



UNICAMP

UNIVERSIDADE ESTADUAL DE CAMPINAS
FACULDADE DE CIÊNCIAS APLICADAS

Jhonatan León Huillca

UM ESTUDO SOBRE PROBLEMAS DE ROTEAMENTO DE VEÍCULOS COM
APLICAÇÃO EM UMA REDE DE DISTRIBUIÇÃO LOGÍSTICA

Limeira
2022



UNIVERSIDADE ESTADUAL DE CAMPINAS
FACULDADE DE CIÊNCIAS APLICADAS

Jhonatan León Huillca

UM ESTUDO SOBRE PROBLEMAS DE ROTEAMENTO DE VEÍCULOS COM
APLICAÇÃO EM UMA REDE DE DISTRIBUIÇÃO LOGÍSTICA

Dissertação apresentada à Faculdade de Ciências Aplicadas da Universidade Estadual de Campinas como parte dos requisitos exigidos para a obtenção do título de Mestre em Engenharia de Produção e de Manufatura, na área de Pesquisa Operacional e Gestão de Processos.

Orientador Prof. Dr. Washington Alves de Oliveira

Este exemplar corresponde à versão final da dissertação defendida pelo aluno Jhonatan León Huillca e orientada pelo Prof. Dr. Washington Alves de Oliveira

Limeira
2022

Ficha catalográfica
Universidade Estadual de Campinas
Biblioteca da Faculdade de Ciências Aplicadas
Ana Luiza Clemente de Abreu Valério - CRB 8/10669

H876e Huillca, Jhonatan Leon, 1992-
Um estudo sobre problemas de roteamento de veículo com aplicação em uma rede de distribuição logística / Jhonatan Leon Huillca. – Limeira, SP : [s.n.], 2022.

Orientador: Washington Alves de Oliveira.
Dissertação (mestrado) – Universidade Estadual de Campinas, Faculdade de Ciências Aplicadas.

1. Problema de roteamento de veículos. 2. Heurística. 3. Logística. I. Oliveira, Washington Alves de, 1977-. II. Universidade Estadual de Campinas. Faculdade de Ciências Aplicadas. III. Título.

Informações para Biblioteca Digital

Título em outro idioma:

A study on vehicle routing problems with application in a logistics distribution network

Palavras-chave em inglês:

Vehicle routing problem

Heuristic

Logistics

Área de concentração: Pesquisa Operacional e Gestão de Processos

Titulação: Mestre em Engenharia de Produção e de Manufatura

Banca examinadora:

Washington Alves de Oliveira [Orientador]

Cristiano Torezzan

Angelo Aliano Filho

Data de defesa: 08-03-2022

Programa de Pós-Graduação: Engenharia de Produção e de Manufatura

Identificação e informações acadêmicas do(a) aluno(a)

- ORCID do autor: <https://orcid.org/0000-0001-5387-832>

- Currículo Lattes do autor: <http://lattes.cnpq.br/9505278175989670>

COMISSÃO JULGADORA - DISSERTAÇÃO DE MESTRADO

Candidato: Jhonatan León Huillca

RA: 229781

Data de Defesa: 08 de março 2022

Instituição: Faculdade de Ciências Aplicadas, Universidade Estadual de Campinas

Título da dissertação: "*Um estudo sobre problemas de roteamento de veículos com aplicação em uma rede de distribuição logística*"

Banca Examinadora

Prof. Dr. Washington Alves de Oliveira (Presidente)

UNICAMP - Limeira

Prof. Dr. Cristiano Torezzan

UNICAMP - Limeira

Dr. Angelo Aliano Filho

UTFPR - Curitiba

A Ata de Defesa, com as respectivas assinaturas dos membros da Comissão Julgadora, encontra-se no SIGA (Sistema de Fluxo de Dissertação) e na Secretaria de Pós-Graduação da Faculdade de Ciências Aplicadas.

Agradecimentos

Este trabalho de dissertação foi desenvolvido com ajuda de muitas pessoas que estiveram comigo no trajeto do mestrado. Em primeiro lugar, quero agradecer a Deus pela oportunidade de continuar crescendo como pessoa e como profissional.

Aos meus pais, Mauro Leon Huaqui e Gabina Huilca Javier por me formar como uma pessoa de valores e virtudes na sociedade, a meu irmão Edwind Leon Huilca por sempre ser meu apoio emocional e ter acreditado em mim e finalmente a minha irmã Elizabeth Leon Huilca por dedicar seus últimos dias a tornar minha vida mais feliz.

Ao Prof. Dr. Washington Alves de Oliveira, por confiar e acreditar na minha capacidade, pela orientação e aprendizagem, e mais do que isso, pela plena disposição e amizade desde o primeiro momento que o conheci.

Aos meus amigos que fiz no mestrado, Fernando, Maria Paula, Alejandra, Felipe, Gabriel, Taka e Wilson por fazer me sentir em casa em todo momento, pelas dicas e orientações em algumas disciplinas, e sobre tudo pela sincera amizade.

Quero agradecer á Faculdade de Ciências Aplicadas pela oportunidade de dar continuidade na minha formação como profissional, também ao programa da Pós-graduação em Engenharia de Produção e de Manufatura por contar com professores e alunos de alta qualidade, que faz ser uma das melhores universidades de América Latina

Aos professores e companheiros da FCA pelo nível da educação e companhia ao longo da minha estadia no Brasil e pelas suas disponibilidades nas situações pessoais mais difíceis que passei. Aos trabalhadores do restaurante universitário e a todos os funcionários que fizeram o trabalho de sempre dar suporte à faculdade para que os alunos desenvolvam seu papel de estudante.

Obrigado pela oportunidade

Esta dissertação apresenta um estudo sobre o problema de roteamento de veículos, fornecendo informações sobre uma classe de variantes encontradas na literatura, partindo da modelagem clássica até os modelos mais elaborados desenvolvidos ao longo do tempo por meio da incorporação de novos parâmetros e restrições que representam problemas práticos em redes de distribuição logística. Além disso, este estudo descreve diferentes métodos de resolução, incluindo técnicas conhecidas de otimização, heurísticas e metaheurísticas propostas e desenvolvidas por diversos autores. Especificamente, a variante do problema de roteamento de veículos com múltiplos depósitos foi estudada e aplicada a um estudo de caso envolvendo a entrega de produtos de um empresa de logística peruana de médio porte. Uma combinação híbrida englobando métodos exatos e heurísticos de resolução foi a estratégia usada para construir rotas com a menor distância possível para distribuição de produtos no estudo de caso. A utilização direta do software comercial para a resolução de problemas de programação linear inteira foi escolhido para a implementação do método exato e da heurística proposta. O método aproximado escolhido refere-se a uma rotina de duas fases, em que a primeira fase envolve o agrupamento de vértices por algum tipo de similaridade e a segunda fase constrói uma rota específica para cada agrupamento. Os algoritmos Concorde e Lin-Kernighan-Helsgaun, prestigiados na literatura para o problema do caixeiro viajante, foram usados na segunda fase da heurística proposta. No estudo de caso, a empresa em análise fabrica e distribui os seus produtos para clientes localizados na região metropolitana da cidade de Lima, utilizando dois depósitos e uma frota de veículos com capacidade homogênea. Os resultados computacionais obtidos por meio da metodologia proposta derivados da resolução de um conjunto de instâncias mostraram que método heurístico combinado com o algoritmo Lin-Kernighan-Helsgaun foi mais eficiente na maioria dos casos analisados. Portanto, a metodologia proposta nesta dissertação considerando os parâmetros do estudo de caso na resolução da versão do problema com múltiplos depósitos poderiam ser estendido para casos similares de outras empresas.

Palavras-chaves: Problemas de roteamento de veículos, Heurística, Logística.

This dissertation presents a study on the vehicle routing problem, providing information about a class of variants from the literature, starting at the classical modeling to the more elaborate models developed over time by incorporating new parameters and constraints that represent practical problems in logistics distribution networks. Furthermore, this study describes different resolution methods, including known optimization techniques, heuristics, and metaheuristics proposed and developed by diverse authors. Specifically, the variant of the vehicle routing problem with multiple depots was studied and applied to a case study involving the delivery of products from a medium-sized Peruvian logistics company. The strategy used to build routes of minimum distance in the case study was a hybrid combination encompassing exact and heuristic resolution methods. The direct use of commercial software for solving integer linear programming problems was chosen for the implementation of the exact method and the proposed heuristic. The approximate method chosen refers to a two-phase routine, in which the first phase involves clustering vertices by some kind of similarity and the second phase builds a specific route for each cluster. The Concorde and Lin-Kernighan-Helsgaun algorithms, renowned in the literature for the traveling salesman problem, were used in the second phase of the proposed heuristic. In the case study, the company under analysis manufactures and distributes its products to customers located in the metropolitan region of the city of Lima, using two warehouses and a fleet of vehicles with homogeneous capacities. In the case study, the company under analysis manufactures and distributes its products to customers located in the metropolitan region of the city of Lima, using two warehouses and a fleet of vehicles with homogeneous capacity. The computational results obtained through the proposed methodology are derived of the resolution of a set of instances showed that heuristic method combined with the algorithm Lin-Kernighan-Helsgaun was more efficient in most cases analyzed. Therefore, the methodology proposed in this dissertation considering the parameters of the case study in solving the version of the problem with multiple deposits could be extended to similar cases from other companies.

Keywords: Vehicle routing problems, Heuristics, Logistics.

Lista de Figuras

2.1	Ilustração de uma solução típica para o PRV. Fonte: figura adaptada de Zhou et al. (2018).	22
2.2	Variantes do PRV de acordo com as restrições. Fonte: figura adaptada de Eksioglu et al. (2009).	26
2.3	Ilustração de uma configuração para o PRVMD. Fonte: figura adaptada de Ho et al. (2008).	37
3.1	Resumo das metodologias para resolução de problemas de roteamento de veículos	39
3.2	Representação da heurística C&W. Fonte: figura adaptada de Clarke & Wright (1964).	45
3.3	Representação do algoritmo LK. Fonte: figura adaptada de Helsgaun (2000).	47
3.4	Representação da heurística de Varredura. Fonte: figura retirada de Nurcahyo et al. (2002).	51
3.5	Representação da heurística de duas fases. Fonte: figura retirada de Beasley (1983).	53
3.6	Representação da meta-heurística AOC. Fonte: figura retirada de Dorigo et al. (2006).	56
3.7	Tipos de técnicas de agrupamento. Fonte: figura adaptada de Jain et al. (1999).	58
3.8	Técnicas de agrupamento não supervisionado. Fonte: figura adaptada de Jain et al. (1999).	59
3.9	Representação do Dendrograma. Fonte: figura adaptada de Jain & Dubes (1988).	60
3.10	Representação da técnica do cotovelo. Fonte: figura adaptada de Nainggolan et al. (2019).	63
4.1	Esquema da metodologia para o estudo de caso	65
4.2	Localização geográfica da empresa em estudo	66
4.3	Localizações dos depósitos e uma amostra aleatória dos clientes	68
4.4	Representação das coordenadas geográficas	68
4.5	Representação da metodologia heurística	72
4.6	Representação da primeira fase da heurística – <i>Clustering</i>	73
4.7	Designação dos clientes e formação dos <i>clusters</i>	74
4.8	Representação da segunda fase da heurística – <i>Routing</i>	75
4.9	GAP das instâncias testadas via solver	77
4.10	Comportamento dos GAPs das instâncias testadas	78
4.11	Análise dos custos obtidos via heurística para 50, 100, 150 e 200 vértices	80
4.12	Análise dos tempos obtidos via heurística para 50, 100, 150 e 200 vértices	81
4.13	Análise dos custos obtidos via algoritmo Concorde para 50, 100, 150 e 200 vértices	84
4.14	Análise dos tempos obtidos via algoritmo Concorde para 50, 100, 150 e 200 vértices	84
4.15	Análise dos custos obtidos via algoritmo LKH para 50, 100, 150 e 200 vértices	87
4.16	Análise dos tempos obtidos via algoritmo LKH para 50, 100, 150 e 200 vértices	88

4.17	Análise dos resultados dos testes computacionais das instâncias 50a, 100a, 150a e 200a . . .	89
4.18	Análise dos resultados dos testes computacionais das instâncias 50b, 100b, 150b e 200b . . .	91
4.19	Análise dos resultados dos testes computacionais das instâncias 50c, 100c, 150c e 200c . . .	93
4.20	Análise dos resultados da heurística em instâncias de 500, 1000, 1500, 200 e 2500 clientes .	94

Lista de Tabelas

4.1	Informações da empresa em análise para o estudo de caso	69
4.2	Serviço de distribuição desde o dia 16 até 25 de junho da empresa em análise	70
4.3	Instâncias criadas para os testes da metodologia	71
4.4	Resultados computacionais obtidos via solver para 50, 100, 150 e 200 vértices	77
4.5	Resultados computacionais obtidos via heurística para 50 e 100 vértices	79
4.6	Resultados computacionais obtidos via heurística para 150 e 200 vértices	80
4.7	Resultados computacionais obtidos via algoritmo Concorde para 50 e 100 vértices	82
4.8	Resultados computacionais obtidos via algoritmo Concorde para 150 e 200 vértices	83
4.9	Resultados computacionais obtidos via algoritmo LKH para 50 e 100 vértices	85
4.10	Resultados computacionais obtidos via algoritmo LKH para 150 e 200 vértices	87
4.11	Resultados dos testes computacionais das instâncias 50a, 100a, 150a e 200a	89
4.12	Resultados dos testes computacionais das instâncias 50b, 100b, 150b e 200b	91
4.13	Resultados dos testes computacionais das instâncias 50c, 100c, 150c e 200c	92
4.14	Resultados computacionais das instâncias de 500, 100, 1500, 200, 2500 e 6804 clientes	94

Introdução	13
1 Apresentação do Problema Abordado e Revisão Bibliográfica	16
1.1 Pesquisa em Estudo	16
1.2 Justificativa da Pesquisa	18
1.3 Objetivos	20
2 Problemas de Roteamento de Veículos	21
2.1 O Problema do Caixeiro Viajante como base do PRV	22
2.2 Elementos do Problema de Roteamento de Veículos	23
2.2.1 Clientes	24
2.2.2 Depósitos	24
2.2.3 Veículos	24
2.2.4 Modelo Clássico para o Problema de Roteamento de Veículos	24
2.3 Variantes do Problema de Roteamento de Veículos	25
2.3.1 Problema de Roteamento de Veículos Capacitado	25
2.3.2 Problema de Roteamento de Veículos com Janelas de Tempo	28
2.3.3 Problema de Roteamento de Veículos com Coleta e Entrega	29
2.3.4 Problema de Roteamento de Veículos Periódico	31
2.3.5 Problema de Roteamento de Veículos com Entrega Dividida	32
2.3.6 Problema de Roteamento de Veículos com Demanda Estocástica	34
2.3.7 Problema de Roteamento de Veículos com <i>Backhauls</i>	35
2.3.8 Problema de Roteamento de Veículos com Múltiplos Depósitos	37
3 Métodos de Resolução	39
3.1 Métodos Exatos	40
3.1.1 Programação Linear Inteira	40
3.1.2 Programação Dinâmica	41
3.1.3 Métodos de Busca em Árvore	41

3.2	Heurísticas	44
3.2.1	Heurísticas Construtivas	44
3.2.2	Heurísticas de Duas Fases	49
3.2.3	Heurísticas de Melhoramento	54
3.3	Meta-heurísticas	55
3.3.1	Algoritmo de Busca Tabu	55
3.3.2	Algoritmo Genético	55
3.3.3	Algoritmo de Colônias de Formiga	56
3.4	Técnicas de Agrupamento	58
3.4.1	Agrupamento Supervisionado	58
3.4.2	Agrupamento não Supervisionado	59
3.4.3	Algoritmo <i>k-means</i>	61
3.4.4	Algoritmo <i>k-means</i> na Formação de Grupos	61
3.4.5	Regra do Cotovelo para a Análises de <i>k-Cluster</i>	63
4	Aplicação da Pesquisa em um Estudo de Caso	65
4.1	Informações Considerados para o Estudo de Caso	66
4.1.1	Caracterização da Procura de uma Solução	67
4.2	Desenvolvimentos Metodológicos para o Estudo de Caso	67
4.2.1	Componentes Sistemáticos Abordados no Estudo	67
4.2.2	Componentes Empíricos Descrito no Estudo	69
4.2.3	Problema de Roteamento com Múltiplos Depósitos Abordado no Estudo	69
4.2.4	Algoritmo Exato na Procura de uma Solução	70
4.2.5	Algoritmo Heurístico na Procura de uma Solução	72
4.2.6	Primeira Fase da Heurística	73
4.2.7	Segunda Fase da Heurística	75
4.3	Ferramentas Utilizadas	75
4.3.1	Linguagem de Programação Julia	75
4.3.2	Informações do Computador	76
4.4	Testes Computacionais e Análise dos Resultados	76
4.4.1	Resultados e Análise dos Testes Computacionais para a Metodologia Exata	76
4.4.2	Resultados e Análise dos Testes Computacionais para a Metodologia Heurística	78
4.4.3	Análise dos Resultados dos Testes Computacionais para as Metodologias	88
4.4.4	Uso da Heurística LKH em Instâncias com Maior Número de Clientes	93
5	Conclusões e Próximos Passos	95
5.1	Conclusões Gerais	95
5.2	Conclusões para o Estudo de Caso	96
5.3	Próximos Passos	97
	Bibliografia	97

Introdução

Uma das aplicações para os problemas de roteamento de veículos é a busca por soluções para a distribuição dos produtos e serviços, projetando rotas que atendam a demanda dos clientes geograficamente distanciados a partir de um armazém. [Dantzig & Ramser \(1959\)](#) foram um dos primeiros autores que estudaram o problema de roteamento de veículos como uma extensão do problema do caixeiro viajante. Com o passar do tempo foram projetados novos modelos baseados no problema de roteamento de veículos, dos quais se desenvolveram gradualmente em conjunto com os métodos de resolução. Tanto os modelos quanto os métodos de resolução são de extrema importância para os sistemas de distribuição e, por essa razão, foram amplamente estudados durante os últimos 60 anos ([Montoya-Torres et al. 2015](#)).

Uma variante muito estudada é aquela que considera mais de um depósito e que, segundo [Solak et al. \(2014\)](#), nasceu no planejamento da distribuição de banco de alimentos. Neste caso, os produtos são coletados e armazenados nos diferentes depósitos e então são distribuídos por um conjunto de veículos para um grupo de clientes localizados em longas distâncias dos depósitos ([Lim & Wang 2005](#)). Dadas as localizações dos depósitos e clientes, o problema com múltiplos depósitos requer designar clientes e criar rotas que são percorridas por veículos. O objetivo principal é atender a todos os clientes, minimizando a distância total das viagens com a condição de que a demanda total precisa ser atendida e não se pode exceder a capacidade do veículo para cada rota ([Crevier et al. 2007](#)). Geralmente as rotas começam e terminam nos depósitos.

O problema de roteamento de veículos é de difícil solução devido à magnitude combinatória gerada pelas diferentes opções de designar veículos aos diversos pontos de distribuição (vértices) disponíveis para encontrar as rotas ideais no atendimento da demanda por produtos. A literatura especializada classifica este problema como NP-difícil, o que implica que não é possível encontrar uma solução exata usando métodos de otimização em tempo polinomial em função da dimensão do problema ([Ho et al. 2008](#)). Por esses motivos, inúmeras metodologias de resoluções propostas na literatura são baseadas em técnicas heurísticas e meta-heurísticas que, embora não garantam a obtenção da solução ótima, produzem em ocasiões soluções de qualidade com baixo tempo computacional ([Montoya-Torres et al. 2015](#)).

Diferentes necessidades e restrições práticas no campo da logística distributiva foram consideradas no desenvolvimento de novos modelos para o roteamento de veículos com a finalidade de simular de forma precisa os problemas de transporte e, assim, decidir o uso dos recursos, minimizar os custos e satisfazer as demandas dos clientes ([Mateos et al. 2013](#)). Para configurar uma rede logística, importantes decisões

estratégicas precisam ser tomadas de tal forma a ajudar a determinar, por exemplo; a quantidade de pontos de vendas e centros de distribuição, a localização destas, o tamanho adequado, as formas de transporte e a designação de veículos. Além disso, em todo o processo logístico é importante não gerar atrasos nos envios dos produtos, evitar ter produtos ausentes na hora da distribuição e sobretudo reduzir custos e outras séries de inconvenientes que colocam em risco as operações. Porém a tomada de decisões em prol de diminuir as dificuldades na distribuição logística baseada em ferramentas de modelagem computacional, com o passar do desenvolvimento, têm ganhado maior importância nas empresas, devido aos seus impactos positivos para superar as complicações vistas na ação da distribuição (Niknejad & Petrovic 2014).

De modo a discutir e contribuir com soluções para o planejamento da distribuição de produtos, esta dissertação estuda alguns algoritmos e modelagens para o problema de roteamento de veículos. Além disso, apresenta uma revisão teórica e a implementação de alguns métodos de resolução. O trabalho também inclui um estudo de caso envolvendo dados reais de uma empresa peruana, que por motivos de confidencialidade não será revelada o nome, a qual necessita alcançar constante melhorias na gestão de distribuição das suas mercadorias. Para começar a resolver o problema da empresa em análise foi necessário identificar alguns parâmetros que foram fornecidos pelo coordenador de distribuição, constituídos pelo número de depósitos, número de clientes atendidos por dia, demanda dos clientes por dia, número total de veículos, número de veículos alocados em cada depósito, capacidade dos veículos, número máximo de clientes atendidos por veículo, número de horas de trabalho do veículo e uma planilha das localizações geográficas dos clientes e dos depósitos. Considerando estes parâmetros, foi definido um modelo matemático baseado no problema de roteamento de veículos com múltiplos depósitos como uma forma de resolvê-lo. Para poder compreender a efetividade da metodologia exata se criaram pequenas instâncias de 50, 100, 150 e 200 clientes onde os parâmetros para os testes dessas instâncias foram proporcional aos parâmetros reais onde se atendem mais de 6000 clientes por dia, usando 90 veículos tendo cada um 5 toneladas de capacidade. Os testes apresentaram dificuldade de resolução para a instâncias a partir de 200 clientes, o que foi superado a com a implementação de um método híbrido de resolução aproximada. Esta combinação híbrida engloba métodos exatos e heurísticos de resolução como estratégia para construir rotas de distância mínima para distribuição de produtos. O método híbrido é composto por uma rotina de duas fases (aqui também foram testadas as mesmas instâncias criadas para o método exato). A primeira fase envolve o agrupamento de vértices por similaridade usando uma adaptação da técnica *k-means* que é baseada em um modelo de programação matemática, esta fase é resolvida de forma exata utilizando o solver Gurobi. A segunda fase constrói uma rota específica para cada agrupamento utilizando os algoritmos exatos para o PCV baseados na programação linear e métodos de *Branch and Bound*, além disso são usados os algoritmos Concorde e Lin-Kernighan-Helsgaun, os quais são algoritmos especializados para resolver o PCV. As metodologias propostas para os testes computacionais foram implementadas na linguagem de programação Julia limitando o tempo de execução em 30 minutos. Os resultados computacionais obtidos por meio da metodologia proposta derivados da resolução de um conjunto de instâncias mostraram que método heurístico combinado com o algoritmo Lin-Kernighan-Helsgaun foi mais eficiente na maioria dos casos analisados. Portanto, a metodologia proposta nesta dissertação considerando os parâmetros do estudo de caso na resolução da versão do problema com múltiplos depósitos poderiam ser estendido para casos similares de outras empresas.

