/PuLP/>. Acesso em: 20 set. 2020.
- QUADAR, N. et al. Smart water distribution system based on IoT networks, a critical review. *In: Zimmermann A.; Howlett R.; Jain L. (org.). Human centred intelligent systems: smart innovation, systems and Technologies*. Singapore: Springer, 2021. cap. 24. p. 293 – 304.
- RAHMAN, S. **Basic graph theory**. New York: Springer, 2017.
- ROMANOWSKI, J. P.; ENS, R. T. As pesquisas denominadas do tipo estado da arte em educação. **Revista diálogo educacional**, v. 6, p. 37-50, 2006.

SAMPAIO, R. F. E.; MANCINI, M. C. Estudos de revisão sistemática: um guia para síntese criteriosa da evidência científica. **Revista brasileira de fisioterapia**, v. 11, p. 83-89, 2007.

SAUNDERS, M.; LEWIS, P.; TORNHILL, A. **Research methods for business students**. 4. ed. London: Pearson Education, 2007.

SCHÖNING, U. Graph isomorphism is in the low hierarchy. **Journal of computer and system sciences**, v. 37, p. 312–323, 1988.

SCIPY. Spatial algorithms and data structures, 2020. Disponível em: <https://docs.scipy.org/doc/scipy/reference/spatial.html>. Acesso em: 20 set. 2020.

SEDGEWICK, R.; WAYNE, K. **Algorithms**. 4 ed. Boston: Addison-Wesley professional, 2011.

SEO, J. et al. Collision avoidance strategies for unmanned aerial vehicles in formation flight. **IEEE Transactions on Aerospace and Electronic Systems**, v.53, n.6, p. 2718–2734, 2017.

SHERAFAT, H. Sistema construtor de circuitos e sua aplicação na roteirização de coleta de lixo domiciliar. **Revista GEINTEC**, v. 3, n. 5, p. 329-347, 2013.

SINNOTT, R. W. Virtues of the Haversine. **Sky and telescope**, v. 68, n.2, p.159, 1984.

SKIENA, S. **The algorithm design manual**. London: Springer, 2008.

SLACK, N.; JONES, A. B.; JOHNSTON, R. **Administração da produção**. São Paulo: Atlas, 2016.

SMIDERLE, A.; STEINER, M. T. A.; CARNIERI, C. Problema de cobertura de arcos: um estudo de caso. In: ENCONTRO NACIONAL DE ENGENHARIA DE PRODUÇÃO, 13., Ouro Preto, 2003. **Anais [...]**. Ouro Preto: ABEPRO, 2003. p. 1-6.

STEFANELLO, F.; ARAÚJO, O. C. B.; MÜLLER, F. M. Matheuristics for the capacitated p-median problem. **International Transactions in Operational Research**, v. 22, n. 1, p. 149–167, 2014.

STERN, H. E.; DROR, M. Routing electric meter readers. **Computational and operational research**, v. 6, p. 209–223, 1979.

SZYDLOWSKI, R. **Advanced metering techniques**. Washington: Pacific Northwest Laboratory, 1993.

TRANSPORT CANADA. Transportation Development Agency. **Improving snow clearing effectiveness in Canadian municipalities**. Montreal, 1975.

UNIVERSITY OF PACIFIC. The Euler archive, 2019. Disponível em: <https://scholarlycommons.pacific.edu/euler-works/>. Acesso em: 20 set. 2020.

USBERTI, F. L.; FRANÇA, P. M. E.; FRANÇA, A. L. M. Roteamento de leituristas: um problema np - difícil. *In: SIMPÓSIO BRASILEIRO DE PESQUISA OPERACIONAL*, 40., João Pessoa, 2008. **Anais [...]**. João Pessoa: SOBRAPO, 2008, p. 1836 – 1847.

USBERTI, F. L.; FRANÇA, P. M. E.; FRANÇA, A. L. M. The open capacitated arc routing problem. **Computer and operations research**, Amsterdam, v. 38 p. 1543-1555, 2011a.

USBERTI, F. L.; FRANÇA, P. M. E.; FRANÇA, A. L. M. GRASP with evolutionary path-relinking for the capacitated arc routing problem. **Computer and operations research**, Amsterdam, v. 40, p. 1-12, 2011b.