O conteúdo do texto desta dissertação está organizado em cinco capítulos como pode-se ser confe-

rido abaixo. O Capítulo 1 problematiza a pesquisa desenvolvida com enfoque nas complicações para o planejamento da logística distributiva e nas dificuldades de empregar técnicas baseadas em modelos matemáticos, apresenta a importância do uso de métodos quantitativos na resolução das variantes do problema de roteamento de veículos e descreve os objetivos gerais e específicos da pesquisa. O Capítulo 2 apresenta conceitualmente o problema de roteamento de veículos e detalha as suas variantes mais importantes no contexto das aplicações desta dissertação, refletidas por diferentes autores na literatura ao longo do tempo. O Capítulo 3 descreve os diferentes métodos numéricos que tiveram sucesso na resolução dos problemas de roteamento de veículos. Em geral, esses métodos estão associados com programação linear inteira, baseados em ramificação e enumeração implícita e combinam técnicas heurísticas e meta-heurísticas com metodologias desenvolvidas em outras áreas, como é o caso das técnicas de agrupamento. O Capítulo 4 descreve uma série de experimentos computacionais que foram dirigidos ao estudo de caso com dados fornecidos por uma empresa que atua na distribuição de mercadorias. Assim, baseados nos conceitos apresentados nos capítulos anteriores, vários exemplares de um modelo considerando múltiplos depósitos foram resolvidos. Os resultados numéricos obtidos foram analisados utilizando um conjunto de estatísticas descritivas. Finalmente, o Capítulo 5 apresenta as conclusões e fornece algumas ideias como próximos passos para esta pesquisa.

Apresentação do Problema Abordado e Revisão Bibliográfica

1.1 Pesquisa em Estudo

Por meio do problema de roteamento de veículos é possível determinar um conjunto de rotas para os mesmos, dos quais percorrem um conjunto de vértices (clientes) uma única vez, tais rotas devem começar e terminar em uma única vértice (depósito ou armazém), considerando que os veículos possuem uma capacidade fixa (Eksioglu et al. 2009). As rotas formadas são aquelas que minimizam os custos de transporte e atendem as demandas dos clientes alocados em diferentes zonas geográficas (Jaillet et al. 2016). Os problemas de roteamento não capturam todos os aspectos importantes dos problemas de distribuição, porque alguns parâmetros usados, como a demanda por produtos e as distâncias entre cliente, não são conhecidos com absoluta certeza e não devem ser definidas como variáveis determinísticas (Lenstra & Kan 1981).

Existem muitas variantes para o problema de roteamento de veículos que visam resolver inumeráveis situações ligados à redução de custo de transporte e minimização do tempo. A maioria baseia-se em: (i) o número de depósitos e clientes atendidos por cada rota; (ii) a existência de janelas de tempo; (iii) a presença de restrições de capacidade dos veículos; (iv) a possibilidade de múltiplas visitas em cada cliente; e (v) o número de rotas que os veículos operam no mesmo período de planejamento (Toth & Vigo 2002, 2014, Golden et al. 2008).

Uma das variantes mais conhecidas para este problema considera a situação onde a distribuição das mercadorias é realizada a partir de mais de um depósito (Crevier et al. 2007). Esta variante pode ser resolvido como múltiplos depósitos individuais apenas no caso em que é possível agrupar os clientes ao redor de cada depósito individualmente. Caso contrário, a abordagem baseada em vários depósitos deve ser usada, onde os clientes precisam ser atendidos a partir de qualquer um dos depósitos usando a frota de veículos disponível. Nesta abordagem, os múltiplos depósitos atendem um conjunto de clientes usando veículos, cada veículo parte de um dos depósitos e retorna ao mesmo depósito pré-atribuído, reduzindo o custo de transporte, a distância e o tempo com o objetivo de garantir a qualidade do serviço na entrega das mercadorias. O problema com múltiplos depósitos atraiu considerável atenção entre pesquisadores e profissionais (Montoya-Torres et al. 2015).

Utilizar modelos para o problema de roteamento de veículos é um caminho usado por diferentes ad-

ministradores para otimizar e melhorar o planejamento do transporte. No entanto, existe uma dificuldade importante em resolver cada modelo. Muitos autores escolheram diversos caminhos para estudar a sua complexidade, pois o esforço computacional necessário para achar uma solução aumenta exponencialmente com a dimensão do problema (Niazy & Badr 2012, Archetti et al. 2011). Assim, muitos trabalhos seguiram a linha de construir métodos numéricos de resolução, sendo que uma grande parte deles são classificados como algoritmos exatos de otimização (Aarts et al. 2003). Esses, por vez, fornecem soluções ótimas, porém o tempo e os recursos computacionais usados são altos. Para superar essas dificuldades que estão relacionadas com a complexidade do problema, vários estudos foram desenvolvidos, como os propostos por Ho et al. (2008), Golden et al. (2008), Montoya-Torres et al. (2015), dentre outros, com a aplicação de métodos heurísticos e meta-heurísticos para a busca de soluções próximas da solução ótima, porém com menor custo computacional. Na sequência, alguns problemas cotidianos na rede de distribuição logística são descritos, e é discutido como uma boa gestão do transporte pode ser baseada em soluções viáveis obtidas pela aplicação de algoritmos exatos ou aproximados.

Na rede de distribuição logística os produtos são entregues diariamente em diferentes localizações de uma região a partir de uma origem (ponto de distribuição ou depósito), sendo os objetivos das distribuidoras determinar a quantidade de veículos e suas rotas necessárias para a entrega das mercadorias de modo a minimizar o custo total do transporte. Algumas empresas que tomam decisões relacionadas ao desenho de redes de distribuição (DRD) na distribuição dos produtos desde os depósitos até os clientes realizam de forma empírica, ou seja, sem usar ferramentas e suportes modernos de análise que contribuam para a otimização de seus processos, embora saibam da existência de novas tendências tecnológicas desenvolvidas para otimizar e agilizar a entrega das mercadorias. Por essa razão, surgem muitos erros na forma de como percorrer as rotas onde o principal afetado é o cliente. Portanto, um eixo principal para alcançar uma vantagem competitiva na indústria de distribuição atual é uma melhor determinação de rotas. Riveros & Silva (2008) indicaram que os problemas relacionados ao DRD envolvem várias decisões a serem otimizadas; como a localização, a alocação, o roteamento e o estoque.

Um péssimo sistema de transporte sobre o desenho da rede de distribuição influencia no sucesso da logística distributiva, gerando grandes perdas dentro das empresas. O desenho e o uso adequado do sistema de transporte, fornece valor aos produtos quando estes são transportados no tempo previsto. Os custos associados ao transporte são altamente representativos, constituindo em torno de 50% a 60% dos gastos logísticos e estão diretamente ligados no relacionamento com os fornecedores, clientes e concorrentes (Mejía et al. 2010, Ballou & de Lemus 1991). Em geral, os desenhos das rotas tornam-se difíceis de estruturar por causa de fatores como: as tendências atuais de crescimento da demanda por mercadorias, tanto em volume quanto na qualidade, o tempo de trajetória dos veículos na distribuição dos produtos e, pela ação de um mercado cada vez mais competitivo. O que leva a uma necessidade de melhoria do serviço através de políticas de despacho logístico (Ramírez 2009). Nessas condições, são necessárias medidas não tradicionais como aquelas voltadas nas ampliações da infraestrutura para armazenar mercadorias, adquirir veículos maiores ou terceirizar serviços (Lacerda 2002). O sistema tradicional de transporte de produtos contempla o despacho de veículos desde os centros de armazenagem até os endereços definidos pelos seus clientes (Franceschini & Rafele 2000). Esses veículos devem percorrer uma rota adequada com o menor tempo possível de forma a minimizar os custos e, além disso, oferecer aos usuários prazos curtos de entrega, por tais motivos o

sistema de transporte precisa revalidar sua estrutura logística, planejando as rotas de distribuição de forma eficaz (Villalobos 2006).

O desenho de rotas para veículos constituem em um conjunto de problemas de decisão que afetam o planejamento no âmbito estratégico, tático e operativo, que senão são resolvidos de forma eficiente, implicam em queda significativa nos lucros das empresas (Rushton et al. 2014), por isso o planejamento e gerenciamento de redes de distribuição requer novas técnicas de otimização de rotas, dado que não só afeta o desenvolvimento das operações, mas também as decisões táticas e estratégicas.

Algumas dificuldades habituais na operação de distribuição e transporte logística são: (i) o estabelecimento de rotas para veículos com certa limitação de capacidade para distribuir mercadorias a um grupo de clientes; e (ii) a programação de horários ou precedência entre destinos para satisfazer o percurso das rotas. Diante dessas dificuldades que afetam a cadeia de distribuição logística e por consequência à empresa, busca-se soluções recorrendo aos algoritmos para o roteamento dos veículos com o menor custo. As rotas devem ser estabelecidas de forma que cada ponto seja visitado apenas uma vez por exatamente um veículo, começando e terminando em um depósito de modo que a demanda total de todos os clientes não exceda a capacidade do veículo (Bochtis & Sorensen 2009). Em suma, o planejamento e gestão de redes de distribuição geram uma grande variedade de problemas de decisão, onde o sucesso depende principalmente da otimização das operações, onde o espectro de soluções possíveis é enorme e também cresce exponencialmente com o número de destinos e com o tamanho da frota. A complexidade das variáveis impedem que a otimização seja, em muitas situações, acessível com as técnicas de soluções exatas. Felizmente, existem procedimentos alternativos que, embora não garantam a solução ótima, fornecem soluções de qualidade para os problemas do dia a dia.

1.2 Justificativa da Pesquisa

O problema de roteamento de veículos é considerado um dos problemas mais desafiadores em termos de modelagem e resolução que já forneceu estudos de grande sucesso na área de pesquisa operacional, ajudando nas últimas décadas a desenvolver soluções para o planejamento de frotas de veículos no transporte de mercadorias (Bochtis & Sorensen 2009). Um grande interesse surgiu em torno dos problemas de roteamento, refletido nas investigações realizadas ao longo da história, nas quais se destacam os seguintes estudos. Tillman (1969) desenvolveu uma metodologia de resolução por meio do algoritmo da economia de Clarke e Wright. Tan et al. (2007) apresentaram um algoritmo evolutivo que incorpora uma heurística específica de busca do local e um método de simulação de rotas que avalia cada solução do problema de roteamento de veículos. Além disso, Dedong et al. (2008) descreveram um algoritmo meta-heurístico híbrido que combina o algoritmo colônia de formigas com heurísticas de busca local.

Grande parte das técnicas usadas na procura de uma solução para os problemas de roteamento são aquelas propostas pelos algoritmos exatos, as quais baseiam-se na enumeração implícita, ramificação e limitações. Estes algoritmos são eficientes para problemas de dimensão modesta (Ignizio & Cavalier 1994). Embora com eles seja possível resolver problemas de qualquer tamanho, na prática não é assim, devido ao grande consumo computacional em problemas de dimensão razoável em relação aos dados de entrada. Outra parte das técnicas concentra-se no uso de métodos qualitativos que não avaliam todas as possibilidades

combinatória, uma vez que os algoritmos exatos não conseguem resolver os problemas mais complexos e obter soluções ótimas em tempo polinomial. Essas técnicas dividem-se em técnicas de busca local convencional (heurísticas) e técnicas de busca local inteligente (meta-heurísticas). Este último não apresenta uma base matemática formal sustentável, porém o seu desenvolvimento acontece de forma intuitiva e é muito útil para a determinação de boas soluções aproximadas (Ignizio & Cavalier 1994).

Neste sentido, esta dissertação aborda um estudo do problema de roteamento de veículos, em particular aqueles considerando mais de um depósito, a qual está constantemente presente na distribuição logística das empresas. Na busca de resolver esta variante do PRV, abordou-se os algoritmos exatos embora não foi possível determinar, dentro do limite de tempo, soluções para as instâncias com mais de 200 clientes. Para superar essa dificuldade, optou-se por metodologias de resolução aproximada, devido a uma certa simplicidade e menor uso de capacidade computacional na obtenção dos resultados para as instâncias testadas (Aarts et al. 2003). O aprendizado será aplicado em um caso de estudo para uma empresa peruana que enfrenta o planejamento da distribuição de produtos e gerenciamento de rotas. Essa aplicação mostra a importância do uso dos algoritmos para resolver as variantes do problema de roteamento de veículos as quais estão presentes na distribuição logística das mercadorias.

A cadeia de distribuição logística, pode ser definida como aquela estrutura ou rede, física ou virtual por meio onde todas as práticas comerciais são desenvolvidas, entre os agentes fornecedores, produtores, distribuidores e consumidores (Johnson et al. 1990), onde é necessário a busca e criação de valor em cada um dos processos que se desenvolvem em toda a cadeia, integrando todos os elos interligados, de forma a gerar redução significativa dos custos, baseada na correta administração e gestão da cadeia de fornecimentos. Portanto, o planejamento de rotas de entrega de produtos dentro da cadeia de distribuição logística favorece na redução dos custos da cadeia de fornecimento inteira.

Muitas empresas sustentam que os objetivos mais significativos da distribuição é obter as mercadorias necessárias, transportá-las aos locais com menor tempo e com um custo mais baixo possível. No entanto, é difícil encontrar um sistema de distribuição que possa simultaneamente maximizar o serviço ao cliente e minimizar os custos de distribuição, uma vez que estão envolvidos um elevado custo, transporte rápido e múltiplos armazéns, sendo estes os fatores que incrementam os custos (Kotler & Turner 1997), então procura-se buscar um equilíbrio que compensem os interesses contrapostos da empresa com os consumidores. Por isso, a importância da eficácia e a eficiência da gestão da distribuição adquire sua verdadeira magnitude quando se consideram os custos. Kotler & Turner (1997) indicaram que os principais elementos dos custos da distribuição são o transporte (37%), o controle das demandas (22%), o armazenamento (21%) e outros como a recepção de ordens, o serviço ao cliente, a distribuição e a administração (20%). Além disso, os autores acham que se pode gerar economias substanciais na área da distribuição, a qual foi descrita como “a última fronteira para obter economias nos custos” e “o continente escuro da economia”. Drucker (1962) descreveu as atividades logísticas como as “áreas menos realizadas e ao mesmo tempo mais promissoras dentro do mundo industrial”.

Ao notarem que se existe uma grande variação entre os custos logísticos, Ballou & de Lemus (1991) indicaram que esses números variam em 4% das vendas das empresas que produzem e distribuem mercadorias de alto valor, e mais de 32% nas vendas das empresas que fazem comércio com mercadorias de baixo valor. Os autores também destacaram que os custos de transporte representam entre um terço e dois terços dos

custos logísticos totais. Estima-se que os custos de distribuição representam quase metade dos custos logísticos totais em algumas indústrias, e em indústrias de alimentos e bebidas eles podem aumentar o custo das mercadorias em 70% (De Backer et al. 1997). Atingindo e diminuindo os alvos dos custos de distribuição, seja de forma mínima, podem resultar em enormes economias financeiras e reduzir os impactos ambientais da poluição e do ruído, além de aumentar significativamente a satisfação dos requisitos do cliente. Desta forma, resolver os problemas de distribuição dentro da rede logística torna-se uma das áreas notáveis da pesquisa operacional (Laporte et al. 2002).

1.3 Objetivos

Objetivo geral:

O objetivo geral desta dissertação é estudar modelos matemáticos para os problemas de roteamento de veículos e alguns algoritmos que tiveram sucesso na resolução desses modelos com a aplicação em um estudo de caso.

Objetivos específicos:

- Estudar e analisar as diferentes variantes para o problema de roteamento de veículos;
- Estudar os algoritmos exatos e os algoritmos aproximados de resolução;
- Estudar e analisar diferentes técnicas de agrupamento;
- Resolver o problema de distribuição de mercadorias e construção de rotas por meio dos algoritmos estudados para um estudo de caso de uma empresa peruana;
- Implementar uma metodologia de resolução exata para o PRVMD com o uso de um solver na empresa em estudo;
- Desenvolver e implementar uma metodologia heurística de resolução aproximada para o PRVMD na empresa em estudo;
- Estudar, analisar e implementar o algoritmo *k-means* no desenvolvimento da heurística de duas fases – agrupar primeiro e rotear depois;
- Analisar e discutir os resultados do algoritmo exato e o método aproximado nas instâncias testadas para o modelo da empresa em estudo;
- Propor um caminho para diminuir a distâncias o percurso dos veículos no transporte das mercadorias e ajudar no gerenciamento da distribuição logística via construção de rotas para a empresa em estudo.

Problemas de Roteamento de Veículos

O problema de roteamento de veículos (PRV) resumidamente pode ser descrito como um mecanismo muito útil para traçar um conjunto de rotas de veículos em um sistema de atendimento de demanda por produtos solicitados por diversos clientes espalhados em uma região mapeada. Em geral, cada veículo (em sua respectiva rota) inicia em um determinado depósito central, atende um conjunto de clientes que demandou por uma parcela de produtos e retorna ao mesmo depósito. Em particular, cada cliente é atendido uma única vez por um único veículo e a capacidade em relação à carga ou ao tempo de deslocamento de cada veículo é limitada. Adicionalmente, o conjunto de rotas deve fazer o atendimento aos clientes de tal forma que o custo total do sistema seja minimizado. O PRV descrito acima apresenta muitas variações dependendo da aplicação e do tipo de sistema de atendimento aos clientes.

[Dantzig & Ramser \(1959\)](#) foram um dos primeiros autores a elaborar modelos e propor aplicações para o PRV. Eles apresentaram uma aplicação na entrega de gasolina a estações de serviço, propondo uma formulação matemática que facilitou representar o problema. Esta formulação surgiu como uma generalização do problema do caixeiro viajante, no qual um vendedor visita uma série de clientes, cada cliente uma única vez, e retorna ao local de partida. [Clarke & Wright \(1964\)](#) propuseram o algoritmo da economia para resolver o PRV, que ficou muito conhecido devido a sua simplicidade e foi uma das primeiras técnicas de resolução aproximada para o problema. A partir desses trabalhos, iniciou-se muitas investigações e contribuições de pesquisadores para o PRV, algumas destas pesquisas buscaram resolvê-lo empregando a técnicas de programação linear inteira e métodos aproximados.

[Toth & Vigo \(2002\)](#) descreveu uma série de variantes para o PRV que envolvem as diferentes características dos clientes, dos depósitos, dos veículos e das múltiplas restrições operacionais sobre as rotas. As restrições de maior relevância que dão origem a novas formas para o PRV são as seguintes:

- Cada veículo tem capacidade limitada (PRV Capacitado);
- Cada cliente deve ser visitado dentro de uma faixa de horário (PRV com Janelas de Tempo);
- Vários locais de início de rotas (PRV com Múltiplos Depósitos);
- Os clientes podem ser atendido por mais de um veículo (PRV com Entrega Dividida);

- Algumas variáveis do problema são aleatórias, como a quantidade de clientes e a quantidade de produto para a entrega (PRV Estocástico);
- As entregas dos produtos em um determinado período (PRV Periódica).

A Figura 2.1 apresenta uma ilustração visual simples do funcionamento de um sistema de atendimento referente ao problema de roteamento de veículos.

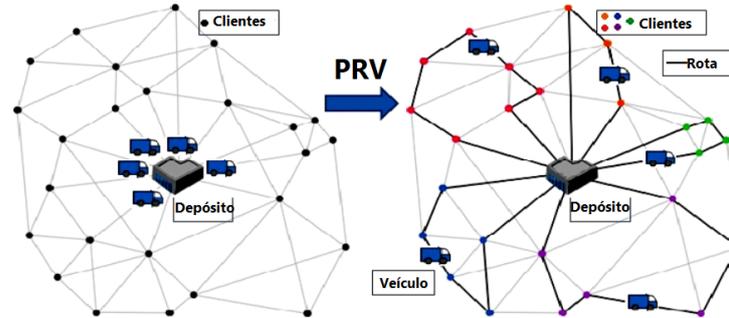


Figura 2.1: Ilustração de uma solução típica para o PRV. Fonte: figura adaptada de Zhou et al. (2018).

2.1 O Problema do Caixeiro Viajante como base do PRV

O problema do caixeiro viajante (PCV) pode ser descrito com base em grafos da seguinte forma. Seja $G = (V, A)$ um grafo completo, onde $V = \{0, 1, \dots, n\}$ é o conjunto de vértices e A é o conjunto de arestas. O conjunto de vértices $\bar{V} = V \setminus \{0\}$ corresponde aos vértices a ser visitados (clientes) e o vértice 0 representa a origem e destino (depósito). Cada aresta (i, j) está associado a um valor não negativo c_{ij} , que expressa o custo de viagem do vértice i ao vértice j . O uso da aresta (i, i) não é permitido, então $c_{ii} = \infty$ para todos $i \in V$. Se G for um grafo direcionado, a matriz de custos C é assimétrica (Toth & Vigo 2002, 2014, Golden et al. 2008). O objetivo do PCV é procurar uma rota que começa e termina em um determinado vértice denotado por 0, passando por todos os vértices apenas uma única vez, para minimizar a distância percorrida. A variável de decisão binária x_{ij} é definida para todos os $(i, j) \in A$, de modo que eles tomam o valor 1 se a aresta (i, j) é parte da solução, e 0 caso contrário.

Modelo matemático para o PCV

O seguinte modelo é apresentado no trabalho de Kulkarni & Bhawe (1985), onde n é o número de vértices e Z é a função objetivo que minimiza o custo total do percurso de todos os vértices.

$$\text{Minimizar } Z = \sum_{i \in V} \sum_{j \in V} c_{ij} x_{ij} \quad (2.1)$$

$$\text{Sujeito a } \sum_{i \in V} x_{ij} = 1 \quad j \in V, \quad (2.2)$$

$$\sum_{j \in V} x_{ij} = 1 \quad i \in V, \quad (2.3)$$

$$x_{ij} \in S. \quad (2.4)$$

$$x_{ij} \in \{0, 1\}, \quad (i, j) \in A \quad (2.5)$$

As restrições (2.2) indica que apenas uma aresta pode sair de cada vértice, enquanto as restrições (2.3) refere-se que uma aresta pode entrar em apenas um vértice. O conjunto S (conjunto de vértices a serem visitados) nas restrições (2.4) são selecionados para proibir soluções com sub-rotas desconectadas que satisfazem as restrições (2.2) e (2.3). No trabalho de [Kulkarni & Bhave \(1985\)](#) diferentes alternativas foram propostas para S . Algumas destas são:

1. $S = \{(x_{ij}) : \sum_{i \in B} \sum_{j \notin B} x_{ij} \geq 1 \text{ para cada subconjunto não vazio } B \subseteq V = \{0, 1, \dots, n\}\};$
2. $S = \{(x_{ij}) : \sum_{i \in B} \sum_{j \in B} x_{ij} \leq |B| - 1 \text{ para cada subconjunto não vazio } B \subseteq V \text{ tal que } 2 \leq |B| \leq n - 1\};$
3. $S = \{(x_{ij}) : y_i - y_j + nx_{ij} \leq n - 1 \text{ para } 1 \leq i \neq j \leq n - 1\}$, onde y_i é um número real arbitrário.

As alternativas 1 e 2 são de conectividade e de eliminação de subrotas de verificação direta, que têm a vantagem de ser restrições fortes do ponto de vista que tornam o problema linear relaxado mais apertado, porém têm a desvantagem de crescer exponencialmente em termos de quantidade em função do número de vértices. Por outro lado, a alternativa 3, proposta por [Miller et al. \(1960\)](#), que também elimina sobrotas, é conhecida por restrições do tipo MTZ. Essa tem a desvantagem de ser restrições fracas, no entanto tem a vantagem de ter quantidade polinomial.

Vamos verificar que a alternativa 3 é válida. Considere y_i indicando uma sequência em que o cliente i é visitado, e vamos considerar $i \neq 1$. Então temos que verificar $y_i - y_j + nx_{ij} \leq n - 1, \forall i, j \in \bar{V} \setminus \{1\}$ e $i \neq j$.