USBERTI, F. L.; FRANÇA, P. M. E.; FRANÇA, A. L. M. On the complexity and heuristic methods for a new arc routing problem. *In: OPERATION RESEARCH PERIPATETIC POSTGRADUATE PROGRAMME*, 6., Cadiz, 2011c. **Anais [...]**. Cadiz: EURO, 2011.

USBERTI, F. L.; FRANÇA, P. M. E.; FRANÇA, A. L. M. Metaheurística grasp com reconexão por caminhos aplicada ao problema do roteamento de leituristas do consumo de energia elétrica. *In: CONGRESSO BRASILEIRO DE AUTOMÁTICA*, 19., Campina Grande, 2012a. **Anais [...]**. Campina Grande: SBA, 2012.

USBERTI, F. L.; FRANÇA, P. M. E.; FRANÇA, A. L. M. Branch-and-bound algorithm for an arc routing problem. *In: SIMPÓSIO BRASILEIRO DE PESQUISA OPERACIONAL*, 44., Rio de Janeiro, 2012b. **Anais [...]**. Rio de Janeiro: SOBRAPO, 2012b.

U.S. ENERGY INFORMATION ADMINISTRATION. How many smart meters are installed in the United States, and who has them?, Acesso em: 20 set. 2020. Disponível em: <https://www.eia.gov/tools/faqs/faq.php?id=108&t=3>. Acesso em: 20 set. 2020.

VALSAMIS, A. et. al. Employing traditional machine learning algorithms for big data streams analysis: The case of object trajectory prediction. **Journal of systems and software**, v. 127, p. 249–257, 2017.

VAN AARDENNE-EHRENFEST, T.; BRUIJN, N.G. Circuits and trees in oriented linear graphs. **Simon stevin**, v. 28, p. 293-217, 1951.

VAN BEVERN, R. et al. Chapter 2: The Complexity of Arc Routing Problems. Arc Routing. *In: CORBERÁN, À.; LAPORTE, G. (org.). Arc routing: problems, methods, and applications*. Philadelphia: SIAM, 2015. cap. 2. p. 19-52.

VECCHI, T. P. B. et al. A sequential approach for the optimization of truck routes for solid waste collection. **Process safety and environmental protection**, v. 102, 238–250, 2016.

WEST, D. B. **Introduction to graph theory**. New York: Pearson, 2018.

WILLIAMSON, D. P.; SHMOYS, D. **The design of approximation algorithms**. Cambridge: Cambridge University Press, 2011.

WILSON, R. **Introduction to graph theory**. London: Pearson, 2012.

WUNDERLICH, J. et al. Scheduling meter readers for southern california gas company. **Interfaces**, v. 22, p. 22-30, 1992.

XAVIER, A. E.; XAVIER, V. L. Solving the minimum sum - of- squares clustering problem by hiperbolic smoothing and partition into boundary and gravitational regions. **Pesquisa operacional para o desenvolvimento**, São Carlos, v. 44, p. 70 -77, 2011.

YAGHINI, M.; KARIMI, M.; RAHBAR, M. A hybrid metaheuristic approach for the capacitated p-median problem. **Applied soft computing**, Amsterdam, v. 13, n. 9, p. 3922–3930, 2013.

YAO, T. et al. Memetic algorithm with adaptive local search for capacitated arc routing problem. *In*: INSTITUTE OF ELECTRICAL AND ELECTRONICS ENGINEERS, 20., Yokohama, 2017. **Anais [...]**. Yokohama: IEEE, 2017.

ZIVIANI, N. **Projeto de algoritmos**. São Paulo: Cengage, 2019.