- Supondo que (r_1, r_2, \dots, r_j) seja ma subrota tal que $j < n$ e $r_i \neq 1, \forall i = 1, \dots, j$;
- Pela definição das variáveis, cada arco da subrota tem variável binária igual a 1, como segue:

$$x_{r_1 r_2} = x_{r_2 r_3} = \dots = x_{r_{j-1} r_j} = 1.$$

- Substituindo em $y_{r_i} - y_{r_{i+1}} + nx_{r_i r_{i+1}} \leq n - 1, i = 1, \dots, j - 1$, obtemos

$$y_{r_i} - y_{r_{i+1}} + n \leq n - 1,$$

$$y_{r_i} - y_{r_{i+1}} \leq -1, i = 1, \dots, j - 1.$$

- Somando-se todas essas desigualdades, para $i = 1, \dots, j - 1$, obtemos

$$y_{r_1} - y_{r_2} \leq -1,$$

$$y_{r_2} - y_{r_3} \leq -1,$$

$$\vdots$$

$$y_{r_{j-1}} - y_{r_j} \leq -1,$$

$$y_{r_j} - y_{r_1} \leq -1.$$

- A conclusão é a contradição $0 \leq -1$. Portanto, as restrições na alternativa 3 são válidas.

2.2 Elementos do Problema de Roteamento de Veículos

Geralmente existem três elementos envolvidos no PRV os quais são: os clientes, os depósitos e a frota de veículos ([Toth & Vigo 2002](#)).

2.2.1 Clientes

Os clientes são os que geram as demandas dos produtos que devem ser entregues por meio dos veículos desde os depósitos. Existe uma variante de clientes os quais podem ser fornecedores, eles querem que as suas mercadorias sejam coletadas e transportadas para o depósito. Os clientes podem ter restrições tais como: horário de serviço (expressado como janelas de tempo), compatibilidade entre veículos e clientes (no caso que um determinado cliente só possa ser atendido por algum veículo em particular), devoluções (quando existe a possibilidade que o cliente retorne mercadoria aos depósitos) e demanda imprevisível (se a quantidade demandada não é conhecida).

2.2.2 Depósitos

São os locais onde se encontram alocados os produtos a distribuir e os veículos que percorrem as rotas levando as mercadorias até os clientes. Pelo geral, cada rota começa e termina no mesmo depósito, mas em certos casos quando a frota não é própria os veículos não regressam ao depósito de origem, esta variável do PRV é conhecida como O-PRV (*Open-PRV*). Também existem problemas com múltiplos depósitos, as quais podem ter atribuída uma frota de veículos determinada. Os depósitos também podem contar com restrições associadas como janelas de tempo, com a capacidade, com a distância das rotas percorridas, dentre outras.

2.2.3 Veículos

São os meios de transporte utilizados para distribuir os produtos ou mercadorias desde os depósitos até os clientes, ou vice-versa. A capacidade pode estar expressa em peso, volume, número de clientes, entre outras expressões. Se os veículos são considerados com as mesmas características ou atributos considera-se frota homogênea, caso contrário, frota heterogênea. Os custos dos veículos contam com um componente fixo, este é incorporado no momento de ser utilizado num componente variável, a qual é proporcional à distância percorrida. Os veículos podem ter restrições relacionadas principalmente com a sua capacidade, horário de serviço (quando os veículos de carga têm restrições de mobilidade pelas vias em dias especiais) e locais de acesso (quando alguns veículos pesados não podem ingressar a certas localidades).

2.2.4 Modelo Clássico para o Problema de Roteamento de Veículos

Segundo [Toth & Vigo \(2002\)](#) o PRV simétrico é definido em um grafo não direcionado $G(V,A)$. Seja o conjunto de vértices $V = \{0, \dots, n\}$, onde a origem ou o depósito é localizado no vértice 0, apresenta-se também um conjunto \bar{V} de clientes onde $\bar{V} = V \setminus \{0\}$. Os vértices estão conectados através do conjunto de arestas A , cujos custos são determinados por c_{ij} (custo de viajar do vértice i até o vértice j), para $i \neq j$. Cada vértice, à exceção do depósito, requer uma demanda que necessita ser atendida por um conjunto K de veículos disponíveis para o transporte.

O PRV, portanto, visa a determinar o conjunto de rotas que minimizam o custo total de viagem respeitando as seguintes restrições:

- Cada cliente é visitado apenas uma vez;
- Todas as rotas iniciam e terminam na origem;

- A capacidade do veículo que percorre uma rota deve de ser maior ou igual à soma total das demandas dos clientes atendidos por ele.

Modelo matemático para o PRV clássico baseado no PCV segundo Toth & Vigo (2002).

$$\text{Minimizar} \quad \sum_{i \in V} \sum_{j \in V} c_{ij} x_{ij} \quad (2.6)$$

$$\text{Sujeito a} \quad \sum_{i \in V} x_{ij} = 1, \quad j \in \bar{V}, \quad (2.7)$$

$$\sum_{j \in V} x_{ij} = 1, \quad i \in \bar{V}, \quad (2.8)$$

$$\sum_{i \in V} x_{i0} = |K|, \quad (2.9)$$

$$\sum_{j \in V} x_{0j} = |K|, \quad (2.10)$$

$$\sum_{i \notin S} \sum_{j \in S} x_{ij} \geq r(S), \quad S \subset \bar{V}, \quad S \neq \emptyset, \quad (2.11)$$

$$x_{ij} \in \{0, 1\}, \quad (i, j) \in A. \quad (2.12)$$

A função objetivo (2.6) representa a somatório do custo das arestas permitidas representadas por x_{ij} . As restrições (2.7), (2.8) e (2.12) garantem que apenas duas arestas estão permitidas para cada vértice, uma para entrada e uma para saída. O conjunto de restrições (2.9) e (2.10) indica que cada veículo é usado, começando e terminando seu percurso no depósito. As restrições (2.11), visa determinar a conectividade do grafo na solução e o limite de capacidade do veículo.

O conjunto S é o conjunto dos clientes que pertencem a \bar{V} , dessa forma, $r(S)$ é número mínimo de veículos necessários para atender os clientes em S . Onde $r(S)$ pode ser substituído pelo somatório da demanda da instância de S dividida pela capacidade dos veículos. As restrições (2.11) pode ser reescrita, na sua formulação mais conhecida como a restrição de eliminação de sub rotas:

$$\sum_{i \in S} \sum_{j \in S} x_{ij} \leq |S| - r(S), \quad S \subseteq V \setminus \{0\}, \quad S \neq \emptyset. \quad (2.13)$$

Segundo Laporte (1984) a formulação acima é base para o desenvolvimento de outras forma de roteamento assim também para modelos baseado na Programação Linear Mista Inteira (PLMI). Para uma melhor abordagem do modelo referido sugere-se consultar Toth & Vigo (2002).

2.3 Variantes do Problema de Roteamento de Veículos

Diversas variantes para o Problema de Roteamento de Veículos foram desenvolvidos por muitos pesquisadores com o objetivo de projetar e dar soluções aos problemas de transporte, estas variações diferenciam-se por contar com novas variáveis assim também com diferentes restrições (Medaglia 2005). Na Figura 2.2 apresenta-se um esquema com as variantes do PRV mais estudadas.

2.3.1 Problema de Roteamento de Veículos Capacitado

O Problema de Roteamento de Veículos Capacitado (PRVC) é a variante mais comum do PRV. Esta inclui a restrição de capacidade dos veículos a qual não deve ser ultrapassada, isto é, a soma das demandas

dos clientes não deve ser maior que a capacidade Q do veículo que a percorre. Além disso, os clientes contam com uma demanda d_i por produtos específica. Em outras palavras, o PRVC é o problema onde um conjunto K de veículos idênticos, localizados em um armazém central percorrerem uma rota para atender a demanda dos clientes sujeito às restrições de capacidade dos veículos (Baldacci et al. 2007).

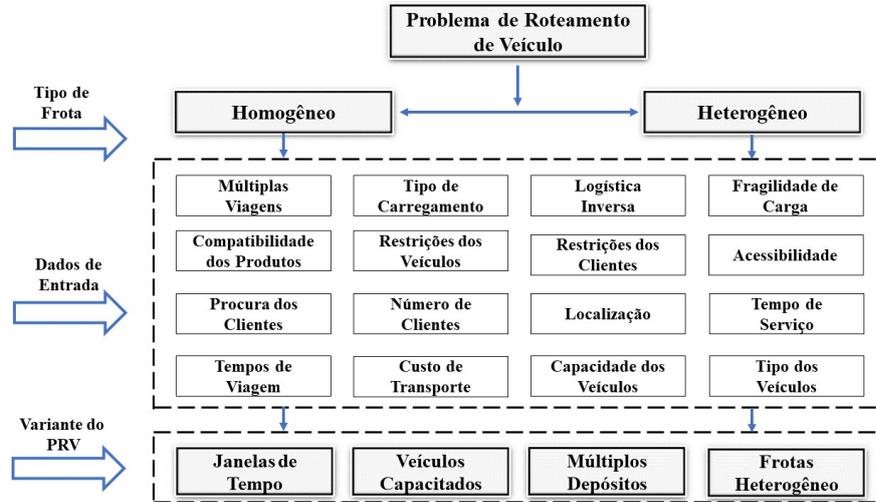


Figura 2.2: Variantes do PRV de acordo com as restrições. Fonte: figura adaptada de Eksioglu et al. (2009).

O objetivo desse problema é geralmente minimizar o custo total, onde este custo pode ser visto em função do tempo ou distância total percorrida, ou minimizar o número de veículos a serem utilizados. O PRVC é estudado nos primeiros capítulos dos livros de Toth & Vigo (2002, 2014).

Modelo matemático para o PRVC

Partindo do trabalho de Toth & Vigo (2002), descrevemos o PRVC. Considera-se um grafo $G = (V, A)$, onde $V = \{0, \dots, n\}$ representa o conjunto de vértices, o vértice 0 representa o depósito ou armazém, \bar{V} é definido como o conjunto de clientes onde $\bar{V} = V \setminus \{0\}$, e A representa o conjunto arestas (i, j) . O custo de transporte do vértice i até o vértice j é representado por c_{ij} para todo $i \neq j$. Cada cliente de \bar{V} possui uma demanda d_i não negativa, ($d_0 = 0$). Dos conjuntos de vértices $S \subseteq \bar{V}$, sendo S um conjunto de vértices (clientes), apresentamos a soma de todas as demandas dos clientes $d(S) = \sum_{i \in S} d_i$.

Seja K o conjunto de veículos idênticos disponíveis no depósito com capacidade Q , tal que, $d_i \leq Q$. Existe um único veículo associado a cada rota $k \in K$. Além disso, definimos uma variável associada a cada aresta do conjunto A , quer dizer, $x_{ijk} = 1$ se o veículo k percorre a aresta (i, j) e $x_{ijk} = 0$, caso contrário, para $i, j \in V, i \neq j$ e $k \in K$.

$$\text{Minimizar} \quad \sum_{i \in V} \sum_{j \in V} \sum_{k \in K} c_{ij} x_{ijk} \quad (2.14)$$

$$\text{Sujeito a} \quad \sum_{k \in K} \sum_{j \in V} x_{0jk} = |K|, \quad j \in \bar{V}, \quad (2.15)$$

$$\sum_{j \in V} x_{0jk} = \sum_{j \in V} x_{j0k} = 1, \quad k \in K, \quad j \in \bar{V}, \quad (2.16)$$

$$\sum_{k \in K} \sum_{j \in V} x_{ijk} = 1, \quad i \in \bar{V}, \quad (2.17)$$

$$\sum_{j \in V} x_{ijk} = \sum_{j \in V} x_{jik}, \quad k \in K, \quad i \in \bar{V}, \quad (2.18)$$

$$\sum_{k \in K} \sum_{i \in S} \sum_{j \in S} x_{ijk} \leq |S| - r(S), \quad S \subseteq \bar{V}, \quad S \neq \emptyset, \quad |S| \geq 2, \quad (2.19)$$

$$\sum_{i \in V} d_i \sum_{j \in V} x_{ijk} \leq Q, \quad k \in K, \quad i \in \bar{V}, \quad (2.20)$$

$$x_{ijk} \in \{0, 1\}, \quad i, j \in V, \quad k \in K. \quad (2.21)$$

A quantidade total de veículos presentes nos depósitos são descritas nas restrições (2.15), as restrições (2.16) prevê que os veículos partam de um depósito e voltem ao mesmo, as restrições (2.17) e (2.18) determinam que existe um veículo para uma determinada rota que percorre a aresta (i, j) . As restrições (2.19) são a restrição de eliminação de sub-rotas, o termo $r(S)$ representa o mínimo de veículos necessários para atender a demandas dos clientes em S . A capacidade de cada veículo é limitada em (2.20), a qual não pode ultrapassar a soma das demandas dos clientes para uma rota, e o último conjunto de restrições (2.21) indicam a integralidade das variáveis.

Extensões do PRVC

O PRVC conforme descrito na seção anterior, resolve uma grande variedade de problemas. No entanto, existem Problemas de Roteamento de Veículos Capacitados com características particulares que estão delimitados pelas diferentes restrições incorporadas ao PRV original, as quais são descritas em continuidade.

- Janelas de tempo. Através delas, cada veículo deve chegar a cada um dos seus clientes antes ou depois de uma data prevista. No caso de janelas de entrega flexíveis, paga-se uma penalidade caso não sejam respeitadas os tempos da entrega do produto;
- Frota Heterogênea (limitada ou ilimitada). Nem sempre toda a frota de transporte é homogênea, quer dizer, com a mesma capacidade Q e os mesmos custos variáveis. No caso da frota heterogênea, são veículos de portes diferentes e podem transportar variações dos carregamentos. A frota pode ainda estar limitada a uma quantidade de veículos pré-definida, produzindo uma possível solução com o uso delas. Outra extensão, geralmente relacionada a problemas de planejamento, não considera a limitação da quantidade de veículos. Essa abordagem é interessante quando o objetivo é definir uma política de atendimento, isto é, com qual veículo e com que frequência atende cada cliente. O resultado esperado é o dimensionamento da frota necessária que precisa ser contratada;
- Entregas divididas (*Split Deliverys*). Cada ponto da demanda tem a possibilidade de ser atendido por vários veículos. Além disso, a soma dos produtos demandados pelos clientes pode exceder a capacidade do veículo;
- Múltiplos Depósitos. Em estes problemas existem vários depósitos de origem onde os quais devem sair de um deles e retornar ao mesmo ou em outro depósito;
- Coleta e Entrega. É uma generalização do PRV, que trata de encontrar rotas ideais para atender às solicitações de transporte. Cada solicitação requer coleta e entrega com restrições de precedência, estes problemas contam com restrições espaciais e de sincronização de carga.

Problema de Roteamento de Veículos Capacitado segundo Kulkarni & Bhawe (1985)

Kulkarni & Bhawe (1985) descreveram que uma formulação simples para o PRVC é apresentada quando a restrição de capacidade e a restrição de custo máximo são iguais para todos os veículos. Seja o grafo $G = (V, A)$, onde $V = \{0, 1, \dots, n\}$ é o conjunto de vértices, 0 é o depósito e $\bar{V} = V \setminus \{0\}$ é o conjunto de clientes. Seja L número máximo de vértices (clientes) visitado por cada veículo, Q é a capacidade de todos os veículos e T é o custo máximo permitido para uma rota (em termo de distância ou tempo), K é o conjunto de veículos, d_i é a demanda dos clientes (com $d_0 = 0$). A variável binária x_{ijk} é igual a 1 se a aresta (i, j) está na rota k , 0 caso contrário, e y_i, u_i, v_i são números reais. Segue a formulação para um único depósito.

$$\text{Minimizar} \quad \sum_{i \in V} \sum_{j \in V} c_{ij} x_{ij} \quad (2.22)$$

$$\text{Sujeito a} \quad \sum_{i \in V} x_{ij} = 1, \quad \forall j \in \bar{V}, \quad (2.23)$$

$$\sum_{j \in V} x_{ij} = 1, \quad i \in \bar{V}, \quad (2.24)$$

$$\sum_{i \in V} x_{i0} = |K|, \quad (2.25)$$

$$\sum_{j \in V} x_{0j} = |K|, \quad (2.26)$$

$$y_i - y_j + Lx_{ij} \leq L - 1, \quad 1 \leq i \neq j \leq n - 1, \quad (2.27)$$

$$u_i - u_j + Qx_{ij} \leq Q - d_i, \quad 1 \leq i \neq j \leq n - 1, \quad (2.28)$$

$$v_i - v_j + Tx_{ij} \leq T - c_{ij}, \quad 1 \leq i \neq j \leq n - 1, \quad (2.29)$$

$$x_{ij} \in \{0, 1\}, \quad i, j \in V. \quad (2.30)$$

As restrições (2.23), (2.24), (2.25) e (2.26) garantem que todos os vértices (clientes) sejam visitados apenas uma vez e que todos os veículos são usados, saindo e retornando ao mesmo depósito. O conjunto de restrições (2.27), (2.28) e (2.29) são as restrições de eliminação de sub-rotas, onde as restrições (2.27) representam o número máximo de vértices visitados, as restrições (2.28) representam a capacidade do veículo em cada rota e as restrições (2.29) representa o custo máximo permitido para cada rota. O conjunto de restrições (2.27) são de eliminação de sub-rotas do tipo MTZ (Miller et al. 1960) e as restrições (2.28) e (2.29) são derivadas da restrição de eliminação das sub-rotas MTZ. Esta descrição para o problema será usada nos estudos computacionais apresentados no Capítulo 4.

2.3.2 Problema de Roteamento de Veículos com Janelas de Tempo

O Problema de Roteamento de Veículos com Janelas de Tempo (PRVJT) é uma extensão do problema de roteamento capacitado onde o atendimento da demanda dos clientes é realizada em uma determinada janela de tempo e o veículo deve permanecer na localização do cliente durante o atendimento. A tarefa é encontrar um conjunto de rotas com a menor distância, cada uma iniciando e terminando o trajeto em um depósito central, de modo que cada cliente seja visitado exatamente uma vez em uma determinada janela de tempo respeitando as capacidades dos veículos (Petersen et al. 2008). Seja S o conjunto de clientes a serem atendidos, então a janela de tempo para cada cliente $i \in S$ é representada pelo intervalo $[a_i, b_i]$. O veículo não pode chegar depois do horário determinado para o atendimento, no entanto, se ele chegar com antecedência, deve aguardar o início da janela de tempo. Assim, a restrição $a_i \leq t_{ik} \leq b_i$ deve ser incorporada

ao PRVJT, onde a variável de decisão t_{ik} é o tempo de chegada do veículo k no cliente i (Toth & Vigo 2002). Diversos pesquisadores estudaram este problema propondo métodos de resolução exatos e aproximados, entre eles, Gendreau et al. (1998), Bräysy & Gendreau (2002), e Sobh & Vacic (2004).

Modelo matemático para o PRVJT

Seja um grafo $G = (V, A)$, onde $V = \{1, \dots, n\} \cup \{0, 0'\}$ é o conjunto dos vértices, os vértices 0 e $0'$ denotam os depósitos início e fim, respectivamente, e $\bar{V} = V \setminus \{0, 0'\}$ é o conjunto de clientes. Cada cliente $i \in \bar{V}$ tem uma demanda d_i , enquanto $d_0 = d_{0'} = 0$. Além disso, s_i é o tempo de serviço associado a uma janela de tempo $[a_i, b_i]$, na qual o cliente i deve ser visitado. Adicionalmente, sejam $A = \{(i, j) : i, j \in V, i \neq j\}$ o conjunto de arestas e K o conjunto de veículos onde cada veículo conta com uma capacidade Q . Os conjuntos $\delta^+(i) = \{(i, j) : j \in V \setminus \{i\}\}$ e $\delta^-(i) = \{(j, i) : j \in V \setminus \{i\}\}$ contêm as arestas deixando e entrando no vértice i , respectivamente. A variável t_{ik} é o tempo de viagem para visitar o cliente i usando o veículo k . A variável de decisão binária x_{ijk} indica se o veículo k usa a aresta (i, j) no trajeto, neste caso o custo c_{ij} é adicionado. O tempo total do percurso na aresta (i, j) é representado por τ_{ij} , que depende do tempo de viagem da aresta e do tempo de atendimento ao cliente i . Golden et al. (2008) estudaram a seguinte formulação para o PRVJT.

$$\text{Minimizar} \quad \sum_{k \in K} \sum_{(i,j) \in A} c_{ij} x_{ijk} \quad (2.31)$$

$$\text{Sujeito a} \quad \sum_{k \in K} \sum_{(i,j) \in \delta^+(i)} x_{ijk} = 1, \quad i \in \bar{V}, \quad (2.32)$$

$$\sum_{(i,j) \in \delta^+(0)} x_{ijk} = \sum_{(i,j) \in \delta^-(0')} x_{ijk} = 1, \quad k \in K, \quad (2.33)$$

$$\sum_{(j,i) \in \delta^-(i)} x_{jik} - \sum_{(i,j) \in \delta^+(i)} x_{ijk} = 0, \quad i \in \bar{V}, \quad k \in K, \quad (2.34)$$

$$\sum_{(i,j) \in A} d_i x_{ijk} \leq Q, \quad k \in K, \quad (2.35)$$

$$x_{ijk}(t_{ik} + \tau_{ij}) \leq t_{jk}, \quad k \in K, \quad (i, j) \in A, \quad (2.36)$$

$$a_i \leq t_{ik} \leq b_i, \quad i \in \bar{V}, \quad k \in K, \quad (2.37)$$

$$x_{ijk} \in \{0, 1\}, \quad i, j \in V, \quad k \in K. \quad (2.38)$$

As restrições (2.32) obrigam que cada cliente i seja visitado e as restrições (2.33) garantem que cada veículo deve sair e retornar do depósito. As restrições (2.34) garantem o fluxo de conservação para cada veículo. Observa-se que o custo zero do arco $x_{0,0',k}$ entre o depósito inicial e final deve estar presente para todos os veículos, isto expressa um trajeto vazio caso um veículo não saia do depósito. As restrições (2.35) garantem que a capacidade de cada veículo não seja excedida e os conjuntos de restrições (2.36) e (2.37) indicam que as restrições da janela de tempo sejam satisfeitas. Observa-se que o conjunto de restrições (2.36), com a suposição de $t_{ij} > 0$ para todos os $i, j \in V$, eliminam todas as sub-rotas. As restrições (2.37) e (2.38) definem o domínio das variáveis.

2.3.3 Problema de Roteamento de Veículos com Coleta e Entrega

O Problema de Roteamento de Veículos com Coleta e Entrega (PRVCE) faz parte de uma família importante dos PRVs, onde os veículos transportam as mercadorias ou passageiros desde uma ou várias origens

para múltiplos destinos. Geralmente o PRVCE é definido sobre um grafo onde os vértices representam os destinos para os diferentes produtos a serem distribuídos. [Toth & Vigo \(2014\)](#) classificaram os PRVCEs em três categorias principais conforme o tipo de demanda e as estruturas das rotas.

- Problemas muitos-para-muitos (M-M): cada produto pode sair e chegar desde múltiplos pontos, e qualquer local pode ser a origem ou o destino de múltiplas mercadorias. Por exemplo, o reposicionamento de estoque entre lojas pequenas ou na gestão de sistemas de compartilhamento de carros;
- Problemas de um para muitos para um (1-M-1): são caracterizados pela presença de alguns produtos para ser entregues a partir de um depósito para muitos clientes e de outras mercadorias a serem coletadas nos clientes ou transportados de volta ao depósito. Esses problemas têm aplicações, por exemplo, na entrega de bebidas e na coleta de latas e garrafas vazias, e em sistemas da logística reversa onde também deve-se planejar a coleta de produtos usados, defeituosos ou obsoletos;
- Problemas um-para-um (1-1): cada mercadoria conta com uma única origem e destino para o seu traslado, com aplicações em transporte de cargas menores e nas operações de entrega via correios.

Modelo matemático para o PRVCE

O modelo é definido em um grafo completo $G = (V, A)$, onde $V = \{0, \dots, n\}$ é o conjunto dos vértices, $A = \{(i, j) : i, j \in V, i \neq j\}$ é o conjunto de arestas e K é o conjunto de veículos com capacidade Q . O vértice 0 é o depósito onde os veículos estão alocados, o conjunto $\bar{V} = V \setminus \{0\}$ representam os clientes e S é um subconjunto formado pelos clientes a serem atendidos onde $S \subseteq \bar{V}$. Cada cliente $i \in V$ conta com uma demanda d_i e a demanda para o depósito é representado por $d_0 = 0$. Se $d_i > 0$ significa que o cliente requer uma coleta e se $d_i < 0$ significa que o cliente requer uma entrega. Os veículos podem começar o seus percursos vazios ou com alguma carga. No trajeto da entrega dos produtos existe o custo de viajar pela aresta $(i, j) \in A$ representada por c_{ij} . O objetivo do PRVCE é procurar as rotas de veículos com a finalidade de minimizar os custos de transporte, cada veículo começa e termina o percurso da rota em um depósito, atendendo todas as demandas sem violar a sua capacidade Q ([Toth & Vigo 2002](#)).

O PRVCE pode ser modelado como um modelo de programação linear inteira mista usando dois conjunto de variáveis de decisão. A variável binária x_{ijk} assume valor 1 se a aresta (i, j) é percorrida por um dos veículos do conjunto K , caso contrário, x_{ijk} assume valor 0, e a variável não negativa f_{ij} é o fluxo de carga transportada na aresta (i, j) . [Toth & Vigo \(2002\)](#) formularam o PRVCE muitos-para-muitos como segue.

$$\text{Minimizar} \quad \sum_{i \in V} \sum_{j \in V} \sum_{k \in K} c_{ij} x_{ijk} \quad (2.39)$$

$$\text{Sujeito a} \quad \sum_{j \in V} \sum_{k \in K} x_{ijk} = 1, \quad i \in \bar{V}, \quad (2.40)$$

$$\sum_{j \in V} x_{ijk} - \sum_{j \in V} x_{jik} = 0, \quad i \in V, \quad k \in K, \quad (2.41)$$

$$0 \leq f_{ij} \leq Q \sum_{k \in K} x_{ijk}, \quad i, j \in V, \quad (2.42)$$

$$\sum_{j \in V} f_{ji} - \sum_{j \in V} f_{ij} = d_i, \quad i \in \bar{V}, \quad (2.43)$$

$$\sum_{i \in S} \sum_{j \in S} x_{ijk} \leq |S| - 1, \quad S \subseteq \bar{V}, \quad S \neq \emptyset, \quad k \in K, \quad (2.44)$$

$$x_{ijk} \in \{0, 1\}, \quad i, j \in V, \quad k \in K. \quad (2.45)$$

A função objetivo (2.39) representa o custo total das rotas a ser minimizado. Cada cliente é visitado uma única vez por (2.40). As restrições (2.41) indicam que o veículo comece e termine seu percurso em um depósito e não em outro ponto. As restrições (2.42) representam a quantidade máxima da demanda em cada rota a qual pode ser menor ou igual a capacidade Q do veículo. As restrições (2.43) definem a conservação do fluxo. As restrições (2.44) evitam a formação de sub-rotas na solução, e (2.45) indica a domínio das variáveis.

2.3.4 Problema de Roteamento de Veículos Periódico

O Problema de Roteamento de Veículos Periódico (PRVP) foi estudado inicialmente por [Beltrami & Bodin \(1974\)](#) e posteriormente um número significativo de trabalhos surgiram com diversas variantes e aplicações. Por exemplo, [Francis et al. \(2005, 2008\)](#) formularam um método de resolução mediante a relaxação lagrangiana para o PRVP, onde as rotas devem ser construídas sobre vários dias (unidade de tempo). A cada dia no período de planejamento, uma frota de veículos capacitados percorre um conjunto de rotas. O objetivo do PRVP é procurar um conjunto de rotas com custo total de viagem mínimo, enquanto satisfaz as restrições de capacidade dos veículos e atendimento dos clientes. Esse último exige visitas repetidas durante o horizonte de planejamento. Assumindo planejamento semanal, a escolha dos dias para visitar cada cliente pode seguir um padrão. Por exemplo, se for factível, (2-4) e (3-5) podem expressar duas visitas semanal aos clientes, na terça e quinta-feira e quarta e sexta-feira, respectivamente. As restrições clássicas do PRV devem ser respeitadas (início e fim de cada rota no depósito, capacidade dos veículos e duração da rota). [Toth & Vigo \(2002\)](#) descreveram as seguintes três principais decisões para o PRVP: i) a seleção dos padrões de visitas de cada cliente; ii) as atribuições das combinações de dia-cliente para as visitas; e iii) o percurso dos veículos para os dias de despacho das mercadorias.

Modelo matemático para o PVRP

Sejam D o conjunto de dias, onde $d \in D$ constitui o planejamento periódico, e \bar{S} o conjunto de todos os cronogramas de atendimento. Cada cronograma $s \in \bar{S}$ está descrito por uma variável binária a_{sd} , onde $a_{sd} = 1$ se o dia $d \in D$ está dentro da programação $s \in \bar{S}$, e $a_{sd} = 0$ caso contrário. O PRVP é descrito sobre um grafo completo $G = (V, A)$ com custos de viagem conhecidos, onde $V = \{0, \dots, n\}$ é o conjunto de vértices, sendo 0 o depósito, e $A = \{(i, j) : i, j \in V, i \neq j\}$ é o conjunto de arestas. O custo da viagem pela aresta (i, j) é dado por c_{ij} . Um conjunto de clientes em $\bar{V} = V \setminus \{0\}$ com demanda total W_i no período de planejamento d exige um número fixo de visitas f_i dentro do conjunto de cronogramas de atendimento \bar{S} . Essas visitas são realizadas por um conjunto de veículos K , cada um deles com capacidade Q . O objetivo é achar uma designação de clientes de modo que estes sejam visitados um número necessário de vezes por dia, ou seja, mais de uma vez no mesmo dia. Nos problemas periódicos deve-se escolher um cronograma de atendimento \bar{S}_i não vazio do conjunto de todos os programas de atendimento \bar{S} , a qual pode ser expressado por $\bar{S}_i \subseteq \bar{S}$, tal que, $\bar{S}_i = \{s \in \bar{S} : \sum_{d \in D} a_{sd} = f_i\}$.

[Christofides & Beasley \(1984\)](#) estudaram a seguinte formulação para o PVRP, onde L representa a distância máxima para o trajeto do veículo e tem os três conjuntos de variáveis de decisão.

$x_{ijkd} = 1$, se o veículo $k \in K$ visita a aresta $(i, j) \in A$ no dia $d \in D$, e $x_{ijkd} = 0$, caso contrário;

$z_{is} = \sum_{k \in K} y_{iks} = 1$, se o vértice $i \in \bar{V}$ é visitado no horário $s \in \bar{S}$, e $z_{is} = 0$, caso contrário;

$v_{id} = 1$, se o vértice $i \in \bar{V}$ é visitado no dia $d \in D$, e $v_{id} = 0$, caso contrário.

$$\text{Minimizar} \quad \sum_{d \in D} \sum_{(i,j) \in A} \sum_{k \in K} c_{ij} x_{ijkd} \quad (2.46)$$

$$\text{Sujeito a} \quad \sum_{s \in \bar{S}} z_{is} = 1, \quad i \in \bar{V}, \quad (2.47)$$

$$v_{id} = \sum_{s \in \bar{S}} z_{is} a_{sd}, \quad d \in D, \quad i \in \bar{V}, \quad (2.48)$$

$$\sum_{k \in K} x_{ijkd} \leq \frac{v_{id} + v_{jd}}{2}, \quad d \in D, \quad i, j \in \bar{V} (i \neq j), \quad (2.49)$$

$$\sum_{j \in \bar{V}} x_{ijkd} = \sum_{j \in \bar{V}} x_{jikd}, \quad i \in V, \quad k \in K, \quad d \in D, \quad (2.50)$$

$$\sum_{k \in K} \sum_{i \in \bar{V}} x_{ijkd} = \begin{cases} v_{jd}, & j \in \bar{V}, \\ |K|, & j = 0, \end{cases} \quad d \in D, \quad (2.51)$$

$$\sum_{j \in P} \sum_{i \in P} x_{ijkd} \leq |P| - 1, \quad P \subseteq \bar{V}, \quad k \in K, \quad d \in D, \quad (2.52)$$

$$\sum_{j \in \bar{V}} x_{0jkd} \leq 1, \quad k \in K, \quad d \in D, \quad (2.53)$$

$$\sum_{i \in \bar{V}} w_i \sum_{j \in V} x_{ijkd} \leq Q, \quad k \in K, \quad d \in D, \quad (2.54)$$

$$\sum_{(i,j) \in A} c_{ij} x_{ijkd} \leq L, \quad k \in K, \quad d \in D, \quad (2.55)$$

$$z_{is}, v_{id} \in \{0, 1\}, \quad i \in \bar{V}, \quad s \in \bar{S}, \quad d \in D, \quad (2.56)$$

$$x_{ijkd} \in \{0, 1\}, \quad (i, j) \in A, \quad k \in K, \quad d \in D. \quad (2.57)$$

A função objetivo (2.46) representa o custo total das arestas selecionadas a ser minimizado. O conjunto de restrições (2.47) garantem a escolha de um cronograma viável de visitas para cada vértice, enquanto (2.48) define os dias v_{id} de visita no cronograma atribuído. As restrições (2.49) permitem a escolha de arestas apenas entre clientes atribuídos para entrega no dia $d \in D$, enquanto (2.50) conserva o fluxo. As restrições (2.51) permitem que os vértices sejam incluídos nas rotas por dias dentro da programação atribuída. As restrições (2.52) são as de eliminação de sub rotas. Um veículo não é usado mais de uma vez por dia por (2.53). As restrições (2.54) e (2.55) são as restrições de capacidade do veículo e da distância máxima da rota, respectivamente. Finalmente, as restrições (2.56) e (2.57) definem o domínio das variáveis de decisão.

2.3.5 Problema de Roteamento de Veículos com Entrega Dividida

Dror & Trudeau (1989) mostraram que com o uso do Problema de Roteamento de Veículos com Entrega Dividida (PRVED) pode ocorrer economia nas entregas das mercadorias, porém da mesma forma que as outras variantes do PRV ele continua sendo um problema computacionalmente complexo de resolver. O problema com entregas dividida envolve uma frota de veículo de capacidade homogênea, que está disponível para visitar um conjunto de clientes mais de uma vez se precisar. Além disso, a demanda pode ser maior que a capacidade do veículo e o veículo precisa iniciar e terminar sua viagem no mesmo depósito. O objetivo do PRVED consiste em desenhar rotas para os veículos de forma a minimizar os custos de transporte e a distância

total percorrida, da mesma forma atender toda a demanda, onde a soma destas não ultrapasse a capacidade do veículo em cada percurso (Archetti & Speranza 2008). Estes problemas aplicam-se principalmente quando a demanda de alguns clientes é maior que a capacidade de um veículo, neste caso o cliente pode ser atendido por mais de um veículo quando o custo total de entrega é reduzido (Belenguer et al. 2000).

Modelo matemático para o PRVED

A formulação baseada na programação inteira mista para o PRVED pode ser definida como um grafo $G = (V, A)$, onde o conjunto de vértices é representado por $V = \{0, \dots, n\}$, sendo 0 o depósito e os outros vértices representam os clientes os quais estão no conjunto $\bar{V} = V \setminus \{0\}$. O conjunto de arestas é representado por $A = \{(i, j) : i, j \in V, i \neq j\}$. O custo do percurso na aresta $(i, j) \in A$ é representada por c_{ij} . Uma demanda inteira d_i está associada a cada cliente $i \in \bar{V}$, considerando $d_0 = 0$ para o depósito. Um conjunto de veículos K , cada veículo com capacidade Q , está disponível para atender um subconjunto de clientes $S \subseteq V$, estes veículos precisam começar e termina o seu percurso no mesmo depósito. O objetivo deste problema consiste em procurar um conjunto de rotas para minimizar a distância total percorrida pelos veículos, considerando que a demanda não pode exceder a capacidade do veículo. Golden et al. (2008) formularam o PRVED com dois seguinte conjuntos de variáveis:

x_{ijk} toma o valor de 1, se o veículo $k \in K$ viaja diretamente de i para j , e 0 caso contrário;

y_{ik} é a quantidade da demanda do vértice i entregue pelo veículo $k \in K$.

$$\text{Minimizar} \quad \sum_{i \in V} \sum_{j \in V} \sum_{k \in K} c_{ij} x_{ijk} \quad (2.58)$$

$$\text{Sujeito a} \quad \sum_{i \in V} \sum_{k \in K} x_{ijk} \geq 1, \quad j \in V, \quad (2.59)$$

$$\sum_{i \in V} x_{ipk} - \sum_{j \in V} x_{pjk} = 0, \quad p = 0, \dots, n, \quad k \in K, \quad (2.60)$$

$$\sum_{i \in S} \sum_{j \in S} x_{ijk} \leq |S| - 1, \quad k \in K, \quad S \subseteq \bar{V}, \quad (2.61)$$

$$y_{ik} \leq d_i \sum_{j \in V} x_{ijk}, \quad i \in V, \quad k \in K, \quad (2.62)$$

$$\sum_{k \in K} y_{ik} = d_i, \quad i \in \bar{V}, \quad (2.63)$$

$$\sum_{i \in V} y_{ik} \leq Q, \quad k \in K, \quad (2.64)$$

$$x_{ijk} \in \{0, 1\}, \quad i, j \in V, \quad k \in K, \quad (2.65)$$

$$y_{ik} \geq 0, \quad i \in V, \quad k \in K. \quad (2.66)$$

As restrições (2.59) indicam a visita do veículo em cada vértice pelo menos uma vez. As restrições (2.60) descrevem os fluxos de conservações. o conjunto de restrição (2.61) são as de eliminação de sub-rotas. As restrições (2.62) e (2.64) representam as distribuições das demandas dos clientes entre os veículos. As restrições (2.62) impõem que a demanda do cliente i seja atendido pelo veículo k somente se este passa pelo cliente i . As restrições (2.63) garantem que toda a demanda de cada vértice seja satisfeito. As restrições (2.64) indicam que a soma das demandas é menor do que a capacidade do veículo. As duas últimas restrições (2.65) e (2.66) fornecem o domínio das variáveis.

2.3.6 Problema de Roteamento de Veículos com Demanda Estocástica

O Problema de Roteamento de Veículos com Demanda Estocástica (PRVDE) é uma variação do problema capacitado onde a demanda do cliente é incerta, embora seja estatisticamente conhecida mediante os históricos das informações de despacho disponíveis, evitando que os veículos ultrapassem a capacidade de carga. Uma abordagem mais profunda é descrita por [Secomandi \(2000\)](#). O PRVDE surge como consequência dos problemas nas entregas das mercadorias para um grupo de clientes cujas demandas não são conhecidas. Para esta variante do problema de roteamento, assume-se a demanda do cliente como uma variável aleatória discreta, quer dizer, a probabilidade do cliente em pedir uma quantidade de produtos. Assume-se também que a demanda torna-se conhecida quando o veículo chega ao local do cliente ([Bertsimas 1992](#)).

Modelo matemático do Fluxo de Rede para o PRVDE

Segundo [Toth & Vigo \(2014\)](#) o problema pode ser representado em um grafo não direcionado completo $G = (V, A)$, onde $V = \{0, 1, \dots, n\}$ é o conjunto de vértices e $A = \{(i, j) : i, j \in V, i < j\}$ é o conjunto de arestas. O vértice 0 é o depósito, onde estão alocados um conjunto K de veículos idênticos de capacidade Q , enquanto os vértices restantes representam os clientes. Cada cliente $i \in \bar{V} = V \setminus \{0\}$ conta com uma demanda estocástica não negativa ξ_i para ser coletado. Além disso, assumimos que essas demandas são variáveis aleatórias independentes com distribuições conhecidas com valores esperados μ_i e também se apresenta um custo de viagem c_{ij} associado a cada aresta (i, j) .

$$\text{Minimizar} \quad \sum_{i < j} c_{ij} x_{ij} + C(x) \quad (2.67)$$

$$\text{Sujeito a} \quad \sum_{j \in \bar{V}} x_{0j} = 2|K|, \quad (2.68)$$

$$\sum_{i < \ell} x_{i\ell} + \sum_{j > \ell} x_{\ell j} = 2, \quad \ell \in \bar{V}, \quad (2.69)$$

$$\sum_{i, j \in S} x_{ij} \leq |S| - \left| \sum_{i \in S} \mu_i / Q \right|, \quad S \subset \bar{V}, \quad 3 \leq |S| \leq n-1, \quad (2.70)$$

$$0 \leq x_{ij} \leq 1, \quad i \in \bar{V}, \quad j \in \bar{V}, \quad i < j, \quad (2.71)$$

$$0 \leq x_{0j} \leq 2, \quad j \in \bar{V}, \quad (2.72)$$

$$x = x_{ij} \in \mathbb{Z}_+, \quad i \in \bar{V}, \quad j \in \bar{V}, \quad i < j. \quad (2.73)$$

A formulação do fluxo de rede para o PRVDE apresenta uma variável de decisão inteira $x_{ij} (i < j)$ igual ao número de vezes que (i, j) aparece na solução no primeiro estágio da rota. Esta variável deve ser interpretada como x_{ji} sempre que $i > j$. Se $i, j > 0$, então x_{ij} só pode assumir os valores 0 ou 1; se $i = 0$, então x_{ij} também pode ser igual a 2, representando uma situação quando um veículo faz uma viagem de volta entre o depósito e o vértice j . Seja $C(x)$ o custo do recurso esperado da solução x , onde este custo é dependente da sua orientação e pode ser expressada como: $C(x) = \sum_{k=1}^{|K|} \min \{C^{k,1}, C^{k,2}\}$, nesta formulação C^δ representa o custo de recurso esperado pela rota k para a orientação δ . Em uma solução de primeiro estágio não direcionada fornecida pela fórmula anterior, o custo de recurso esperado de cada rota é calculado para cada direção onde a orientação com menor custo é selecionada.

O cálculo de $C^{k,1}$ para a rota k é definido por $(i_1 = 0, i_2, \dots, i_{t+1} = 0)$, assumindo que $\xi_i \leq Q$ pode contar com probabilidade 1 para todo i , a qual é expresso por $C^{k,1} = 2 \sum_{j=2}^t \sum_{l=1}^{j-1} P \left(\sum_{s=2}^{j-1} \xi_{i_s} \leq lQ < \sum_{s=2}^j \xi_{i_s} \right) c_{0i_j}$, onde l representa a probabilidade do número do erro no cliente i_j e c_{0i_s} é o custo de viagem de retorno ao depósito.

A função objetivo (2.67) consiste em minimizar os custos de viagem e o custo de recurso esperado. Neste modelo, as restrições (2.68) e (2.69) especificam o estágio de cada vértice, enquanto as restrições (2.70) eliminam as sub-rotas e garantem que a demanda esperada de qualquer rota não exceda a capacidade do veículo.

2.3.7 Problema de Roteamento de Veículos com *Backhauls*

No Problema de Roteamento de Veículo com *Backhauls* (PRVB) os clientes contam com a possibilidade de retornar alguns produtos. Portanto é necessário determinar a quantidade de mercadorias para uma devolução, incluindo a capacidade de um veículo transportá-la. Um estudo para este problema foi desenvolvido por [Anbuudayasankar et al. \(2012\)](#), que apresentaram soluções através de algoritmos genético.

Nos PRV com *Backhauls*, os clientes são divididos em dois conjuntos, os clientes *Linehaul* e *Blackhaul*. Os clientes *Linehaul* exigem a entrega de mercadorias do depósito, e os clientes *Backhaul* precisam a devolução das mercadorias para o depósito. Uma reorganização da carga dentro de um veículo com frequência é impossível ou indesejável porque o espaço de carga do veículo só é acessível desde a parte traseira, porém é recomendável estabelecer uma relação de precedência no serviço de clientes *Linehaul* e *Backhaul* ([Toth & Vigo 2003](#)). Uma forma de dar solução pode ser usando o serviço combinado de *Linehaul* e *Backhaul* na mesma rota, esta é possível sempre que todos os clientes *Linehaul* recebem sua demanda antes do carregamento dos produtos dos clientes *Backhaul*. Este problema pertence à classe de entrega frequente e está incluída na variante dos problemas de roteamento mais estudadas, melhorando de forma considerável os serviços de distribuição e coleta ([Golden et al. 2008](#)).

Modelo matemático para o PRV com *Backhauls*

O problema pode ser formulado através do seguinte modelo em grafos, onde os vértices representam os clientes e o depósito. Seja $G = (V, A)$ um grafo não direcionado completo com conjunto de vértices $V = \{0\} \cup L \cup B$, os subconjuntos $L = \{1, \dots, n\}$ e $B = \{n+1, \dots, n+m\}$ correspondem aos clientes de *Linehaul* e *Backhaul*, respectivamente. Uma quantidade não negativa d_i de produto a ser entregue ou coletado (demanda) está associada a cada vértice $i \in V \setminus \{0\}$. O vértice 0 corresponde ao depósito (com uma demanda fictícia $d_0 = 0$) e K é o conjunto de veículos de capacidade homogênea Q alocadas no depósito, onde $|K| \geq r(S)$. Seja c_{ij} o custo não negativo associado à aresta $(i, j) \in A$, com $c_{ii} = +\infty$ para cada $i \in V$ e $c_{ij} = c_{ji}$ para cada $i, j \in V$ de forma que $i \neq j$ ([Toth & Vigo 1997, 2002](#)). O PRVB procura os custos mínimos das rotas percorridas pelos $|K|$ veículos, de modo que:

- Cada veículo visite o vértice 0;
- Cada vértice $i \in V \setminus \{0\}$ seja visitado exatamente uma vez;
- A soma das demandas dos clientes de *Linehaul* e *Backhaul* visitados por um veículo não deve exceder a sua capacidade Q ;

- Em cada rota os clientes de *Linehaul* precedem aos clientes de *Backhaul*, se houver.

Segundo o trabalho de [Toth & Vigo \(1997\)](#), este problema pode ser apresentado de forma linear onde se define o conjunto $L_0 = L \cup \{0\}$ e $B_0 = B \cup \{0\}$. Apresenta-se também um grafo $G' = (V', A')$ produzida de G onde definimos $V' = V$ e $A' = A_1 \cup A_2 \cup A_3$. O conjunto de arestas A' pode ser dividido em três sub conjuntos separados, $A_1 = \{(i, j) \in A : i \in L_0, j \in L\}$, $A_2 = \{(i, j) \in A : i \in B, j \in B_0\}$ e $A_3 = \{(i, j) \in A : i \in L, j \in B_0\}$. O primeiro abrange todas as arestas desde o depósito até os vértices dos clientes de forma consecutiva, o segundo conta com todos os vértices de retorno desde os clientes até o depósito e o terceiro integra as chamadas “arestas de interfaces”, conectando os vértices de saída para o conjunto de vértices de retorno ou para o depósito. Observe que A' não inclui arcos que não pode pertencer a uma solução viável, além disso é um sub conjunto próprio de A . Cada aresta $(i, j) \in A'$ dispõe um custo c'_{ij} igual ao custo correspondente a aresta $(i, j) \in A$, quer dizer $c'_{ij} = c_{ij}$.