APÊNDICE A – DADOS ENGENHO DO MEIO

Aresta	Vértices		Coordenadas Geográficas	c_{ij}	l_{ij}	Casas		Condomínios				
	i	j				i	j	C. 1	C.2	C.3	C.4	C.5
1	1	2	-8.063590, -34.942005	61	1	6	7					
2	1	4	-8.062664, -34.941568	190	3	22	21					
3	2	5	-8.062772, -34.942171	210	3	21	17					
4	2	3	-8.063662, -34.942588	87	2	4	5					
5	3	7	-8.063405, -34.943021	50	1	3	2					
6	7	8	-8.063117, -34.943216	70	1	2	3					
7	7	6	-8.062395, -34.942818	190	3	17	20					
8	6	5	-8.061680, -34.942326	80	1	4	4					
9	4	5	-8.061866, -34.941699	69	1	4	3					
10	4	25	-8.061750, -34.941350	46	1	3	4					
11	25	23	-8.061346, -34.941941	160	2	7	7					
12	23	6	-8.061340, -34.942679	49	1	3	2					
13	23	24	-8.060880, -34.942606	49	1	3	0					
14	21	24	-8.060464, -34.943284	170	3	4	7					
15	24	26	-8.060915, -34.941839	160	2	8	8					
16	25	26	-8.061321, -34.941259	51	1	3	4					
17	3	11	-8.063326, -34.943722	160	2	1	0					
18	11	13	-8.062521, -34.943923	77	1	10	8					
19	11	18	-8.061889, -34.944799	170	3	12	0					
20	18	19	-8.060357, -34.945424	140	2	6	0					
21	19	30	-8.059484, -34.945563	66	1	8	0					
22	30	31	-8.058901, -34.945666	65	1	8	0					
23	19	20	-8.059917, -34.945030	99	2	9	11					
24	20	21	-8.060138, -34.944338	71	1	10	7					
25	20	16	-8.060520, -34.944715	96	2	5	9					
26	16	18	-8.060930, -34.944981	43	1	1	0					
27	16	131	-8.061010, -34.944469	71	1	4	5					
28	131	15	-8.061344, -34.944195	46	1	4	7					
29	131	14	-8.061207, -34.943886	55	1	5	2					
30	14	12	-8.061928, -34.943766	120	2	10	10					
31	14	9	-8.061336, -34.943427	56	1	9	4					
32	9	10	-8.062328, -34.943345	190	3	24	23					
33	9	6	-8.061477, -34.942950	58	1	5	6					
34	22	23	-8.060926, -34.943313	170	3	10	9					
35	21	22	-8.060462, -34.944071	51	1	4	4					
36	22	131	-8.060913, -34.944137	50	1	4	5					
37	31	57	-8.058597, -34.945841	31	1	3	0					
38	29	30	-8.059449, -34.944724	190	3	20	23					
39	21	29	-8.059927, -34.943994	63	1	7	0					
40	28	29	-8.059913, -34.943156	170	3	16	19					

41	24	28	-8.060394, -34.942529	64	1	4	0					
42	27	28	-8.060340, -34.941743	160	2	19	15					
43	26	27	-8.060823, -34.941137	62	1	9	6					
44	27	40	-8.060291, -34.941011	63	1	8	1					
45	40	41	-8.059327, -34.940774	150	2	20	15					
46	39	40	-8.059935, -34.941161	55	1	8	6					
47	38	39	-8.059788, -34.941639	56	1	8	6					
48	37	38	-8.059642, -34.942138	57	1	7	7					
49	28	37	-8.059846, -34.942448	60	1	7	1					
50	36	37	-8.059497, -34.942660	56	1	8	0					
51	35	36	-8.059348, -34.943153	57	1	9	3					
52	34	35	-8.059208, -34.943628	56	1	3	5					
53	29	34	-8.059378, -34.943922	61	1	4	6					
54	33	34	-8.059057, -34.944117	54	1	8	2					
55	32	33	-8.058928, -34.944558	49	1	9	1					
56	31	32	-8.058737, -34.945241	110	2	12	16					
57	56	57	-8.057603, -34.946157	170	3	50	0					
58	56	54	-8.057367, -34.945456	170	3	9	11					
59	54	55	-8.057371, -34.944646	46	1	5	3					
60	32	54	-8.058237, -34.944732	150	2	16	16					
61	52	54	-8.057642, -34.944444	52	1	0	0					
62	52	33	-8.058333, -34.944289	140	2	18	18					
63	52	51	-8.057768, -34.943961	59	1	2	0					
64	50	51	-8.057910, -34.943422	59	1	7	0					
65	50	49	-8.058044, -34.942964	55	1	7	0					
66	49	46	-8.058182, -34.942475	56	1	6	0					
67	46	45	-8.058327, -34.941917	63	1	7	3					
68	45	44	-8.058489, -34.941378	67	1	8	6					
69	44	41	-8.058639, -34.940824	55	1	6	7					
70	34	51	-8.058524, -34.943794	150	2	18	16					
71	35	50	-8.058685, -34.943310	150	2	19	18					
72	36	49	-8.058790, -34.942810	150	2	17	17					
73	46	37	-8.058923, -34.942330	150	2	18	17					
74	38	45	-8.059030, -34.941798	150	2	7	18					
75	39	44	-8.059204, -34.941250	150	2	5	15					
76	56	58	-8.056308, -34.946169	190	3	1	1					
77	58	59	-8.055529, -34.945907	53	1	2	3					
78	59	60	-8.055765, -34.945799	53	1	5	5					
79	59	61	-8.055687, -34.945363	68	1	8	4					
80	61	62	-8.055820, -34.944802	55	1	7	4					
81	62	63	-8.056035, -34.944671	41	1	4	1					
82	62	64	-8.055979, -34.944228	54	1	5	0					
83	64	65	-8.056149, -34.943714	55	1	4	0					
84	65	51	-8.057000, -34.943606	180	3	4	0					
85	65	66	-8.056279, -34.943233	64	1	4	0					