Seja \mathcal{L} (respectivo \mathcal{B}) todos os subconjuntos dos vértices em L (respectivo B) onde $\mathcal{F} = \mathcal{L} \cup \mathcal{B}$. Para cada subconjunto de clientes $S \in \mathcal{F}$ se tem um menor número de veículos $r(S)$ de capacidade Q para atender a demanda. Em cada $i \in V$ definimos $\Delta_i^+ = \{j : (i, j) \in A'\}$ e $\Delta_i^- = \{j : (j, i) \in A'\}$. [Toth & Vigo \(1997\)](#) descreveram o seguinte modelo matemático para o PRV com *Backhauls*.

$$\text{Minimizar} \quad \sum_{(i,j) \in A'} c_{ij} x_{ij} \quad (2.74)$$

$$\text{Sujeito a} \quad \sum_{i \in \Delta_j^-} x_{ij} = 1, \quad j \in V' \setminus \{0\}, \quad (2.75)$$

$$\sum_{j \in \Delta_i^+} x_{ij} = 1, \quad i \in V' \setminus \{0\}, \quad (2.76)$$

$$\sum_{i \in \Delta_0^-} x_{i0} = |K|, \quad (2.77)$$

$$\sum_{j \in \Delta_0^+} x_{0j} = |K|, \quad (2.78)$$

$$\sum_{j \in S} \sum_{i \in \Delta_j^- \setminus S} x_{ij} \geq r(S), \quad S \in \mathcal{F}, \quad (2.79)$$

$$\sum_{i \in S} \sum_{j \in \Delta_i^+ \setminus S} x_{ij} \geq r(S), \quad S \in \mathcal{F}, \quad (2.80)$$

$$x_{ij} \in \{0, 1\}, \quad (i, j) \in A'. \quad (2.81)$$

A variável x_{ij} pode assumir o valor de 1 se, e somente se, a aresta $(i, j) \in A'$ está no percurso do veículo. As restrições (2.75) e (2.76), e (2.77) e (2.78) impõem as restrições de entrada e saída para os clientes e os depósitos, respectivamente. As chamadas restrições de redução de capacidade, (2.79) e (2.80), impõem tanto a conectividade quanto ao limite de capacidade. Devido às restrições (2.75) e (2.78), para qualquer S , as restrições (2.79) e (2.80) são iguais, quer dizer, o número de arcos que entram em S é igual o número de arcos que saem dele. Portanto se as restrições (2.79) são impostas, então as restrições (2.80) são redundante e vice-versa. Alternativamente, um modelo equivalente para o PRVB é formulado introduzindo as restrições (2.79) apenas para cada $S \in \mathcal{L}$ e (2.80) para cada $S \in \mathcal{B}$. Note-se também que o valor de $r(S)$ nas restrições (2.79) e (2.80) pode estar substituído por qualquer limite inferior na solução ideal.

2.3.8 Problema de Roteamento de Veículos com Múltiplos Depósitos

O Problema de Roteamento de Veículo com Múltiplos Depósitos (PRVMD) é caracterizada por possuir mais de um depósito que atendam os clientes (Montoya-Torres et al. 2015). Quando os clientes estão agrupados entorno dos depósitos, o problema de distribuição pode ser modelado como um Sistema de Roteamento de Veículos Independente. No entanto, se os clientes e depósitos forem misturados o problema deve ser resolvido como um sistema de Roteamento de Veículos com Vários Depósitos (Ho et al. 2008). Para ilustrar a diferença de um problema de roteamento e o PRVMD podemos observar a Figura 2.3.

Nestes problemas os veículos podem sair de qualquer depósito, transportar as mercadorias e voltar ao mesmo depósito de saída. O tamanho da frota deve variar entre um mínimo e um máximo especificado. O PRVMD é de difícil solução, portanto métodos baseados na Programação Linear dificilmente são adequados para a obtenção de soluções, porém o desenvolvimento de algoritmos heurísticos ou meta-heurísticos são fundamentais para resolvê-lo (Tansini et al. 2001).

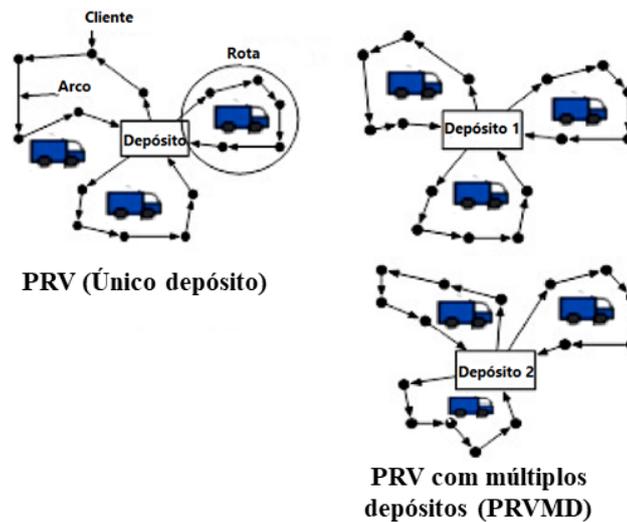


Figura 2.3: Ilustração de uma configuração para o PRVMD. Fonte: figura adaptada de Ho et al. (2008).

Como existe mais de um depósito, torna-se difícil determinar o atendimento dos clientes pelos depósitos respeitando as limitações de capacidade dos veículos, portanto o agrupamento de clientes (*clustering*) é uma forma de resolvê-lo. Geralmente as técnicas de agrupamento consideram a distância entre os clientes e os depósitos como uma característica de similaridade para a procura de uma solução. O objetivo do problema é encontrar rotas para veículos com custo mínimo em relação ao número de veículos e distância total da viagem, sem violar as limitações de capacidade e tempo de viagem dos veículos.

Utilizar algoritmos heurístico de duas fases é uma das formas de resolver este problema. Na primeira fase da heurística os clientes são designados aos depósitos formando os *clusters* e, na segunda fase, rotas de custo mínimo devem ser identificadas para cada *cluster* (Geetha et al. 2012).

Modelo Matemático para o PRVMD estudado por Kulkarni & Bhawe (1985)

O PRVMD pode ser descrito como segue. Seja um grafo $G = (V, A)$, onde $V = \{1, 2, \dots, n + m\}$ é o conjunto de vértices e A é ainda o conjunto de arestas conectando cada par de vértices (i, j) . V é dividido em dois subconjuntos. $\bar{V} = \{1, 2, \dots, n\}$ é o conjunto de clientes e $V_d = \{n + 1, n + 2, \dots, m\}$ é o conjunto

de depósitos, com $V = \bar{V} \cup V_d$. Cada vértice $i \in \bar{V}$ conta com uma demanda não negativa d_i . Cada aresta $(i, j) \in A$ está associada a um custo representado por c_{ij} . Sejam L o número máximo de vértices (clientes) visitado por um veículo, K o conjunto de veículos com capacidade homogênea Q , e T o custo máximo permitido para uma rota (em termos de distância ou tempo). Segundo [Renaud et al. \(1996\)](#), o problema pode ser resumido em determinar um conjunto de rotas para os $|K|$ veículos tal que:

- Os veículos devem começar e terminar em um mesmo depósito;
- Cada cliente deve ser atendido uma única vez no percurso de um veículo;
- Para cada rota, a demanda total dos clientes não deve exceder a capacidade do veículo;
- Seja minimizado o custo total das rotas de transporte.

Apresenta-se o modelo matemático para o PRVMD proposto por [Kulkarni & Bhave \(1985\)](#). Este modelo foi implementado e resolvido no estudo de caso apresentado no Capítulo 4. Ele foi usado para obter soluções para o problema de distribuição de produtos na rede logística de uma empresa.

$$\text{Minimizar} \quad \sum_{i \in V} \sum_{j \in V} c_{ij} x_{ij} \quad (2.82)$$

$$\text{Sujeito a} \quad \sum_{i \in V} x_{ij} = 1, \quad j \in \bar{V}, \quad (2.83)$$

$$\sum_{j \in V} x_{ij} = 1, \quad i \in \bar{V}, \quad (2.84)$$

$$\sum_{i \in V_d} \sum_{j \in V} x_{ij} = |K|, \quad (2.85)$$

$$\sum_{j \in V_d} \sum_{i \in V} x_{ij} = |K|, \quad (2.86)$$

$$y_i - y_j + Lx_{ij} \leq L - 1, \quad 1 \leq i \neq j \leq n, \quad (2.87)$$

$$u_i - u_j + Qx_{ij} \leq Q - d_i, \quad 1 \leq i \neq j \leq n, \quad (2.88)$$

$$v_i - v_j + Tx_{ij} \leq T - c_{ij}, \quad 1 \leq i \neq j \leq n, \quad (2.89)$$

$$x_{ij} \in \{0, 1\}, \quad i, j \in V. \quad (2.90)$$

A variável binária x_{ijk} é igual a 1 se a aresta (i, j) é percorrida pelo veículo $k \in K$, e y_i, u_i, v_i são variáveis contínuas. As restrições (2.83), (2.84), (2.85) e (2.86) garantem que todos os vértices sejam visitados apenas uma vez e que todos os $|K|$ veículos sejam usados, onde o percurso precisa começar e terminar no depósito. O conjunto de restrições (2.87), (2.88) e (2.89) eliminam as sub-rotas desconectadas, isto é, as restrições (2.87) indica o número máximo de vértices visitados, as restrições (2.88) expressa a capacidade do veículo em cada rota e as restrições (2.89) indica o custo máximo permitido em cada rota.

Métodos de Resolução

Este capítulo apresentará algumas metodologias importantes desenvolvidas ao longo do tempo por diversos autores e que foram adequadamente direcionadas a uma ampla classe de problemas de roteamento. A Figura 3.1 ilustra a partir de um organograma um resumo das diversas abordagens de resolução, as quais podem ser em síntese separadas em metodologias para obtenção de soluções exatas e aproximadas. Primeiramente os algoritmos clássicos para a busca de soluções exatas são apresentados e, em seguida, aparecem as heurísticas e meta-heurísticas mais conhecidas na literatura que, separadas ou combinadas entre si, configuram como métodos potentes para obtenção de soluções aproximadas, principalmente para exemplares do PRV relativamente grandes em relação aos dados de entrada.

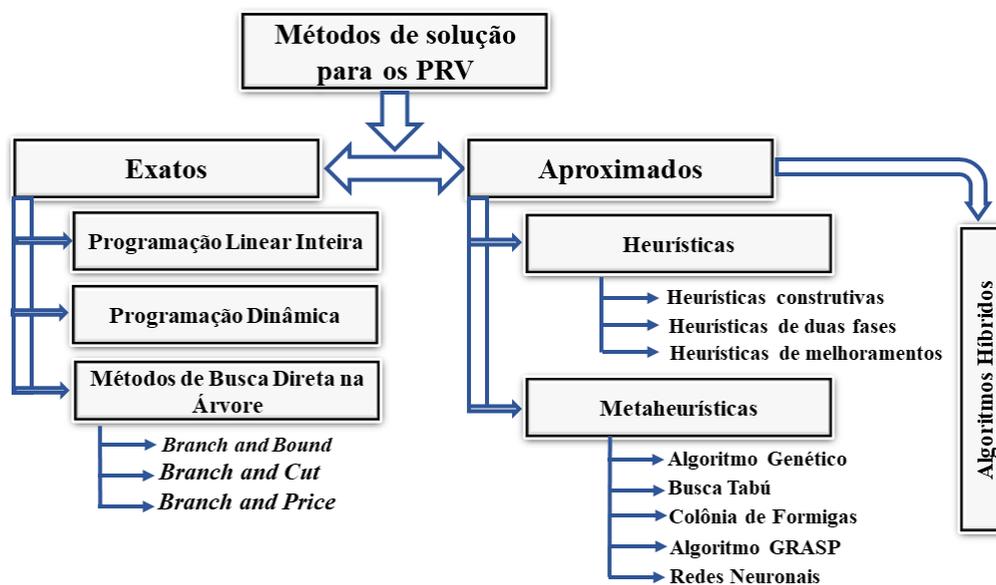


Figura 3.1: Resumo das metodologias para resolução de problemas de roteamento de veículos

Encontrar uma boa solução ou com sucesso encontrar a solução ótima para exemplares do PRV em muitas oportunidades é uma tarefa difícil, principalmente devido ao alto custo computacional envolvido na resolução de exemplares dimensionalmente grandes. Sabe-se que o custo computacional para executar metodologias exatas de resolução cresce de forma exponencial em relação a dimensão do problema e, mesmo com o uso de algoritmos aproximados, torna-se importante desenvolver uma boa arquitetura de implementa-

ção da linguagem computacional, uma vez que pode ultrapassar a capacidade da memória de computadores pessoais potentes. Em particular, neste trabalho foi utilizado via Solver comercial de otimização a versão default do algoritmo exato *Branch and Bound* em exemplares considerados pequenos (50, 100, 150 e 200 vértices). Outros métodos exatos que aparecem na literatura para resolver o PRV incluem a programação dinâmica e técnicas para a programação linear e inteira (Lüer et al. 2009).

Na atualidade, as técnicas para resolver as múltiplas variantes do PRV diferem entre si dependendo da classe de algoritmos a que pertencem, isto é, se eles são baseados por exemplo em programação linear, ou se pertencem as técnicas heurísticas ou meta-heurísticas (Potvin 1996). Devido à complexidade dos PRVs foram ainda propostas diferentes formatos de algoritmos de otimização combinatória com configurações híbridas para fornecer soluções de boa qualidade, próximas da solução ótima e, em alguns casos, com tempo computacional aceitável. Segundo Jourdan et al. (2009), os algoritmos híbridos são métodos usados para problemas específicos, portanto, são conhecidos do tipo *ad-hoc*, estes combinam aspectos de diferentes algoritmos exatos, heurísticos e/ou meta-heurísticos.

Conforme referido nos parágrafos anteriores, as possibilidades de implementação de um método ou de outro variam dependendo das características específicas do problema, da dimensão do mesmo e das características consideradas no processo de tomada de decisão. Com base numa revisão bibliográfica das principais pesquisas sobre o PRV, constatou-se que as técnicas mais utilizadas são as meta-heurísticas (80% aproximadamente dos casos nas últimas duas décadas), em segundo lugar estão as heurísticas (15%) e, finalmente, os métodos exatos e outros (5%).

3.1 Métodos Exatos

Os métodos exatos de resolução são algoritmos capazes de resolver de forma ótima os PRV e as suas variantes. A maioria destas variantes geralmente precisam de um tempo computacional elevado para o seu desenvolvimento na obtenção de uma solução eficaz (este tempo aumenta de forma exponencial segundo tamanho da entrada do problema (Mateos et al. 2013). Laporte & Nobert (1987) apresentaram uma classificação dos métodos exatos, onde definem três grandes grupos.

- Busca direta em árvore;
- Programação Dinâmica;
- Programação Linear Inteira.

Para essas formulações, a resolução se concentra em métodos de ramificação e limitação com geração de coluna ou combinando métodos de restrições válidas, ramificação e corte, além disso, soluções viáveis podem ser obtidas estabelecendo limites inferiores e superiores.

3.1.1 Programação Linear Inteira

Dentro dos métodos exatos para resolver os PRV contém a Programação Linear Inteira, onde a ideia é considerar um conjunto de rotas factíveis e um coeficiente binário que é igual a 1 se, e somente se, determinado depósito pertence a uma rota. O custo ótimo de uma rota também é levado em consideração e

a variável binária é igual a 1 se, e somente se, essa rota for usada na solução ótima (Desrochers et al. 1992, Desrosiers et al. 1984).

Na Programação Linear Inteira estão classificadas três técnicas descritas a seguir (Laporte 1992):

1. Conjunto de partições e geração de colunas. Proposta por Balinski & Quandt (1964), esta técnica consiste em resolver o problema linear aplicando o Método Simplex, no entanto, dado o grande número de colunas para cada veículo, em vez de avaliar todas as colunas em relação ao seu custo reduzido, é melhor determinar qual delas deve ser adicionada à base, com o objetivo de procurar a coluna com o maior custo reduzido;
2. Formulação de fluxo de veículos de dois índices: Proposta por Fisher & Jaikumar (1978), aqui, a ideia de usar as variáveis binárias $x_{i,j}$ foram introduzidas para determinar se a aresta (i, j) é usada ou não na solução;
3. Fórmula de fluxo de veículos de três índices: Proposta por Fisher & Jaikumar (1978), esta é usada em formulações baseadas em índice, as quais incorporam 2 ou 3 índices ao programa linear original.

3.1.2 Programação Dinâmica

A Programação Dinâmica foi proposta por Eilon et al. (1974), sendo considerado um método algorítmico poderoso, onde um problema é resolvido identificando uma coleção de sub problemas e lidando com eles um por um. Os problemas menores são tratados primeiro e seus resultados são usados para resolver os maiores, até que o problema original seja resolvido (Cooper & Cooper 2016, Brimberg & Love 1998).

3.1.3 Métodos de Busca em Árvore

Nestes métodos a solução é realizada em todos os vértices de uma árvore seguindo critérios específicos (Laporte 1992). Estes são amplamente usados e resolveram com sucesso alguns problemas combinatórios tão difíceis quanto os PRV (Held & Karp 1971, Hansen & Krarup 1974, Crowder & Padberg 1980). É preciso dar destaque para algumas das técnicas incluídas para este método, como são aos algoritmos *Branch and Bound* (B&B), *Branch and Cut* e *Branch and Price*, os quais funcionam sobre o conceito de criar uma árvore de solução e “podar” ramos sobre um critério estabelecido até alcançar uma solução.

Método de ramificação e limitação (*Branch and Bound*)

O método B&B, proposto por Little et al. (1963), é considerado um método de enumeração onde se definem limites superiores e inferiores para achar a solução inteira de cada ramificação ou *branch*. O termo *bound*, por sua vez, ressalta a prova de otimalidade a qual determina-se comparando os limitantes achados ao longo da enumeração. A enumeração implícita, trata-se de reduzir o espaço de busca descartando sub conjuntos de soluções que se formam e que não pertencem a solução ótima. Os sub conjuntos formados são excluídos utilizando os limites superior e inferior para o valor da solução achada até esse momento. O método B&B então, parte da ideia de formar possíveis aspirantes à solução ótima inteira de um problema.

O primeiro a aplicar um método exato em um problema de transporte baseado no algoritmo B&B foi Gavish & Graves (1978) para o PCV, anos posteriores Laporte et al. (1986) aplicaram a metodologia de

ramificação aos PRVC. O próximo a abordar e resolver um PRVC mediante B&B foram [Fischetti et al. \(1994\)](#) onde se considerou uma matriz de custo assimétrico. No mesmo ano, [Fisher \(1994\)](#) resolveu o problema com instâncias de 100 clientes, usando K problemas de árvore de expansão mínima. Posteriormente, aplicou-se de forma similar nos estudos de [Miller \(1995\)](#) e [Hadjiconstantinou et al. \(1995\)](#).

Desenvolvimento do algoritmo *Branch and Bound*

Segundo o algoritmo, os limitantes (inferiores e superiores) são obtidos resolvendo as relaxações lineares dos vértices (PL^i). Considera-se que os vértices ativos, que estão armazenados em uma lista B , são aqueles vértices que não foram eliminados por nenhuma regra de eliminação. Um exemplo de regra de eliminação de vértices é a regra a *last-in first-out*, está consiste em validar o último vértice que entrar na lista é ser o primeiro ao sair da mesma. Segundo a revisão bibliográfica, a procura da solução ótima é mais fácil em níveis mais profundos da árvore. Outra regra de eliminação para os problemas de maximização é resolver primeiro os vértices com maior limitante e para problemas de minimização podemos escolher vértices com o menor limitante, está regra produz uma árvore menos profunda e maiores vértices ativas para resolver.

O foco do algoritmo B&B é ignorar as variáveis inteiras e resolver o problema como se fosse um problema de programação linear. O algoritmo inicia partindo de uma iteração inicial, se nesta iteração se consegue resultados inteiros então o problema está resolvido, de modo que a solução obtida com a PL é a solução ótima ([Toth & Vigo 2002](#)). Se a solução não é inteira, teremos que continuar a busca usando essa variável não inteira x_i e delimitando as cotas inteiras inferiores e superiores, as quais são os inteiros mais próximos de x_i , onde x_{menor} e x_{maior} são o menor e o maior número inteiro de x_i respectivamente. O B&B forma, então, duas novas restrições que se incorporam uma a uma ao problema original obtendo-se dois novos problemas, essa obtenção desses novos problemas se conhece como ramificação. Logo, procede-se ao buscar uma solução inteira em cada um de esses novos problemas, como se fosse a iteração inicial.

O B&B utiliza para sua análise no caso de problemas de maximização uma cota inferior (CI), para a qual o algoritmo apenas examina os problemas com solução fornecida pelo PL que estejam acima da CI. Uma forma normal é começar com uma CI para os valores de x_i , não obstante, existem procedimentos de B&B que esperam a obter uma CI igual à primeira solução com valores inteiros. B&B muda seu CI quando existe um problema com uma solução inteira cujo valor da função objetiva é maior que a CI atual.

Algoritmo de ramificação e corte (*Branch and Cut*)

Este algoritmo, proposto por [Gomory](#) em 1958, é considerado como aquele que fornece soluções ótimas com maior exatidão para o PRV. O algoritmo usa a relaxação linear do modelo de Programação Linear Inteira (PLI), quer dizer, para problema de PLI eles relaxam com a premissa de que todas as variáveis são inteiras. No capítulo 3 de [Toth & Vigo \(2002\)](#) é explicado em detalhes. O algoritmo de *Branch and cut* (B&C) pode ser definida como a modificação no plano de corte (restrições adicionais que buscam se aproximar à convexidade da região factível de um problema inteiro) para melhorar a qualidade da relaxação linear do algoritmo B&B ([Balas et al. 1996](#)). A ideia consiste em gerar desigualdades válidas ao longo de toda a árvore B&B e não só no vértice raiz. As desigualdades são escolhidas de um conjunto pré-selecionado, em cada vértice deve-se escolher entre melhorar a cota inferior ou processar o vértice de forma acelerada. Geralmente não se formariam desigualdades válidas em um vértice se a melhora da cota inferior

fosse pequena após um certo número de iterações. Nestas condições, é melhor ramificar e gerar maiores quantidades de desigualdades nos vértices filhos.

Um estudo para este algoritmo foi feito por [Laporte et al. \(1985\)](#), onde usaram a relaxação linear para o PRVC sem as restrições de capacidade e de distância máxima. Uma vantagem deste algoritmo consiste na capacidade de resolver instâncias euclidianas e não euclidianas geradas aleatoriamente, usando dois ou três veículos para um máximo de 60 clientes. Uma abordagem completa sobre o algoritmo *Branch and Cut* no PRVC foi feita no trabalho de [Augerat et al. \(1995\)](#), eles propuseram etapas da heurísticas para separar qualquer desigualdade válida. Usar essas abordagens podem fornecer soluções para várias instâncias no PRVC com até 314 clientes. Note-se, então, que os algoritmos de B&C foram usados satisfatoriamente na otimização inteira, um caso notável é a solução do PCV dado por [Applegate et al. \(2003\)](#) e outras estudadas por [Karaoglan et al. \(2012\)](#).

Algoritmo *Branch and Price*

O algoritmo *Branch-and-Price* (B&P) utiliza a técnica de geração de colunas em cada nó de uma árvore de busca *branch-and-bound* para obtenção de novas variáveis não-básicas ([Barnhart et al. 1998](#)). Segundo [Boyd & Mattingley \(2007\)](#), a técnica de geração de colunas realiza uma analogia entre os conceitos de variáveis do problema e colunas, sendo muito útil quando nos deparamos com um problema que possui um número de variáveis muito maior que o número de restrições. Segundo [Jünger & Thienel \(1998\)](#), B&P é um conceito ligado ao algoritmo B&C e pode ser aplicado na Programação Inteira nas quais contam com grandes números de variáveis, portanto, os algoritmos B&P segue um esquema B&B com a diferença de como este resolve a Programação Linear. Como em todos os procedimentos de ramificação na árvore de busca e geração de colunas, o processo de ramificação se executa quando o programa linear não proporciona uma solução ótima inteira. No entanto, alguns autores descrevem ([Barnhart et al. 1998](#); [Vance et al. 1997](#); [Savelsbergh 1997](#)) que não é apropriado aplicar um *Branch and Bound* “normal” sobre as colunas existentes já que podem inferir na geração delas, então é conveniente desenhar um conjunto de regras especiais de ramificação. A regra de ramificação deve ser compatível com o problema *pricing*, que deve conseguir modificar o sub problema de forma que as colunas que são ineficazes devido às restrições de ramificação não sejam geradas e o sub problema de geração de colunas permaneçam tratáveis ([Savelsbergh 1997](#)).

Desenho do algoritmo *Branch and Price*

O esquema de *Branch and Price* baseia-se em dois conceitos. O primeiro, consiste na decomposição do problema, transformando o problema original desde sua formulação compacta em uma formulação que contém muitas colunas e menos filas que a formulação original. Essa transformação denomina-se formulação extensa (*extensive formulation*). O segundo conceito é a geração de colunas. Para poder resolver essa nova formulação extensa não se sugere desenvolver o modelo completo, pois este têm uma quantidade grande de variáveis (a quantidade de variáveis geralmente é exponencial ao tamanho do problema), porém se propõe gerar as colunas dinamicamente. Quando a cota inferior é calculada dinamicamente usando geração de colunas dentro de um esquema de *Branch and Bound*, o algoritmo é chamado *Branch and Price*. Acrescentar uma busca de *Branch and Bound* sobre um problema linear tratado com geração de colunas pode parecer bastante simples, mas tem suas dificuldades.