86	66	67	-8.056422, -34.942673	58	1	4	0					
87	67	68	-8.056563, -34.942177	47	1	3	0					
88	68	48	-8.057074, -34.942048	95	2	4	0					
89	48	46	-8.057888, -34.942174	89	2	3	0					
90	47	45	-8.057965, -34.941606	86	2	3	2					
91	43	44	-8.058129, -34.941019	91	2	3	3					
92	42	41	-8.058275, -34.940515	90	2	13	9					
93	42	43	-8.057836, -34.940670	60	1	4	4					
94	43	47	-8.057681, -34.941240	68	1	5	5					
95	48	47	-8.057529, -34.941821	63	1	4	4					
96	47	69	-8.057221, -34.941486	92	2	3	3					
97	43	70	-8.057339, -34.940874	88	2	3	3					
98	42	71	-8.057496, -34.940321	91	2	3	13					
99	68	69	-8.056716, -34.941662	66	1	4	4					
100	69	70	-8.056876, -34.941097	70	1	5	5					
101	70	71	-8.057053, -34.940463	65	1	5	5					
102	58	72	-8.055014, -34.946147	100	2	2	0					
103	59	73	-8.055154, -34.945599	98	2	3	3					
104	61	74	-8.055316, -34.945089	97	2	3	3					
105	62	75	-8.055436, -34.944557	89	2	3	4					
106	64	76	-8.055624, -34.943958	98	2	3	3					
107	65	77	-8.055779, -34.943438	98	2	3	2					
108	66	78	-8.055779, -34.943438	93	2	3	3					
109	67	79	-8.056054, -34.942388	97	2	3	3					
110	68	80	-8.056192, -34.941908	95	2	3	3					
111	69	81	-8.056356, -34.941349	99	2	3	3					
112	70	82	-8.056523, -34.940728	100	2	3	3					
113	71	83	-8.056688, -34.940135	100	2	3	3					
114	83	95	-8.055830, -34.939909	98	2	3	3					
115	82	94	-8.055638, -34.940534	95	2	3	3					
116	81	93	-8.055465, -34.941185	100	2	3	3					
117	80	92	-8.055333, -34.941751	100	2	3	3					
118	79	91	-8.055204, -34.942328	100	2	3	3					
119	78	90	-8.055084, -34.942848	110	2	3	3					
120	77	89	-8.054883, -34.943309	98	2	3	3					
121	76	88	-8.054769, -34.943916	100	2	3	3					
122	75	87	-8.054535, -34.944509	100	2	3	3					
123	74	86	-8.054450, -34.945033	100	2	3	3					
124	73	85	-8.054288, -34.945543	99	2	3	3					
125	72	84	-8.054193, -34.946124	100	2	3	0					
126	84	96	-8.053312, -34.946096	97	2	4	0					
127	85	97	-8.053409, -34.945509	100	2	4	4					
128	86	98	-8.053571, -34.944979	100	2	4	4					
129	87	99	-8.053732, -34.944425	97	2	4	4					
130	88	100	-8.053881, -34.943787	94	2	5	5					