Algoritmo Concorde

O Concorde é o algoritmo mais eficaz de procura de soluções para o PCV. Este algoritmo usa uma estratégia de busca de ramificação e limitação, assim também o plano de corte de Dantzig na redução do espaço de busca. Além disso, usa técnicas de programação linear inteligentes para obter e verificar uma solução ótima. Em geral, o processo de verificação leva mais tempo do que a obtenção da solução (Sanches et al. 2017, Applegate et al. 2011).

3.2 Heurísticas

Métodos aproximados de resolução

São aqueles métodos que partem de uma formulação similar aos modelos de Programação Linear Inteira dos quais acham uma solução factível inteira a partir de técnicas de melhoramento dentro de um conjunto de soluções factíveis – heurísticas e meta-heurísticas (Toth & Vigo 2002). Devido ao carácter combinatório, um número grande de heurísticas foi proposto para o PRV, começando pelos algoritmos de construção de rotas, porém mais tarde, os métodos meta-heurísticas foram desenvolvidas gerando melhor solução do que as heurísticas. A seguir, revisamos separadamente essas duas famílias de algoritmos das quais foram desenvolvidas, descritas e testadas para o PRV simétrico nos últimos anos por diferentes pesquisadores. Além disso, quase todos estes métodos pressupõem um número ilimitado de veículos disponíveis. No entanto, deve-se observar que muitos deles podem adaptar-se com maior facilidade ao considerar restrições práticas adicionais (Toth & Vigo 1996).

Heurísticas para o PRV

As heurísticas são procedimentos que exploram o espaço de busca de forma limitada, gerando soluções aceitáveis em curto tempo de execução (Reeves 1993). Além do curto tempo computacional, estes algoritmos modelam o problema de forma precisa, da mesma forma são mais flexíveis e aptas dentro das operações com a função objetivo e/ou restrições mais complexas comparado com os algoritmos exatos.

Toth & Vigo (2002) classificaram as heurísticas para o PRV da seguinte forma:

- Heurísticas construtivas;
- Heurísticas de duas fases;
- Heurísticas de melhoramentos.

3.2.1 Heurísticas Construtivas

Estas heurísticas tentam construir uma solução seguindo algumas regras construtivas, criando gradualmente uma solução factível, centrando a atenção em reduzir o custo final, embora não contêm uma fase de melhora, ou seja, a solução que se obtém é a solução factível e deverá ser aplicada mesmo esta não seja ótima (Toth & Vigo 2002). Segundo Dorigo & Stützle (2019), as heurísticas construtivas servem para construir uma solução de um problema de otimização etapa por etapa. Iniciam-se a partir de uma solução vazia e, a cada etapa é adicionado um novo componente da solução até que seja completada.

Os elementos a serem inseridos na solução pode ser aleatório ou guloso, quando a heurística de construção trabalha de maneira aleatória, a cada iteração, um elemento entre os elementos ainda não inseridos na solução, é escolhido aleatoriamente e adicionado à solução. No caso de uma heurística de construção gulosa, quando é feita a adição de um novo elemento na solução que está sendo construída, é selecionado o elemento dentre os ainda não inseridos que proporciona maior benefício ao valor da função objetivo. Alguns dos métodos heurísticos de construção são:

Método heurístico das economias de *Clarke and Wright (C&W)*

O método construtivo mais implementado para o PRV de grande porte, consiste no método das economias ou poupanças proposta por [Clarke & Wright \(1964\)](#), este algoritmo é muito conhecido e usado dentro dos PRV na formação das rotas. Ele é utilizado quando o número de veículos é uma variável de decisão, gerando soluções para os problemas direcionados e os não direcionados. A lógica do algoritmo é aproveitar a economia que pode se formar quando se juntam duas rotas existentes. ([Osman 1993](#)).

O algoritmo da economia funciona da seguinte forma: em primeiro lugar, é atribuído um veículo para cada cliente, criando n rotas do tipo (v_0, \dots, v_i, v_0) , onde v_0 é o depósito central. Depois tenta combinar duas rotas (v_0, \dots, v_i, v_0) e (v_0, \dots, v_j, v_0) onde podem ser juntadas de forma viável em uma rota só $(v_0, \dots, v_i, v_j, \dots, v_0)$, desde que a capacidade do veículo permita. Para determinar a ordem de combinação das rotas, aproveita-se a economia gerada pela união de dois clientes de duas rotas distintas. A referida poupança, é calculada por $s_{ij} = c_{i0} + c_{0j} - c_{ij}$ ([Toth & Vigo 2002](#)).

Este algoritmo possui duas abordagens, as versões paralela e sequencial. A abordagem paralela cria rotas simultaneamente, enquanto a abordagem sequencial cria uma rota de cada vez. O método das economias segue os seguintes passos:

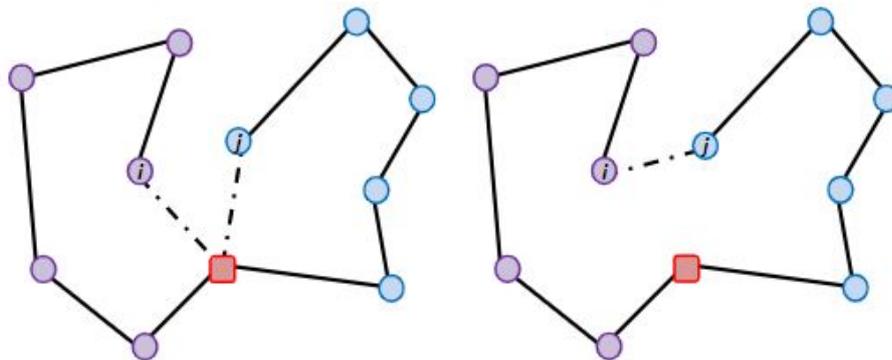


Figura 3.2: Representação da heurística C&W. Fonte: figura adaptada de [Clarke & Wright \(1964\)](#).

1. Calcular a matriz das poupanças s_{ij} . Determina-se a poupança s_{ij} se a rota $(v_0, \dots, v_i, v_j, \dots, v_0)$ fosse construído, para todos os pares de vértices i e j a qual resulta em uma nova rota $(v_0, \dots, v_i, v_j, \dots, v_0)$ ao invés de atender i e j em duas rotas (v_0, \dots, v_i, v_0) e (v_0, \dots, v_j, v_0) separadas como se apresenta na [Figura 3.2](#). Então temos que $s_{ij} = c_{i0} - c_{ij} + c_{0j}$;
2. Ordenar de forma decrescente as poupanças s_{ij} ;

3. Determinar o máximo valor não usado de s_{ij} . Se juntam as rotas (v_0, \dots, v_i, v_0) e (v_0, \dots, v_j) considerando as restrições da viabilidade de roteamento, da capacidade do veículo e da restrição de tempo máximo da viagem.

Existe duas variações deste método:

Versão Paralela. Na versão paralela do algoritmo C&W, a combinação de rotas que oferece a maior economia é sempre escolhida e implementada, obtida dos segmentos (v_0, v_j) e (v_i, v_0) e formando a nova rota (v_i, v_j) , como se explicou nos parágrafos anteriores. A versão paralela fornece resultados de menor custo do que a versão sequencial (Cordeau et al. 2002).

4. Se ao incluir $(v_0, \dots, v_i, v_j, \dots, v_0)$ à rota e formar-se uma melhor rota factível que respeita as restrições do PRV, então anexar essa rota à solução, caso contrário, excluí-la;
5. Tentar com a próxima rota da lista e repetir o Passo 4 até que todas as rotas sejam incluídas, dessa forma resolver o problema com a menor poupança nos custos.

Versão Sequencial. Considere, por sua vez, cada primeira rota $(v_0, \dots, v_i, v_j, \dots, v_0)$ determinada e salva com a poupança s_{ij} que pode ser usado de forma viável para a rota atual, junto com outra rota que contém a aresta $(i, 0)$ ou a aresta $(0, i)$. O próximo passo é introduzir a combinação das rotas e repetir esta operação para a rota atual. Se não houver combinação viável, considere a próxima rota e faça as mesmas operações. Pode parar a operação quando nenhuma combinação de rota for viável (Toth & Vigo 2002).

Existe uma grande diferença nos resultados numéricos relatados para as heurísticas das poupanças, diversas vezes, os autores não mencionam se a versão paralela ou sequencial é considerada nos resultados. No trabalho de Christofides (1979) fizeram os testes, usando distâncias reais, com a finalidade de conhecer qual versão é melhor. Os resultados obtidos indicaram que a versão paralela do método das poupanças, domina sobre a sequencial. Devido ao mais recente aprimoramento do C&W para resolver o PRV, mencionado na literatura acima, muitos autores propuseram novos parâmetros aplicados na busca das poupanças. Gaskell (1967) e Yellow (1970) apresentaram o parâmetro λ de formato de saída que controla a significância relativa do arco entre dois clientes. Sua fórmula para determinar a poupança é $s_{i,j} = c_{0,i} + c_{j,0} - \lambda c_{i,j}$.

Na nova formulação, λ é um parâmetro que penaliza a união de rotas com clientes distantes (denominado parâmetro de forma), este parâmetro também pode ser usado para gerar um conjunto de soluções diferentes, executando repetidamente o algoritmo com diferentes valores de λ . Em geral, no início da execução do algoritmo, duas alternativas podem parecer equivalentes, mas a escolha de uma delas gera um grande impacto na solução final. As soluções obtidas com o algoritmo de C&W, em geral, podem ser melhoradas usando operadores de busca locais como o algoritmo 3-opt (Lin 1965).

Algoritmo LKH

O algoritmo Lin–Kernighan–Helsgaun pertence a classe de algoritmos de busca local (Lin & Kernighan 1973). O algoritmo é especificado em termos de trocas (movimentos entre arcos) que podem transformar uma rota em outra. Dado uma rota viável, o algoritmo realiza trocas repetidamente reduzindo a duração da rota atual até que seja alcançada uma rota onde nenhuma troca produza uma melhoria. Este processo pode ser repetido muitas vezes, a partir de uma rota inicial gerada de forma aleatória (Helsgaun 2000).

Segundo [Helsgaun \(2000\)](#), o LKH parte do conceito do algoritmo Lin–Kernighan (LK), neste algoritmo procura-se achar dois conjuntos de *links* X e Y , onde $X = x_1, \dots, x_r$ e $Y = y_1, \dots, y_r$. Se os *links* de X forem excluídos da rota T , (T é a rota inicial), e substituído pelos *links* de Y , então a rota T inicial será melhorada. Essa troca de *links* pode ser chamada de movimento *r-opt*. A Figura 3.3 ilustra a troca para um $r = 3$.

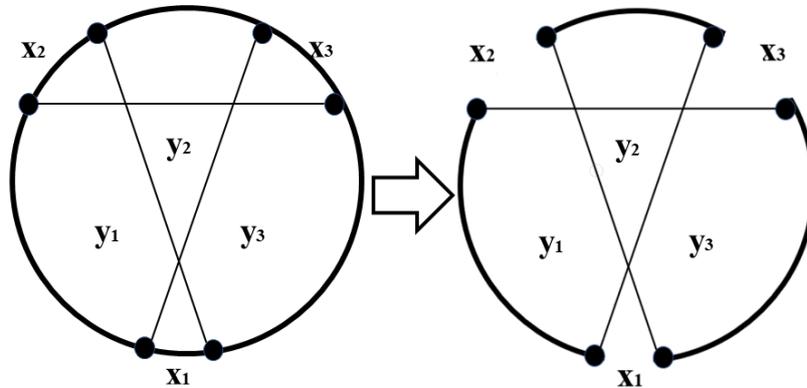


Figura 3.3: Representação do algoritmo LK. Fonte: figura adaptada de [Helsgaun \(2000\)](#).

Os passos para a funcionalidade do algoritmo LKH baseado no algoritmo LK

- Passo 1. Escolher de forma aleatória uma rota inicial T .
- Passo 2. Selecionar um vértice aleatório i .
- Passo 3. Escolher um arco $x_1 = (i, j)$ de forma que o arco $x_1 \in T$.
- Passo 4. Escolher um arco $y_1 = (i, j)$ que compartilhe um vértice com x_1 de forma que $y_1 \notin T$. Se não é possível achar um y_i , volte para o Passo 3 para a escolha de outro arco x_i , se ainda não é possível achar um y_i , voltar ao Passo 2 para a escolha de outro vértice.
- Passo 5. Selecionar um arco $x_2 = (i, j)$ que compartilhe um vértice com y_1 de forma que $x_2 \in T$.
- Passo 6. Escolher de forma aleatória $x_3 = (i, j)$ da rota T , onde $x_3 \neq x_1$ e $x_3 \neq x_2$.
- Passo 7. Escolher y_2 e y_3 de forma que ao substituir x_1, x_2 e x_3 por y_1, y_2 e y_3 a rota resultante seja Hamiltoniana.
- Passo 8. Se alguma troca encontrada no Passo 7 produzir uma distância inferior ao inicial, a substituição é feita e volte para o Passo 1, caso contrario, volta para o Passo 6.
- Passo 9. Se foram testados todos os x_3 possíveis, volte para o Passo 4 selecionando um arco diferente y_1 . Se já foram testados com todas os arcos y_1 , então uma solução foi encontrada.

Método heurístico de inserção

Dentro das heurísticas de inserção encontram-se dois algoritmos cada um deles é desenvolvida em duas fases, que resolvem os problemas com um número de veículos não específicos ([Toth & Vigo 2002](#), [Gendreau et al. 1998](#)). Essas heurísticas criam soluções mediante sucessivas inserções de clientes nas rotas, isto é, em cada iteração tem-se uma solução parcial cujo as rotas só visitam um sub conjunto dos clientes e depois selecionam um cliente não visitado para ser introduzido na última rota criada.

Descreve-se dois algoritmos representativos que foram baseados em inserções sequenciais. Ambos são aplicados a problemas com número de veículo não específico. O primeiro foi desenvolvido por [Mole & Jameson \(1976\)](#), onde expande-se uma rota ao mesmo tempo. O segundo formulado por [Christofides et al. \(1981\)](#), na qual gera-se rota sequencial e paralela com procedimentos construtivos. Ambos métodos contêm fases de melhoria do tipo *3-opt* ([Toth & Vigo 2002](#)).

Heurística de inserção sequencial de Mole e Jameson

A heurística de [Mole & Jameson](#) (M&J) baseia-se na inserção de clientes na rota atual até que ela seja concluída e uma nova rota possa ser iniciada. Esta heurística utiliza duas medidas para decidir o cliente a ser inserido na rota parcial. Para todos os clientes que ainda não foram visitados, calcula-se a melhor posição para colocar este cliente na rota parcial, considerando as distâncias, as exigências e sem reordenar os vértices que estão atualmente na rota.

O algoritmo usa os parâmetros λ e μ para modificar a seleção dos clientes. Se fizermos crescer o parâmetro λ então favorecemos a introdução de clientes entre os vértices mais distantes e, aumentando o valor de μ privilegia-se a inserção do cliente longe do depósito. Em geral, os últimos clientes não visitados estão distantes uns dos outros, portanto as últimas rotas construídas são de baixa qualidade. Seja (v_0, \dots, v_i, v_0) uma rota inicial onde v_0 é o depósito. Selecionando w um vértice ainda não visitado, o custo $c_1(v_i, w)$ para inserir w na rota (v_1, \dots, v_0) é determinado pela seguinte regra. Se $(v_0, \dots, v_i, w, v_{i+1}, \dots, v_0)$ for factível, então calcula-se $c_1(v_i, w) = c_{v_i, w} + c_{w, v_{i+1}} - \lambda c_{v_i, v_{i+1}}$. Caso contrário, toma-se $c_1(v_i, w) = \infty$. A equação $i(w) = \arg \min_{i=0, \dots, t} c_1(v_i, w)$ é usada para alocar o vértice w na melhor posição na rota atual.

Se apenas c_1 fosse usada para decidir o próximo cliente a inserir na rota, é provável que os clientes distantes ao depósito não sejam considerados até chegar às iterações finais do algoritmo, quer dizer, quando são as únicas alternativas viáveis. Portanto, é necessário utilizar um incentivo adicional para a inserção de clientes distantes ao depósito, este incentivo é definido por $c_2(v_i, w) = \mu c_{0w} - c_1(v_i, w)$ para cada cliente w .

Além das medidas anteriores, deve-se considerar a viabilidade das inserções. Quando nenhuma inserção for viável e se ainda houver clientes não visitados, um cliente é selecionado para iniciar uma nova rota. Podemos aplicar o algoritmo seguindo os próximos passos:

1. Inicialização da rota. Se todos os clientes pertencerem a alguma rota, encerre. Caso contrário, selecione um cliente não visitado e crie a rota $r = (v_0, v_w, v_0)$;
2. Inserção do vértice. Seja $r = (v_0, v_1, \dots, v_0)$ onde $v_0 = 0$, que representa o depósito. Para cada cliente não visitado w , calcula-se $i(w) = \arg \min_{i=0, \dots, n} c_1(v_i, w)$. Se não existe interação factível voltar para o Passo 1. Logo, calcular $w^* = \arg \max_w c_2(v_{i(w)}, w)$ e inserir w^* , depois $v_{i(w^*)}$ em r ;
3. Otimização da rota. Aplicar o algoritmo *3-opt* de [Lin \(1965\)](#) para r e volte para o Passo 2.

Heurística de inserção sequencial de Christofides, Mingozzi e Toth

Considerado uma heurística de duas fases, onde é usado os parâmetros λ e μ . Estes parâmetros permitem que se altere o comportamento do algoritmo de diversas formas, conforme consta em ([Goldberg & Luna 2005](#)). Na primeira fase determina-se a quantidade de rotas a utilizar aplicando a primeira fase da heurística

M&J, para obter rotas compactas e conservar os clientes iniciais da cada rota com as quantidade de rotas da solução final. Na segunda fase criam-se as rotas e depois insira-se o resto dos clientes nelas.

Fase 1 Construção de rotas sequenciais

1. Primeira rota. Defina um índice para a primeira rota, $k := 1$;
2. Inserção do custo. Selecione qualquer vértice não roteado i_k para inicializar a rota k . Em cada vértice não roteado i , calcular $\delta_i = c_{0i} + \lambda c_{ii_k}$;
3. Inserção do vértice. Seja $\delta_{i^*} = \min_{i \in S_k} \{\delta_i\}$, onde S_k é o conjunto de vértices não roteados que pode ser inserido de forma viável na rota k . Insira o vértice i^* na rota k . Pode-se otimizar a rota k usando o algoritmo de *3-opt*. Repita o Passo 3 até que nenhum outro vértice possa ser atribuído à rota k ;
4. Próxima rota. Se todos os vértices foram inseridos nas rotas, pare. Caso contrário, defina $k := k + 1$ e volte para o Passo 2.

Fase 2 Construção das rotas paralelas.

5. Inicializações das rotas. Iniciar k rotas $R_t = (0, i_t, 0)$ ($t = 1, \dots, k$), onde o número de rotas obtidas no final da Fase 1 é representado por k . Além disso se dispõe do conjunto $J = \{R_1, \dots, R_k\}$;
6. Custo da associação. Para cada vértice i ainda não associado a uma rota e para cada rota viável $R_t \in J$, calcule $\epsilon_{ti} = c_{0i} + \mu c_{ii_t}$ e $\epsilon_{t^*i} = \min_t \{\epsilon_{ti}\}$. Associar o vértice i com a rota R_{t^*} e repita o 6 garantindo associar cada ponto em uma rota;
7. Inserção do custo. Tome qualquer rota $R_t \in J$ e defina $J = J \setminus \{R_t\}$. Para cada vértice i associado à rota R_t , calcule $\epsilon_{t'i} = \min_{R_t \in J} \{\epsilon_{ti}\}$ e $\tau_i = \epsilon_{t'i} - \epsilon_{ti}$;
8. Inserção do vértice. Inserir na rota R_t o vértice i^* satisfazendo $\tau_{i^*} = \max_{i \in S_t} \{\tau_i\}$, onde S_t é o conjunto de vértices não roteados associados à rota R_t que pode ser inserido na rota R_t . Otimize a rota R_t , usando o algoritmo *3-opt*. Repita o Passo 8 até que não exista mais vértices que pode ser inserido na rota R_t ;
9. Verificação da terminação. Se $|J| \neq \emptyset$, volte para 6. Caso contrário, se todos os vértices forem roteados, pare. Se vértices não roteados ainda permanecerem, crie novas rotas começando no Passo 1 da Fase 1.

Maiores detalhes da heurística de inserção pode ser vista em [Toth & Vigo \(2002, 2014\)](#).

3.2.2 Heurísticas de Duas Fases

Esta heurística compreende duas fases no desenvolvimento da solução. A primeira fase, chamado agrupamento dos vértices (*clustering*), designa todos os pontos ou dados em um grupo ou *cluster* usando métricas de similaridade, geralmente para os PRV usam-se a distância. A segunda fase, chamado roteamento (*routing*), se refere à construção da rota final, onde define a sequência de visita dos clientes. Note que essas fases podem ser invertidas ([Cordeau et al. 2007](#)), como segue.

- a. Agrupar primeiro e rotear depois. Do inglês *clustering first and routing second*, aqui os clientes são organizados em grupos, atribuídos a um veículo, e forma-se uma rota para cada grupo ou *cluster*.
- b. Rotear primeiro e agrupar depois. Do inglês *routing first and clustering second*, onde se gera uma solução que visita todos os clientes, para logo criar rotas de diferentes divisões que sejam viáveis.

Agrupar primeiro e rotear depois

O algoritmo que agrupa primeiro e roteia depois é feito em duas fases. Na primeira fase, criam-se os conjuntos de clientes ou *clusters*, os clientes pertencentes a este grupo estarão em uma mesma rota na solução final. Na criação dos grupos deve-se considerar que a soma das demandas associadas aos clientes não deve ultrapassar a capacidade do veículo. Na segunda fase, criam-se as rotas para cada um dos grupos de clientes usando métodos baseados no PCV, esse será resolvido de forma exata ou aproximada, dependendo do tamanho do *clusters* (Toth & Vigo 2002).

Os principais algoritmos que usam o método de agrupar primeiro e rotear depois são:

- Algoritmo de Fisher e Jaikumar (FJ);
- Algoritmo de Bramel e Simchi-Levi (BS);
- Algoritmo de varredura (*The Sweep algorithm*).

Algoritmo de Fisher e Jaikumar

O algoritmo de Fisher e Jaikumar (FJ) (Fisher & Jaikumar 1981), é um método bem conhecidos na classe dos algoritmos que agrupam primeiro e roteiam depois. Na primeira fase os clientes designam-se em m -grupos ou *cluster*, como um problema de designação generalizado, onde um *cluster* denota um conjunto de clientes visitado por um veículo. Seleciona-se um cliente semente j_k para cada *cluster* ou rota k (do conjunto de veículos K), o custo de designar o cliente i ao *cluster* k é calculado como $d_{ik} = \min \{c_{0i} + c_{ij_k} + c_{j_k 0}, c_{0j_k} + c_{j_k i} + c_{i0}\} - (c_{0j_k} + c_{j_k 0})$. Na segunda fase, determina-se a melhor rota para cada *cluster*, resolvendo o problema como um PCV. A fase de designação é definida pelo seguinte modelo:

$$\text{Minimizar} \quad \sum_{k=1}^K \sum_{i \in \bar{V}} d_{ik} x_{ik} \quad (3.1)$$

$$\text{Sujeito a} \quad \sum_{k=1}^K x_{ik} = 1, \quad i \in \bar{V}, \quad (3.2)$$

$$\sum_{i \in \bar{V}} d_{ik} x_{ik} \leq Q, \quad k \in K, \quad (3.3)$$

$$x_{ik} \in \{0, 1\}, \quad i \in \bar{V}, \quad k \in K. \quad (3.4)$$

A variável x_{ik} indica se o cliente está designado a um k grupo. O objetivo é minimizar o custo total da designação. A restrição (3.2) indica que cada cliente é designado exatamente a um *cluster*. Na restrição (3.3) é imposto que a demanda do cliente no *cluster* não pode exceder a capacidade do veículo.