131	89	101	-8.054027, -34.943193	95	2	4	2					
132	90	102	-8.054154, -34.942747	93	2	3	3					
133	91	103	-8.054344, -34.942203	98	2	3	3					
134	92	104	-8.054499, -34.941663	93	2	4	3					
135	93	105	-8.054640, -34.941017	93	2	4	4					
136	94	106	-8.054828, -34.940376	96	2	4	4					
137	95	107	-8.054993, -34.939692	99	2	8	13					
138	82	83	-8.056188, -34.940284	68	1	5	6					
139	81	82	-8.056011, -34.940926	73	1	6	5					
140	80	81	-8.055863, -34.941519	64	1	5	5					
141	79	80	-8.055723, -34.942063	61	1	4	4					
142	78	79	-8.055592, -34.942577	54	1	4	4					
143	77	78	-8.055449, -34.943075	63	1	4	5					
144	77	76	-8.055280, -34.943657	66	1	4	5					
145	76	75	-8.055133, -34.944207	69	1	5	5					
146	75	74	-8.054966, -34.944824	58	1	4	4					
147	74	73	-8.054832, -34.945270	58	1	4	4					
148	73	72	-8.054692, -34.945848	66	1	4	4					
149	84	85	-8.053804, -34.945748	68	1	3	3	12				
150	85	86	-8.053938, -34.945222	58	1	4	3	12				
151	86	87	-8.054074, -34.944714	61	1	4	4					
152	87	88	-8.054226, -34.944141	72	1	2	4	12				
153	88	89	-8.054387, -34.943531	44	1	4	6					
154	89	90	-8.054533, -34.943016	63	1	3	3					
155	90	91	-8.054659, -34.942552	70	1	4	4					
156	91	92	-8.054824, -34.941945	66	1	4	5					
157	92	93	-8.054973, -34.941409	75	1	5	5					
158	93	94	-8.055146, -34.940766	74	1	3	3					
159	94	95	-8.055332, -34.940099	76	1	6	6					
160	106	107	-8.054478, -34.939896	73	1	5	0					
161	105	106	-8.054327, -34.940544	77	1	4	0					
162	104	105	-8.054157, -34.941222	78	1	3	0					
163	103	104	-8.054000, -34.941801	57	1	2	0					
164	102	103	-8.053848, -34.942341	62	1	2	0					
165	101	102	-8.053695, -34.942854	56	1	2	0					
166	100	101	-8.053557, -34.943366	65	1	3	0					
167	99	100	-8.053381, -34.944016	80	1	3	0					
168	99	98	-8.053204, -34.944667	62	1	3	0					
169	98	97	-8.053061, -34.945206	46	1	3	0					
170	96	97	-8.052900, -34.945739	47	1	2	0					
171	108	101	-8.052805, -34.943035	180	3	1	0					
172	109	104	-8.053152, -34.941486	180	3	1	0					
173	110	107	-8.053783, -34.939411	180	3	25	0					
174	111	112	-8.051734, -34.944039	65	1	4	1					
175	112	108	-8.051912, -34.943346	92	2	7	0					

176	108	109	-8.052225, -34.942135	180	3	10	0					
177	109	110	-8.052753, -34.940203	240	3	24	0	9	32			
178	113	111	-8.051133, -34.944287	110	2	6	0					
179	114	112	-8.051172, -34.943701	130	2	7	7					
180	119	108	-8.051635, -34.942874	85	2	7	0	50				
181	115	119	-8.050938, -34.942778	73	1	3	6	12	6			
182	120	109	-8.051959, -34.941268	180	3	9	1	21	14			
183	116	120	-8.051067, -34.941141	100	2	7	6					
184	110	121	-8.052295, -34.939116	140	2	1	17					
185	118	121	-8.051184, -34.939417	140	2	9	0					
186	119	120	-8.051398, -34.942022	95	2	3	7	6	6	6	18	8
187	120	121	-8.051699, -34.940145	220	3	9	16	24	18	12	8	
188	113	114	-8.050597, -34.943934	55	1	3	0					
189	114	115	-8.050611, -34.943177	110	2	8	1					
190	115	116	-8.050636, -34.941854	180	3	10	5	9	9	6		
191	116	117	-8.050635, -34.940687	77	1	3	1	18				
192	117	118	-8.050650, -34.939980	82	2	8	5					
193	130	113	-8.050188, -34.944179	86	2	7	0					
194	130	129	-8.049765, -34.943322	170	3	13	2	12	48	12		
195	129	115	-8.050132, -34.942693	96	2	7	5					
196	129	128	-8.049661, -34.942044	110	2	7	7	16				
197	128	127	-8.049590, -34.941239	78	1	6	5					
198	127	116	-8.050151, -34.940994	120	2	3	6	4	8	4		
199	126	117	-8.050140, -34.940432	120	2	7	7	8				
200	126	125	-8.049483, -34.940210	57	1	3	0	18				
201	125	118	-8.049998, -34.939797	140	2	7	0					
202	125	123	-8.049462, -34.939791	37	1	0	0					
203	123	132	-8.050491, -34.939294	260	4	2	0	36	28			
204	121	132	-8.051738, -34.939058	44	1	4	0					
205	132	122	-8.051974, -34.938757	88	2	13	10	240				
206	110	122	-8.052646, -34.938668	150	2	22	18					
207	127	126	-8.049537, -34.940664	49	1	0	0	24				

APÊNDICE B – DADOS FLORES

Aresta	Vértices		Coordenadas Geográficas	c_{ij}	l_{ij}	Casas	
	i	j				i	j
1	1	2	-7.867291, -37.970994	230	3	33	26
2	2	5	-7.867317, -37.972669	60	1	12	14
3	2	3	-7.867125, -37.972356	19	1	1	0
4	3	4	-7.866889, -37.972334	32	1	3	0
5	3	6	-7.866967, -37.972764	110	2	10	7
6	4	7	-7.866525, -37.973274	210	3	27	26
7	5	6	-7.867253, -37.973248	62	1	2	3
8	5	9	-7.867531, -37.973341	59	1	6	5
9	9	10	-7.867751, -37.973802	42	1	1	0
10	8	10	-7.867335, -37.974073	88	1	2	9
11	7	8	-7.866723, -37.974085	62	1	10	9
12	7	17	-7.866126, -37.974130	73	1	12	8
13	17	18	-7.865729, -37.973515	100	1	10	15
14	18	19	-7.865741, -37.972729	98	1	4	9
15	4	19	-7.866174, -37.972306	120	2	11	9
16	19	20	-7.865036, -37.972291	160	2	15	24
17	20	53	-7.864242, -37.972575	73	1	7	3
18	18	53	-7.864973, -37.973080	170	2	14	20
19	20	21	-7.864299, -37.971569	130	2	9	15
20	21	22	-7.864138, -37.971163	39	1	2	0
21	22	23	-7.863558, -37.971066	89	1	11	0
22	23	24	-7.863517, -37.969985	190	2	27	0
23	24	25	-7.863674, -37.969027	49	1	5	0
24	25	28	-7.862841, -37.968768	180	2	17	18
25	28	27	-7.861997, -37.968423	71	1	5	6
26	26	27	-7.862657, -37.968192	160	2	13	24
27	28	29	-7.861997, -37.968963	46	1	5	0
28	29	30	-7.862577, -37.969182	180	2	18	12
29	24	30	-7.863407, -37.969235	140	2	4	0
30	30	31	-7.863165, -37.969507	67	1	4	6
31	31	106	-7.862464, -37.969639	140	2	16	13
32	29	106	-7.861980, -37.969322	36	1	3	4
33	31	32	-7.863056, -37.970053	50	1	6	0
34	33	106	-7.861967, -37.969730	57	1	7	0
35	33	32	-7.862482, -37.970133	130	2	8	17
36	33	36	-7.861928, -37.970219	48	1	5	0
37	35	36	-7.862123, -37.970539	49	1	5	1
38	35	34	-7.862601, -37.970655	83	1	8	7
39	34	32	-7.862976, -37.970473	63	1	4	8
40	34	23	-7.863079, -37.970986	51	1	2	2