Segundo o livro de Toth & Vigo (2002) descreve-se este algoritmo em três passos:

1. Iniciação. Selecionar um vértice semente $j_k \in V$ para a formação do *cluster* k ;

2. Designação dos vértices. Determinar o custo de designar cada vértice i para cada $cluster$;
3. Roteamento. Para cada $cluster$ se usa um algoritmo que resolva o PCV.

Algoritmo de varredura (*The Sweep algorithm*)

No algoritmo de varredura, a primeira fase é criar os *clusters*, eles são formados girando uma semirreta com origem no depósito, “varrendo” aos clientes a seu alcance e incorporando ao grupo, enquanto não se viole a restrição da capacidade do veículo. Na segunda fase aplica-se para cada *cluster* uma metodologia de solução para o PCV, seja de forma exata ou aproximada, para definir a sequência da rota. Para nosso conhecimento, as primeiras menções deste método são encontradas no livro de [Wren & Carr \(1971\)](#) e dentro de um artigo desenvolvido por [Wren & Holliday \(1972\)](#).

O método é comumente atribuído a [Gillett & Miller \(1974\)](#). Para facilitar a implementação deste método, é preferível representar os vértices por suas coordenadas polares (θ_i, ρ_i) , onde θ_i é o ângulo e ρ_i é o comprimento do raio. Atribua um valor $\theta_{i^*} = 0$ a um vértice arbitrário i^* e calcule os ângulos restantes de $(1, i^*)$. Classifique os vértices em ordem crescente de θ_i ([Toth & Vigo 2002](#)).

Como se mencionou anteriormente, o algoritmo de varredura consiste de duas fases (agrupamento e roteamento). Na primeira fase, todos os vértices se agrupam em função de sua capacidade. Na segunda fase, a capacidade das rotas dos veículos é o número máximo de mercadorias que podem ser transportados em uma rota. Neste caso, o número máximo de produtos transportadas pelo veículo depende da capacidade do próprio veículo ([Nurcahyo et al. 2002](#)).

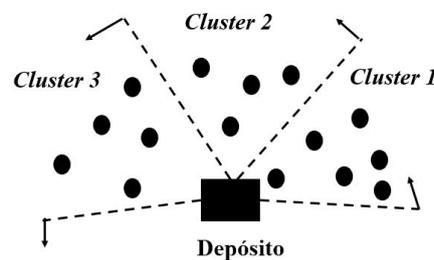


Figura 3.4: Representação da heurística de Varredura. Fonte: figura retirada de [Nurcahyo et al. \(2002\)](#).

- Fase da formação dos *clusters*: A Figura 3.4 ilustra um processo de agrupamento de vários vértices, onde cada ponto representa um vértice i e as linhas retas representam a mão de varredura que se move em sentido anti-horário. O agrupamento começa desde o vértice mais próximo até qualquer vértice afastado, onde o vértice mais próximo é o que tem o ângulo θ_i menor. Este processo repete-se unindo o segundo mais próximo, o terceiro e assim sucessivamente até que satisfaça a restrição de capacidade do veículo. Quando o vértice $[i + 1]$ se une com o vértice do primeiro grupo, a demanda $[i + 1]$ se incorpora à demanda d_i e depois se verifica a restrição de capacidade. O vértice $[i + 1]$ se une com o vértice inicial se a demanda total é menor que a restrição de capacidade. Quando um vértice não pode ser incluído no primeiro *cluster*, por violar a restrição de capacidade, este vértice se converte no primeiro vértice do segundo *cluster*. O processo completa-se quando todos os vértices estão incluídos nos *clusters*;

- Fase da geração da rota. Visa vincular todos os vértices que forma um *cluster*, começando e terminando no mesmo depósito. O resultado desse processo é um caminho mais curto conectando todos os vértices do *cluster*. Nesta fase, uma solução para o problema do caixeiro viajante é necessária para as formações das rotas.

Segundo [Toth & Vigo \(2002\)](#) uma possível implementação do método é a seguinte:

1. Escolha um veículo não utilizado k ;
2. Partindo do vértice não roteado com o menor ângulo, atribua vértices ao veículo, desde que sua capacidade não seja excedida. Se vértices não roteados permanecerem, volte ao Passo 1;
3. Otimize cada rota de veículo separadamente, resolvendo o PCV correspondente (de forma exata ou aproximada). Realize trocas de vértices entre rotas adjacentes se isso economizar a distância. Otimize novamente e pare.

Algoritmo de Bramel e Simchi-Levi (BS)

O algoritmo de [Bramel & Simchi-Levi \(1995\)](#) é semelhante à de Fisher e Jaikumar. Em ambos, existem clientes designados para um conjunto de clientes sementes. Porém, nesta proposta, os clientes sementes são determinados pelo algoritmo, resolvendo um Problema de Localização dos Concentradores Capacitados (PLCC), descrito da seguinte forma. Disponibilizam-se de m possíveis localizações para os concentradores capacitados $Q_j = \{1, \dots, m\}$ e n terminais, cada um dos quais usa $W_i = \{1, \dots, n\}$ da capacidade do concentrador onde se conecta. O custo de instalar um concentrador na localização j é f_j e o custo de conectar o terminal i ao concentrador j é \hat{c}_{ij} . O PLCC consiste em decidir quais concentradores colocar e quais terminais conectar a cada concentrador, de modo que cada terminal se conecte a um único concentrador, satisfazendo as restrições de capacidade e de minimizar os custos.

$$\text{Minimizar} \quad \sum_{j=1}^m f_j y_j + \sum_{i=1}^n \sum_{j=1}^m \hat{c}_{ij} x_{ij} \quad (3.5)$$

$$\text{Sujeito a} \quad \sum_{j=1}^m x_{ij} = 1, \quad i = 1, \dots, n, \quad (3.6)$$

$$\sum_{i=1}^n w_i x_{ij} \leq y_j Q_j, \quad j = 1, \dots, m, \quad (3.7)$$

$$x_{ij}, y_j \in \{0, 1\}, \quad i = 1, \dots, n, \quad j = 1, \dots, m. \quad (3.8)$$

A variável binária y_j indica se um concentrador está instalado no vértice j e a variável binária x_{ij} , indica se o terminal está conectado ao concentrador j . A função objetiva (3.5) é a soma dos custos fixos de instalação dos concentradores e os custos variáveis de conexão dos terminais. A restrição (3.6) garante que cada terminal se conecte exatamente a um concentrador e a restrição (3.7) indica, por um lado, que se um concentrador for instalado na localização j , a demanda dos terminais conectados a eles não ultrapassem sua capacidade e, por outro lado, que se for decidido não instalar um concentrador em j , não se conectam terminais a esse concentrador.

O PLCC fornece uma forma para resolver o PRV, propondo às possíveis sementes como pontos para instalar os concentradores. Definimos m subconjuntos de clientes T_1, \dots, T_m , o custo de usar o subconjunto

T_j na solução é $t(T_j)$, considerando que a demanda dos clientes de T_j não superem a capacidade do veículo. Além disso, para qualquer cliente i se gera um custo de incorporação ao sub conjunto t_j representado por: $\hat{c}_{ij} = t(T_j \cup \{i\}) - t(T_j)$. O problema de decidir quais sementes usar e quais clientes conectar com quais sementes é um PLCC onde os possíveis concentradores são as possíveis sementes, os terminais são os clientes, os custos fixos são os custos de roteamento de cada semente e os custos de conexão são os custos de inserção de clientes nas sementes.

Os passos que descrevem este algoritmo é apresentado em [Bramel & Simchi-Levi \(1995\)](#), onde.

1. Início. Selecione m sementes T_1, \dots, T_m e calcular para cada semente j , $f_j = t(T_j)$ e $Q_j = Q - \sum_{k \in T_j} q_k$. Para cada semente j e cada cliente i , calcular $\hat{c}_{ij} = t(T_j \cup \{i\}) - t(T_j)$. Definir $w_i := q_i$.
2. Localização. Resolva um PLCC com as sementes como possíveis pontos para a instalação de concentradores e os clientes como terminais, usando os custos e capacidades calculados no Passo 1.
3. Roteio. Cada concentrador representa um *cluster* formado. Resolva o roteamento em cada *cluster*.

Rotear primeiro e agrupar depois

Os métodos de rotear primeiro e agrupar depois, do inglês *Route First and Cluster Second Methods*, consiste em duas fases, como podemos ver a seguir. Na primeira calcula-se uma rota que visita todos os clientes resolvendo um PCV sem ter em conta as restrições do problema. Depois na segunda fase, esta rota decompõe-se em várias rotas factíveis, considerando a solução da primeira fase, determina-se a melhor partição sem violar a restrição da capacidade do veículo ([Toth & Vigo 2002](#))

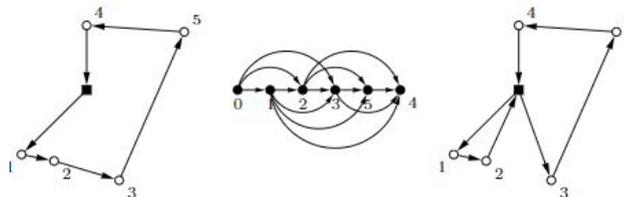


Figura 3.5: Representação da heurística de duas fases. Fonte: figura retirada de [Beasley \(1983\)](#).

Segundo [Beasley \(1983\)](#), seja $r = (0, 1, \dots, n, 0)$ a solução do PCV da primeira fase, para a segunda fase, determina-se a melhor partição de r que respeita a restrição de capacidade do veículo. Este problema pode-se formular como a busca de um caminho de distância mínima em um grafo dirigido acíclico. Seja o grafo $G = (X, V, W)$ onde $X = \{0, 1, \dots, n\}$. As arestas de G , conectam todos os pares de clientes i e j com $i < j$, e a demanda dos clientes $i + 1, \dots, j$ não superem a capacidade do veículo: $V = \{(i, j) \mid i < j, \sum_{k=i+1}^j d_{v_k} \leq Q\}$. Cada aresta (i, j) calcula-se com o custo da rota $(0, v_{i+1}, \dots, v_j, 0)$, isto é, $w(i, j) = c_{0, i+1} + c_{j, 0} + \sum_{k=i+1}^{j-1} c_{k, k+1}$.

Uma aresta (i, j) representa a rota $r = (0, 1, \dots, n, 0)$. Cada sub rota de 0 para n representa uma possível partição da rota r , respeitando a restrição de capacidade da demanda. Portanto, a sob rota de custo mínimo entre 0 para n representa a partição de custo mínimo da rota original em rotas que respeitam a restrição de capacidade.

Algoritmo de pétalas

Proposto por [Ryan et al. \(1993\)](#), esta técnica surge como extensão do algoritmo de varredura e se utiliza para gerar diversas rotas chamadas pétalas com o objetivo de uma seleção final. Dispõe-se de um conjunto de rotas, em onde, a cada cliente é visitado por vários veículos e deve ser selecionado um sub conjunto das mesmas que visite exatamente uma vez cada cliente ([Rochat & Semet 1994](#)).

Os algoritmos de pétalas consiste de duas fases. Na primeira calcula-se uma grande rota que visita a todos os clientes resolvendo um PCV, sem considerar as restrições do problema. Depois na segunda fase, esta rota decompõe-se em várias rotas semelhantes, considerando as restrições de capacidade dos veículos ([Cordeau et al. 2007](#)).

Uma forma de apresentar o algoritmo de pétalas é usando o seguinte modelo de [Toth & Vigo \(2014\)](#).

$$\text{Minimizar} \quad \sum_{c \in S} c_k x_k \quad (3.9)$$

$$\text{Sujeito a} \quad \sum_{k \in K} a_{ik} x_k = 1, \quad i = 1, \dots, n, \quad (3.10)$$

$$x_k \in \{0, 1\}, \quad k \in K. \quad (3.11)$$

Suponha-se que temos um conjunto de veículos K , de modo que cada veículo $k \in K$ é usado, onde cada cliente é visitado por vários veículos. O problema é selecionar um subconjunto de K que visita cada cliente exatamente uma vez com o mínimo custo, onde a_{ik} é 1, se o cliente for visitado pelo veículo k e 0, caso contrário, e c_k é o custo da rota para o veículo k . A variável x_k determina se a rota do veículo k está na rota final ([Balinski & Quandt 1964](#)).

3.2.3 Heurísticas de Melhoramento

Nessas heurísticas, partimos de uma determinada solução inicial e a melhoramos progressivamente. A cada iteração, o algoritmo deve encontrar uma melhor solução do que a anterior, finalizando o processo quando não for encontrado nenhuma iteração que melhore a solução atual ([Tavares et al. 2009](#)). [Lin \(1965\)](#) foi o responsável pela proposta de um algoritmo baseado em trocas de arcos ou movimentos de clientes. A solução usando este algoritmo pode ser melhorada ou piorada por cada uma das interações ou trocas efetuadas, o objetivo é melhorar a rota de modo a manter o controle sobre o processo de busca. A mistura de diferentes movimentos para alterar a estrutura da solução realizada neste método faz com que o ótimo seja alcançado lenta e gradualmente ([Van Breedam 1995](#)).

As heurísticas que melhoram a resolução do PRV operam em cada rota de veículo escolhido separadamente ou em várias rotas em simultâneo. No primeiro caso, qualquer heurística de melhoria para o PCV pode ser aplicado. No segundo caso, os procedimentos que exploram as estruturas de múltiplas rotas do PRV podem ser vistas em [Toth & Vigo \(2014\)](#). As heurísticas de melhoria trabalham em uma solução viável. Nelas existem as variantes intra-rota, que trocam as arestas na mesma rota (heurísticas *2-opt*, *3-opt*) e as variantes extra-rota, que trocam os arcos entre dois ou mais rotas diferentes.

3.3 Meta-heurísticas

No tópico anterior mostramos as vantagens de utilizar as heurísticas para resolver as variantes do PRV, em muitas ocasiões estes procedimentos produzem soluções que podem estar não tão próxima das soluções ótimas globais. Como uma forma de aproximar ainda mais de uma solução aceitável, desenvolveram-se as denominadas “meta-heurísticas”, as quais contam com estratégias que permitem resolver de forma inteligente um problema (Melián et al. 2003).

As meta-heurísticas foram desenvolvidas a partir dos anos 90 e são caracterizadas por técnicas projetadas para resolver problemas de otimização difíceis, nos quais as heurísticas clássicas, em grande maioria, não são eficazes. Elas também se apresentam como modificações de outras heurísticas, combinando diferentes conceitos para produzir melhores soluções. Com o uso das meta-heurísticas não se garante a exploração completa do espaço de soluções, no entanto elas exploram aquelas regiões de soluções viáveis (Vera 2005).

As meta-heurísticas mais usadas para resolver os PRVs são: Otimização de Colônias de Formiga (ACO), Algoritmo Evolutivo (AE) – Algoritmos Genéticos (AG), procedimentos de busca Míope, aleatorizados e adaptativos do tipo GRASP, Busca Local Iterativa (BLI), Busca Dispersa (BD), Busca Tabu (BT), dentre outras (Cordeau et al. 2002, Toth & Vigo 2002).

3.3.1 Algoritmo de Busca Tabu

Foi introduzido por Glover (1986) como uma meta-heurística iterativa para resolver problemas de otimização combinatória. Muitas experimentos computacionais mostraram que a BT é uma técnica bem estabelecida de aproximação, que pode competir com quase todas as técnicas conhecidas e que, por sua flexibilidade, podem ser melhores que alguns procedimentos clássicos. Podemos interpretar esta meta-heurística como uma forma de busca de vizinhos locais. Cada solução S tem um conjunto associado de vizinhos $N(S)$. Uma solução $S' \in N(S)$ pode ser alcançado a partir de S por uma operação de movimento. Então a Busca Tabu é uma técnica iterativa que explora um conjunto de soluções no problema, fazendo movimentos repetitivos de uma solução S para outra solução S' localizada no conjunto $N(S)$ (Laguna & Glover 1993).

Os algoritmos de Busca Tabu resolve o PRV passando de uma solução para outra solução admissível, mesmo que a função objetiva se deteriore. Para evitar a repetição, as soluções que foram recentemente exploradas são declaradas proibidas ou tabu para uma série de iterações. A lista tabu de uma solução é substituído quando determinados critérios (critérios de aspiração) são satisfeitos. Às vezes, estratégias de intensificação e diversificação são usadas para melhorar a busca. No primeiro caso, a busca é acentuada nas regiões promissoras do domínio viável. No segundo caso, uma tentativa é feita considerando soluções em uma área ampla de busca (Marinakis et al. 2010, Toth & Vigo 2002).

3.3.2 Algoritmo Genético

Proposta por Holland (1962), esta técnica é baseado na teoria da evolução darwiniana, basicamente uma população de soluções (conhecido como cromossomos) evolui de uma geração para a próxima por meio da aplicação de operadores semelhantes aos encontrados na natureza, como seleção do mais apto, cruzamento genético e mutação. Graças à seleção, apenas as melhores soluções podem se tornar pais e gerar descendentes. O processo conhecido como cruzamento, leva duas soluções pai e combina seus caracteres mais

relevantes para criar uma ou duas soluções descendentes. Isso é repetido até que uma nova população de descendentes seja obtido. Finalmente, cada filho é perturbado aleatoriamente por um operador de mutação. A partir de uma população inicial gerada aleatoriamente ou heurísticamente, este ciclo se repete por várias gerações, e a melhor solução encontrada é retornada no final. Quando este algoritmo é aplicado ao PRV, o clássico o esquema do AG é frequentemente modificado (Golden et al. 2008).

3.3.3 Algoritmo de Colônias de Formiga

Estes algoritmos foi proposto por Dorigo (1992), a qual foi inspirado na estratégia utilizada por colônias de formigas na busca de alimento. Quando um formiga encontra seu caminho para o alimento deposita uma substância (feromônio) esta depende da extensão do caminho e da qualidade da comida. As formigas tendem a seguir os caminhos com maior quantidade de feromônios, este conceito é aplicado ao PRV, pela qual causa um reforço das melhores rotas, quer dizer, aquelas que demoram menos tempo e por onde transita o maior número de formigas (Dorigo 1992).

O Algoritmo de Colônias de Formiga (OAC) pertencem ao campo da *swarm intelligence* ou inteligência de colônia, a qual esta compostos por indivíduos simples que cooperam de forma auto-organizada, isto é, sem nenhuma forma de controle central sobre os membros da colônia. Esta é uma técnica de otimização que imita o comportamento de colônias de formigas reais, concretamente nas feromônio que depositam entre a comida e o ninho para marcar desta forma o melhor caminho encontrado. A quantidade de feromônio em um caminho aumenta a cada vez que uma formiga o atravessa. À medida que esta quantidade incrementa, a probabilidade de que uma formiga siga esse caminho também o faz, pelo que a quantidade de feromônios no caminho mais curto será maior após um determinado tempo e, como consequência, um maior número de formigas tenderão a selecionar dito caminho. No entanto, a decisão de seguir uma rota nunca é determinista, pois se permite a exploração contínua de rotas alternativas. Estes algoritmos utilizam também um procedimento de evaporação que reduz a quantidade de feromônio ao longo do tempo, dando mais ênfase em novas direções de busca para evitar a parada de progressão em mínimos locais (Ostfeld 2011)

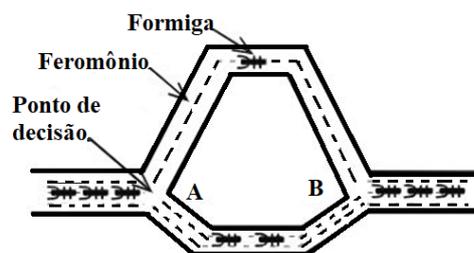


Figura 3.6: Representação da meta-heurística AOC. Fonte: figura retirada de Dorigo et al. (2006).

Aplicação do algoritmo AOC

No algoritmo AOC cada formiga fornece uma solução do problema sequencialmente. Em cada etapa, a formiga realiza um movimento para completar a solução parcial real, escolhendo entre os elementos de um conjunto A_k seguindo uma função de probabilidade (Dorigo & Di Caro 1999, Dorigo et al. 2006, Mazzeo & Loiseau 2004). Quando uma formiga escolhe um caminho esta atua seguindo o rastro de feromônio, a qual

é proporcional á probabilidade do caminho selecionado pela formiga e á atratividade do feromônio. Se esta formiga passou pelo caminho com maior atração de feromônio, a sua probabilidade de escolher outro é zero, no caso que a formiga não consiga perceber a atração do feromônio terá que criar outra rota. O caminho que percole a formiga pode ser definida pela formulação de [Dorigo et al. \(1996\)](#).

Para a cada formiga k a probabilidade $P_k(i, j)$ de passar do ponto i para outro ponto j é calculado como

$$P_k(i, j) = \begin{cases} \frac{[\tau_{ij}(t)]^\alpha \times [\eta_{i,j}(t)]^\beta}{\sum_{l \in N_i^k} [\tau_{il}(t)]^\alpha \times [\eta_{i,l}(t)]^\beta}, & \text{se } (i, j) \in N_i^k \\ 0, & \text{caso contrário.} \end{cases}$$

Segundo está regra, denominada regra de transição, a probabilidade ρ de uma formiga k , que está situada no ponto i e que pretende caminhar até j , é dada pelos seguintes parâmetros:

$\tau_{ij}(t)$: feromônio presente na aresta (i, j) no instante t ;

α e β : variáveis que determinam a influência do feromônio τ_{ij} e da informação heurística η_{ij} ;

η_{ij} : representa a atratividade da formiga pela aresta (i, j) , quer dizer a atração da formiga de viajar do ponto i até j , esta atração pode ser representado por $\eta_{ij} = \frac{1}{d_{ij}}$, onde d_{ij} representa a distância, geralmente é usada a versão Euclidiana $\left(d_{ij} = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2}\right)$;

N_i^k : Esta variável apresenta as vizinhança factível de uma formiga k , isto é, todas as cidades adjacentes ainda não visitadas pela mesma.

O próximo passo é o processo de atualização do feromônio nas arestas, aqui o feromônio τ_{ij} atua de duas formas:

1. No depósito de feromônio: procura reforçar positivamente a escolha da rota, tornando-se o melhor caminho selecionado pelas formigas no momento de percorrer as arestas (i, j) .
2. Na evaporação: visa esquecer as decisões erradas tomadas nos passos anteriores, evitando também o crescimento indefinido do feromônio.

A quantidade $\tau_{ij}^k(t)$ da substância a ser depositada em uma aresta (i, j) pela formiga k é dada pela fórmula $\Delta\tau_{ij}^k(t) = \frac{Q}{L_k(t)}$, na qual Q corresponde ao valor total de feromônio exalado por uma formiga a cada iteração e $L_k(t)$ diz respeito ao comprimento completo da rota descoberta pela formiga k , naquele instante t . Desta forma, o feromônio $\tau_{i,j}$ associado à aresta (i, j) será atualizado. Esta atualização é formulada por $\tau_{ij}(t+1) = (1 - \rho)\tau_{ij}(t) + \Delta\tau_{ij}(t)$, onde:

$$\Delta\tau_{ij}(t) = \sum_{k=1}^m \Delta\tau_{ij}^k(t);$$

m : corresponde a quantidade total de formigas;

ρ : trata-se da taxa de evaporação do feromônio ($0 < \rho \leq 1$);

$\Delta\tau_{ij}(t)$: refere-se ao cálculo para depósito da substância.

No caso de VRP, o modo de operar deste algoritmo é resumido da seguinte forma: O algoritmo é inicializado colocando uma formiga em cada nó. Para a construção de caminhos, uma regra probabilística é usada, a qual é igual a zero se o nó já foi visitado e diferente de zero para o caso oposto. A formiga visita o nó que tem uma probabilidade maior. Em cada arco o “feromônio” é atualizado e termina quando é resolvida uma solução inferior a uma dimensão pré-estabelecida, caso contrário são recalculados probabilidades e a formiga continua construindo soluções ([Dorigo et al. 2006](#)).