41	36	37	-7.861832, -37.970679	53	1	4	8
42	37	38	-7.861994, -37.970992	50	1	10	6
43	38	35	-7.862275, -37.970797	52	1	3	0
44	38	39	-7.862510, -37.971118	72	1	14	10
45	39	34	-7.862888, -37.970991	42	1	5	3
46	39	40	-7.863399, -37.971349	130	2	12	12
47	40	41	-7.863854, -37.971699	52	1	7	7
48	41	42	-7.863548, -37.971840	44	1	4	0
49	41	44	-7.863550, -37.972140	69	1	5	0
50	44	42	-7.863337, -37.971990	54	1	3	10
51	42	43	-7.863137, -37.971731	59	1	4	7
52	43	39	-7.862910, -37.971467	49	1	3	2
53	43	45	-7.862770, -37.971988	77	1	8	0
54	44	45	-7.862954, -37.972355	73	1	6	7
55	44	51	-7.863206, -37.972504	59	1	2	0
56	45	46	-7.861898, -37.972122	190	2	12	6
57	46	37	-7.861524, -37.971391	140	2	1	8
58	74	46	-7.859945, -37.972753	300	4	8	0
59	74	73	-7.859131, -37.974538	130	2	13	3
60	73	72	-7.859445, -37.975474	110	1	2	3
61	73	75	-7.859246, -37.976179	220	3	6	11
62	75	76	-7.859543, -37.976901	80	1	4	6
63	76	72	-7.859673, -37.976073	150	2	10	11
64	72	71	-7.859887, -37.975419	52	1	5	8
65	76	77	-7.860026, -37.976400	83	1	7	0
66	77	71	-7.860182, -37.975796	87	1	14	13
67	77	68	-7.860991, -37.976015	200	3	2	0
68	71	69	-7.860930, -37.975295	180	2	20	23
69	68	69	-7.861629, -37.975544	100	1	12	15
70	69	70	-7.861093, -37.974906	150	2	22	29
71	69	62	-7.861786, -37.974826	97	1	5	8
72	68	67	-7.861816, -37.975945	43	1	4	0
73	67	62	-7.861988, -37.975197	130	2	19	21
74	62	63	-7.861721, -37.974588	69	1	8	3
75	63	64	-7.861313, -37.974618	45	1	9	1
76	64	47	-7.861084, -37.973992	130	2	21	16
77	47	48	-7.861252, -37.973391	46	1	3	0
78	63	48	-7.861501, -37.973984	140	2	14	7
79	48	49	-7.861692, -37.973416	51	1	1	1
80	49	62	-7.861949, -37.973961	140	2	7	8
81	49	50	-7.862192, -37.973518	62	1	3	5
82	50	62	-7.862356, -37.974022	130	2	15	15
83	62	61	-7.862469, -37.974654	75	1	7	0
84	61	60	-7.862818, -37.974372	62	1	7	1
85	60	52	-7.862928, -37.973897	51	1	4	4

86	52	50	-7.862667, -37.973629	59	1	2	0
87	50	51	-7.862695, -37.972808	150	2	5	12
88	52	51	-7.863075, -37.973182	120	1	11	2
89	51	53	-7.863652, -37.972809	100	1	2	17
90	53	56	-7.864048, -37.973186	65	1	2	6
91	52	56	-7.863427, -37.973799	120	2	9	11
92	56	55	-7.864335, -37.974023	130	2	7	19
93	55	17	-7.865420, -37.974170	75	1	9	7
94	56	57	-7.863925, -37.974134	45	1	2	7
95	57	60	-7.863370, -37.974252	120	2	8	11
96	57	58	-7.863869, -37.974612	65	1	5	3
97	58	55	-7.864464, -37.974529	160	2	9	21
98	58	59	-7.863505, -37.974832	75	1	6	8
99	59	61	-7.862983, -37.974734	52	1	7	5
100	59	65	-7.863175, -37.975055	91	1	5	2
101	61	66	-7.862638, -37.975152	100	1	12	8
102	67	66	-7.862309, -37.975834	62	1	1	0
103	66	65	-7.862873, -37.975576	82	1	1	10
104	10	15	-7.868214, -37.975824	350	1	39	21
105	10	11	-7.867797, -37.974700	64	1	14	0
106	11	12	-7.867403, -37.975034	80	1	3	4
107	12	8	-7.867136, -37.974598	130	2	12	9
108	12	13	-7.867097, -37.975232	58	1	7	8
109	13	14	-7.867105, -37.975663	50	1	5	6
110	14	15	-7.867072, -37.976142	70	1	6	5
111	15	82	-7.866613, -37.976642	88	1	14	11
112	82	95	-7.865825, -37.977215	140	2	5	0
113	82	83	-7.866051, -37.976749	24	1	2	0
114	83	84	-7.865757, -37.976647	45	1	3	0
115	83	80	-7.866260, -37.975931	150	2	21	18
116	80	13	-7.866583, -37.975412	73	1	2	0
117	80	16	-7.866366, -37.975128	50	1	8	7
118	16	12	-7.866643, -37.974978	57	1	1	0
119	16	7	-7.866424, -37.974491	86	1	11	12
120	16	78	-7.866128, -37.974903	72	1	1	2
121	78	17	-7.865876, -37.974492	89	1	12	13
122	78	81	-7.865797, -37.975083	60	1	4	9
123	81	80	-7.866023, -37.975350	65	1	6	13
124	81	79	-7.865297, -37.975290	81	1	8	9
125	79	55	-7.865039, -37.974741	86	1	9	16
126	79	58	-7.864434, -37.975222	120	2	5	11
127	58	89	-7.864003, -37.975445	60	1	4	1
128	89	86	-7.864649, -37.975779	160	2	17	13
129	86	81	-7.865582, -37.975648	40	1	8	7
130	86	85	-7.865310, -37.976086	69	1	6	8

131	85	84	-7.865343, -37.976456	55	1	5	2
132	85	87	-7.864879, -37.976226	79	1	1	6
133	87	88	-7.864879, -37.976226	29	1	2	0
134	88	89	-7.864026, -37.975992	69	1	2	7
135	88	90	-7.863727, -37.976388	69	1	2	0
136	90	65	-7.863378, -37.975771	130	2	22	10
137	91	90	-7.863447, -37.976654	72	1	16	0
138	91	92	-7.863810, -37.977052	70	1	5	0
139	92	88	-7.864064, -37.976655	73	1	2	4
140	92	93	-7.864415, -37.977030	80	1	4	0
141	93	85	-7.864947, -37.976717	94	1	13	15
142	93	94	-7.864864, -37.977331	58	1	3	0
143	94	84	-7.865340, -37.976973	110	1	18	0
144	94	95	-7.865183, -37.977496	34	1	2	0
145	95	96	-7.864941, -37.978199	98	1	12	14
146	96	100	-7.864933, -37.978813	130	1	13	18
147	96	97	-7.864550, -37.978706	85	1	4	11
148	97	93	-7.864496, -37.977809	130	2	18	12
149	97	92	-7.864178, -37.977558	220	2	23	19
150	97	98	-7.863860, -37.978620	130	2	13	6
151	98	91	-7.863529, -37.977712	170	2	28	27
152	97	99	-7.864044, -37.979117	110	1	9	7
153	97	100	-7.864495, -37.979257	135	2	5	4
154	100	101	-7.864683, -37.980587	290	4	17	16
155	101	105	-7.864364, -37.982364	120	1	11	10
156	101	103	-7.863942, -37.981709	100	1	10	8
157	73	102	-7.858389, -37.973812	220	2	21	14
158	27	103	-7.860964, -37.967503	210	2	10	0
159	104	103	-7.860484, -37.967871	38	1	20	1
160	104	27	-7.860835, -37.968398	81	1	3	2