Algoritmo GRASP

Greedy Randomized Adaptative Search Procedure (GRASP) é uma técnica desenvolvida por [Feo & Resende \(1989\)](#) para resolver problemas de alta complexidade combinatória, cuja tradução literal seria “procedimentos de busca, aleatorizados e adaptativos”. Esta metaheurística caracteriza-se de outras por dirigir a maior parte de seu esforço a construir soluções de alta qualidade que são posteriormente desenvolvidas para obter outras melhores ([Feo & Resende 1995](#)). O GRASP desenvolve-se em duas fases. A primeira fase inclui a construção sequencial de soluções viáveis. Em cada iteração, todos os possíveis movimentos se ordenam de acordo com uma função adaptativa, que seleciona aleatoriamente um movimento para uma lista restritiva de candidatos, o qual supõe um equilíbrio entre a qualidade e a variabilidade das configurações na construção da solução. A segunda fase trata de melhoria da solução anterior mediante uma busca local. As vezes prefere-se a construção de um conjunto de opções iguais e depois procede-se uma busca com as mais promissoras ([Feo & Resende 1989](#)).

3.4 Técnicas de Agrupamento

Nesta seção aborda-se algumas técnicas de agrupamento, especialmente o algoritmo *k-menas*. No Capítulo 4 usará-se este algoritmo para resolver o PRVMD formando os grupos de clientes em cada depósito.

As técnicas de agrupamento analisam um conjunto de observações disponíveis para determinar a tendência dos padrões no agrupamento. [Kendall \(1966\)](#) considerou essas técnicas como uma forma especial de classificação. Elas permitem efetuar uma classificação atribuindo cada observação para um *cluster*, de forma que cada agrupamento seja aproximadamente homogêneo e distinguível dos demais ([Jain et al. 1999](#)).



Figura 3.7: Tipos de técnicas de agrupamento. Fonte: figura adaptada de [Jain et al. \(1999\)](#).

No PRV, a distância é geralmente usada como uma forma de agrupamento (medida de similaridade), algumas das métricas de distância mais utilizadas são a distância Euclidiana e a distância de Mahalanobis ([Dondo & Cerdá 2007](#)). A Figura 3.7 ilustra uma classificação sugerida por [Lance & Williams \(1967\)](#) onde cada nível define um tipo diferente para os problemas de classificação. Quando a técnica de agrupamento é supervisionada considera-se uma classificação, e quando não é supervisionado chama-se de *clustering*. Este último pode servir também para extrair características latentes dos dados considerando sua natureza.

3.4.1 Agrupamento Supervisionado

O agrupamento supervisionado possui um conhecimento “a priori” das informações dos objetos, quer dizer, para poder classificar um objeto dentro de uma categoria ou classe conta-se com os modelos prontos (objetos agrupados que possuem características comuns). Podemos diferenciar duas fases nesta técnica de

agrupamento, na primeira fase temos um conjunto de treinamento ou aprendizagem (para o desenho do classificador) e outro chamado teste ou validação (para a classificação), estes servirão na construção de um modelo ou regra geral para o agrupamento. Na segunda fase, o processo de classificação dos objetos ou amostras é desconhecida.

Para se entender melhor a técnica de agrupamento supervisionado suponha um conjunto de dados/amostras, que são conhecidas a priori, além disso, conta-se com as variáveis de entrada x e uma de saída Y , onde um algoritmo é usado para aprender a função de mapeamento de entrada e saída dos dados. O objetivo é aproximar a função de mapeamento de forma que quando se tiver os novos dados de entrada x , pode-se prever a variável de saída Y para esses dados. Estas técnicas podem ser agrupados em problemas de regressão e classificação.

- Classificação: quando a variável de saída é categórica, por exemplo “vermelho” ou “azul”;
- Regressão: quando a variável de saída é um número real, por exemplo “dólares” ou “reais”.

3.4.2 Agrupamento não Supervisionado

O agrupamento não supervisionado consiste em agrupar os objetos com alto grau de similaridade, onde as semelhanças são interpretadas por alguma função de distância. Dentro das técnicas de agrupamento não supervisionadas não existe necessidade de informações “a priori” sobre o domínio avaliado, apenas a disposição dos dados e suas propriedades internas (Kwon et al. 2017). Segundo Jain et al. (1999) existem diversos métodos para a procura do número ideal de conjuntos finais que devem ser criados na representação do problema em estudo.

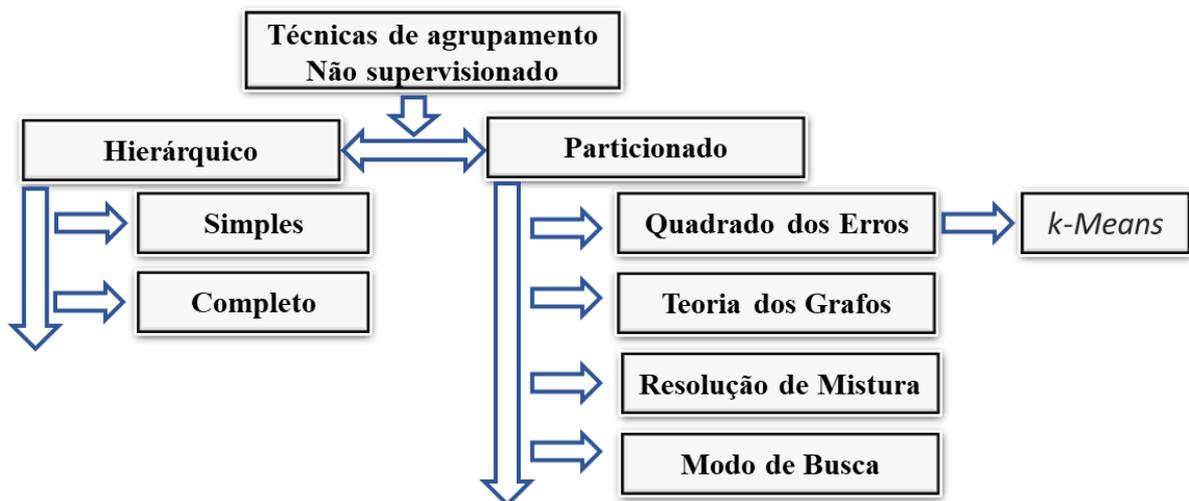


Figura 3.8: Técnicas de agrupamento não supervisionado. Fonte: figura adaptada de Jain et al. (1999).

As técnicas de agrupamento não supervisionadas podem ser classificadas segundo a forma de como são interpretadas os dados e a maneira desses objetos em organizar grupos. Em análise de dados, distingui-se dois grandes grupos de agrupamento não supervisionado: hierárquicos e particionais (MacQueen et al. 1967, Everitt & Hothorn 2011).

Diferentes abordagens para formar os *clusters* dentro das técnicas não supervisionadas podem ser observadas na Figura 3.8, outras representações taxonômicas da metodologia de agrupamento são possíveis,

este trabalho baseia-se na livro de [Jain & Dubes \(1988\)](#). No nível superior da figura, há uma distinção entre as abordagens hierárquica e particionada, os métodos hierárquicos produzem uma série alinhada de partições, enquanto os métodos particionados produzem apenas uma. Segundo [Jain et al. \(1999\)](#), a taxonomia mostrada na Figura 3.8 deve ser complementada por uma discussão de questões transversais que pode, em princípio, afetar todas as diferentes abordagens, independentemente de sua posição taxonômica.

Agrupamento hierárquico.

Um método de agrupamento hierárquico é um procedimento para transformar uma matriz de proximidade em uma sequência de partições alinhadas ([Jain & Dubes 1988](#)). As técnicas hierárquicas produzem uma sequência alinhada de partições, com um único *cluster* completo na parte superior e *clusters* individuais na parte inferior. Cada nível intermediário pode ser visto combinando dois *clusters* do próximo nível inferior ou dividindo um *cluster* do próximo nível superior. O resultado do agrupamento hierárquico pode ser interpretado graficamente por uma árvore, chamada de dendrograma ([Steinbach et al. 2000](#)).

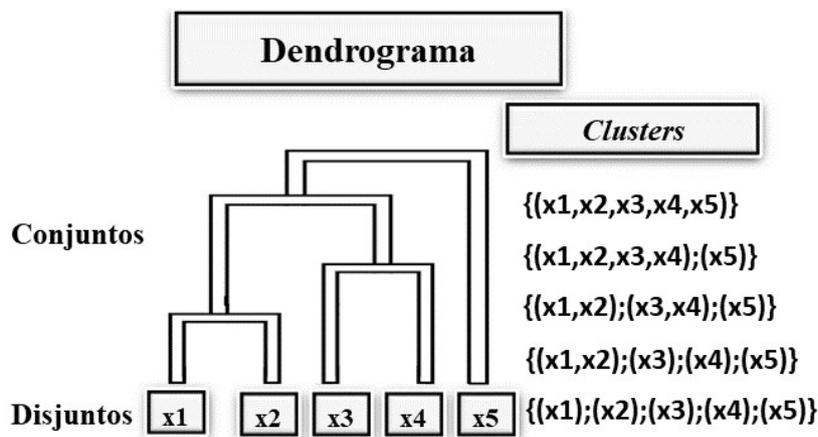


Figura 3.9: Representação do Dendrograma. Fonte: figura adaptada de [Jain & Dubes \(1988\)](#).

Segundo [Jain & Dubes \(1988\)](#) e [Kaufman & Rousseeuw \(2009\)](#) existem duas abordagens básicas no agrupamento hierárquico:

- Aglomerativa: que opera criando conjuntos a partir dos elementos isolados;
- Divisiva: que começa com um grande conjunto e vai dividindo-se em partes até chegar aos elementos isolados.

Agrupamento particionado

Em contraste com as técnicas hierárquicas, as técnicas de agrupamento particionado criam um nível de particionamento não alinhado dos objetos da análise ([Tan et al. 2016](#)). Seja k o número desejado de *clusters*, então as abordagens de partição normalmente encontram todos os k *clusters* de uma vez. Compare isso com os esquemas hierárquicos tradicionais, que dividem um *cluster* para obter dois, ou mesclam dois para obter um. ([Steinbach et al. 2000](#)). Esta forma de agrupamento pode ser usada para gerar uma partição plana de k *clusters* e, da mesma forma, aplicar um esquema repetido de partições pode produzir um *cluster* hierárquico.

Existem várias técnicas de particionamento, para este estudo descreveremos apenas o algoritmo *k-means*, que é amplamente usado para resolver o PRVMD. O algoritmo *k-means* é fundamentada na ideia de

que um ponto central pode representar um *cluster*, usando conceitos de centroide, que representa o ponto médio de um conjunto de dados.

3.4.3 Algoritmo *k-means*

Este algoritmo, também chamado de *k*-médias, foi proposto por MacQueen et al. (1967), este é o algoritmo de agrupamento mais conhecido para resolver problemas de *clustering*, também considerado como uma das melhores técnicas para agrupamento não hierárquico (San et al. 2004). O algoritmo *k-means* é um método que serve no pré-processamento de dados e na identificação de padrões ocultos, também considerado como a ferramenta de agrupamento mais popular utilizado em aplicações científicas e industriais (Berkhin 2006). A análise de *cluster* por *k-means*, determina um critério de como dividir um conjunto de observações em *k*-grupos, o algoritmo está baseado na minimização da distância, isto é, a soma das distâncias dos elementos designados para cada grupo ao centroide dele. Além disso, a distância comumente empregada no desenvolvimento do algoritmo é a distância Euclidiana (Berkhin 2006).

3.4.4 Algoritmo *k-means* na Formação de Grupos

Segundo Jain & Dubes (1988) o algoritmo *k-means* “é um método de agrupamento particionado ou não hierárquico”. O procedimento segue uma etapa de classificação para um conjunto de dados, usando *k cluster* ou grupos fixados “a priori”. O objetivo é determinar os *k* centroides também chamado de baricentros, os quais devem ser alocados nos grupos formados, estes centroides poderiam produzir resultados desfavorável se são alocados nos lugares errôneos, porém é melhor considerar esta alocação o mais longe possível. O próximo passo é a designação dos dados e associá-lo ao centroide mais próximo. Quando todos os pontos estiverem designados num *cluster* a primeira etapa é concluída. O procedimento continua na seguinte etapa onde se recalcula *k* novos centros dos grupos formados na etapa anterior. Após terminar a recalculação dos novos centroides, é realizada novas interações entre os pontos designados nos grupo formados da etapa anterior e os novos centroides recalculados. Como resultado pode se observar a mudança da localização dos baricentros, esta etapa é repetida até conseguir que os centros dos grupos formados não mudem mais de localização (Potvin 1996).

Apresentamos a formulação para a formação dos *cluster* como uma forma de revisão literária. Segundo Selim & Ismail (1984) este algoritmo pode ser expressado como um conjunto $D = \{X_1, \dots, X_n\}$ de dados numéricos com *n* objetos, um número natural $k \leq n$ e uma distância *d*, o algoritmo visa encontrar uma partição de *D* em *k cluster*s disjuntos não vazios, de forma que a soma total das distâncias quadradas entre os objetos e os seus centros de agrupamento é minimizada. A variável $w_{i,l}$ assume o valor de 1, se o objeto X_i estiver designado num *cluster* e 0, caso contrário. O problema pode ser modelado em termos de um problema de otimização não linear da seguinte forma (Selim & Ismail 1984).

$$\text{Minimizar} \quad P(W, Q) = \sum_{l=1}^k \sum_{i=1}^n w_{i,l} d(X_i, Q_l) \quad (3.12)$$

$$\text{Sujeito a} \quad \sum_{l=1}^k w_{i,l} = 1, \quad 1 \leq i \leq n, \quad (3.13)$$

$$w_{i,l} \in \{0, 1\}, \quad 1 \leq i \leq n, \quad 1 \leq l \leq k, \quad (3.14)$$

onde $W = [w_{i,l}]_{n \times k}$ é uma matriz de partição, $Q = \{Q_1, \dots, Q_k\}$ é um conjunto de centros dos *clusters*, e $d(X_i, Q_l)$ é a distância euclidiana ao quadrado entre dois objetos analisados.

Minimizar a função (3.12) é a abordagem usual, sujeito à restrição (3.13) para usar uma otimização parcial para Q e W , ou seja, primeiro fixamos Q e encontramos as condições necessárias para W de forma a minimizar P . Na sequência, fixamos W e minimizamos P de acordo com Q . O algoritmo *k-means* interage por meio dos passos descritos abaixo, até que P esteja próximo de algum mínimo (Selim & Ismail 1984).

1. Seleção inicial de $Q^{(0)} = \{Q_1^{(0)}, \dots, Q_k^{(0)}\}$ e defina $t := 0$;
2. Mantenha-se $Q^{(t)}$ fixo e resolva $P(W, Q^{(t)})$ para obter $W^{(t)}$, considerando $Q^{(t)}$ como os centros do *cluster*, depois atribua cada objeto ao centro do *cluster* mais próximo;
3. Manter $W^{(t)}$ fixo e gerar $Q^{(t+1)}$ de tal forma que $P(W^{(t)}, Q^{(t+1)})$ é minimizado, ou seja, construir novos centros de aglomeração de acordo com a distribuição atual dos objetos;
4. Pare no caso de convergência. Caso contrário, defina $t = t + 1$ e volte ao Passo 2.

No caso de agrupamento de dados numéricos, a norma Euclidiana é dada por $d(X, Y) = \sqrt{\sum_{j=1}^m (x_j - y_j)^2}$.

Algoritmo *k-means* no PRVMD

Nesta parte apresentamos o algoritmo *k-means* que serviu na formação dos *cluster* no caso de estudo apresentado no Capítulo 4. Resolver a primeira fase da heurística que primeiro agrupa e depois roteia, a qual é usada como forma de resolver o PRVMD, baseamos-nós no modelo de *k-cluster* apresentado no trabalho de Seref et al. (2014), onde a variável binária y_i é 1, se o vértice i é selecionado como centro do grupo, caso contrário é 0, e a variável z_{ij} é 1, se o vértice j é designado ao grupo onde o centro do *cluster* é i , caso contrário é 0.

Para o modelo seguinte, se tem n objetos na análise com d características e um número inteiro k que é o número de *cluster*, a distância (similaridade) entre i e j é representado por δ_{ij} . O modelo apresenta a seleção de k objetos como centros de *cluster* e uma designação de objetos aos centros de *clusters* de forma a minimizar a soma das distâncias desde os objetos até os centros dos *clusters*.

$$\text{Minimizar} \quad \sum_{i=1}^n \sum_{j=1}^n \delta_{ij} z_{ij} \quad (3.15)$$

$$\text{Sujeito a} \quad \sum_{i=1}^n z_{ij} = 1, \quad j = 1, \dots, n, \quad (3.16)$$

$$\sum_{j=1}^n z_{ij} \leq n y_i, \quad i = 1, \dots, n, \quad (3.17)$$

$$\sum_{i=1}^n y_i = k, \quad (3.18)$$

$$y_i, z_{ij} \in \{0, 1\}, \quad i, j = 1, \dots, n. \quad (3.19)$$

A restrição (3.16) garante a designação de cada objeto exatamente para um dos centros dos *clusters*. A restrição (3.17) indica que no máximo n objetos são atribuídos ao vértice i , se este fosse selecionado como centro de *cluster*, e que nenhum objeto é atribuído de outra forma. A restrição (3.18) serve para que exatamente k objetos sejam selecionados como centros dos *clusters*.

3.4.5 Regra do Cotovelo para a Análises de k -Cluster

Uma das principais dificuldades da análise dos *clusters* é achar o número k correto de grupos para um conjunto de dados, do qual raramente é conhecido na prática. Assim, identificar este número é uma tarefa importante para qualquer problema de agrupamento, na prática deve ser enfrentado com muitos desafios operacionais. Uma forma tratável da análise dos *clusters* é pedir ao usuário final inserir um número com antecedência, o qual precisa do conhecimento e domínio especializado sobre os conjuntos de dados que serão agrupados. Por outro lado, muitos critérios estatísticos ou índices de validade de agrupamento foram investigados no sentido de selecionar automaticamente um número k apropriado de agrupamentos, entre eles temos: regras práticas, regra do cotovelo, abordagem via teoria de informação, método da silhueta e validação cruzada, dentre outros. Um método bastante utilizado para determinar o melhor número de *clusters* é a regra do cotovelo ou método do cotovelo, sendo considerada uma das mais antigas regras para determinar o número ideal de *clusters* (Kodinariya & Makwana 2013).

Método do cotovelo

Elbow method ou método do cotovelo é usado para produzir informações na determinação do melhor número de *clusters*, a qual é determinada observando a porcentagem da comparação entre os números de agrupamentos que formam o cotovelo num ponto, a Figura 3.10 é apresentado para compreender melhor esta técnica. Este método fornece ponto a ponto selecionando valores de *cluster* e, em seguida, adicionando o valor do *cluster* a ser usado como um modelo de dados para determinar o melhor *cluster*. E, além disso, a porcentagem do cálculo resultante é uma comparação entre o número de *clusters* adicionados (Muningsih & Kiswati 2018).

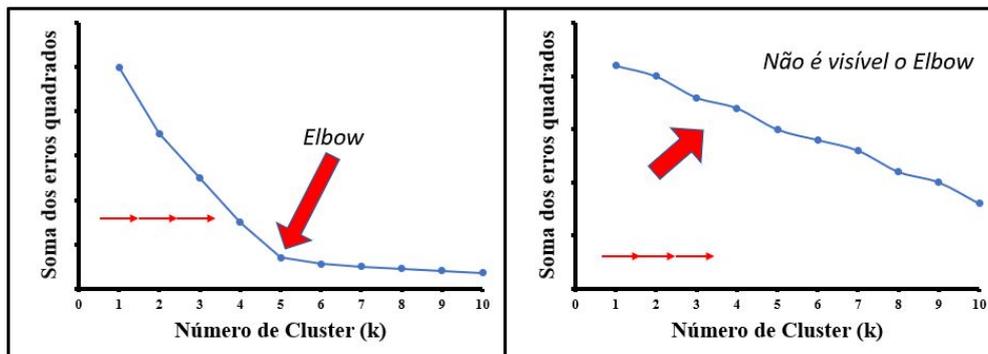


Figura 3.10: Representação da técnica do cotovelo. Fonte: figura adaptada de Nainggolan et al. (2019).

Diferentes resultados de cada valor de *clusters* podem ser mostrados usando o gráfico das somas dos erros quadrados vs os números de *clusters* propostos. Se o valor do primeiro *clusters* com o valor do segundo forma um ângulo no gráfico, então a quantidade do *clusters* é achado (Nainggolan et al. 2019).

O resultado do valor k no método de cotovelo para o algoritmo *k-means* é analisado pelo gráfico que relaciona a quantidade dos *clusters* com a diminuição dos erros, ou seja, quando aumentado o valor de k a soma quadrado dos erros diminuirá lentamente até que o resultado do valor k seja estável. Por exemplo, na Figura 3.10, o valor do *cluster* $k = 4$ para $k = 6$ e de $k = 5$ para $k = 6$, mostram uma diminuição drástica com formato de cotovelo no ponto $k = 5$, então $k = 5$ é possivelmente a quantidade ideal de *cluster* (Sujatha &

Sona 2013). Agora apresentaremos os passos para achar o número ideal de *cluster* para o algoritmo *k-means* segundo o estudo de Syakur et al. (2018).

1. Achar K como o número de *clusters* formados. O método do cotovelo é expresso pela Soma do Erro Quadrado: $SSE = \sum_{k=1}^{|K|} \sum_{x_i \in S_k} \|X_i - C_k\|_2^2$, onde K é o conjunto que contém todos os *clusters* formados. Seja C_i o i -ésimo *cluster*, e x o vetor dos objetos presentes em cada *cluster*;
2. Determine o ponto central do *cluster* no início de forma aleatória. A determinação inicial do centroide é feita aleatoriamente a partir dos objetos disponíveis no conjunto K , em seguida para calcular o próximo centroide do i -ésimo *cluster*, é usada a seguinte fórmula: $v = \frac{\sum_{i=1}^n x_i}{n}$; $i = 1, 2, 3, \dots, n$;
3. Calcule a distância dos objetos a cada centroide, usando a distância Euclidiana: $d(x, y) = \|x - y\| = \sqrt{\sum_{i=1}^n (x_i - y_i)^2}$; $i = 1, 2, 3, \dots, n$. Onde x_i é a variável no objeto x para cada i , y_i é a variável de saída y e n é o número de objetos ou pontos;
4. Aloque cada objeto no centroide mais próximo;
5. Alocação de objetos em cada *cluster* com *k-means*. Onde cada objeto membro do *cluster* foi medido pela distância de proximidade ao ponto central do *cluster*;
6. Execute a iteração e determine a posição do novo centroide usando a fórmula no Passo 2;
7. Se a nova posição do centroide com o antigo não for a mesma, repita o Passo 3.

 Aplicação da Pesquisa em um Estudo de Caso

Este capítulo apresenta uma aplicação do Problema de Roteamento de Veículos com Múltiplos Depósitos em um estudo de caso, utilizando dados práticos fornecidos por uma empresa peruana que procura alternativas para gerir e planejar as suas operações na distribuição de mercadorias. Os parâmetros considerados na modelagem da aplicação com múltiplos depósitos foram a quantidade e capacidade dos veículos, o tipo de frota, a quantidade de clientes, a demanda por produtos e as localizações dos clientes e depósitos.

Para a resolução do modelo multi-depósito empregamos métodos e algoritmos clássicos direcionados a resolução de problemas de programação linear inteira, principalmente o algoritmo *Branch and Bound* via solver e heurísticas descritas no Capítulo 3. Estes métodos foram implementado na linguagem de programação Julia. Experimentos preliminares utilizando metodologia exata esbarraram na complexidade computacional ao resolver exemplares maiores que 200 clientes. Assim, uma abordagem heurística foi desenvolvida para superar as dificuldades presentes no algoritmo exato na resolução do problema.

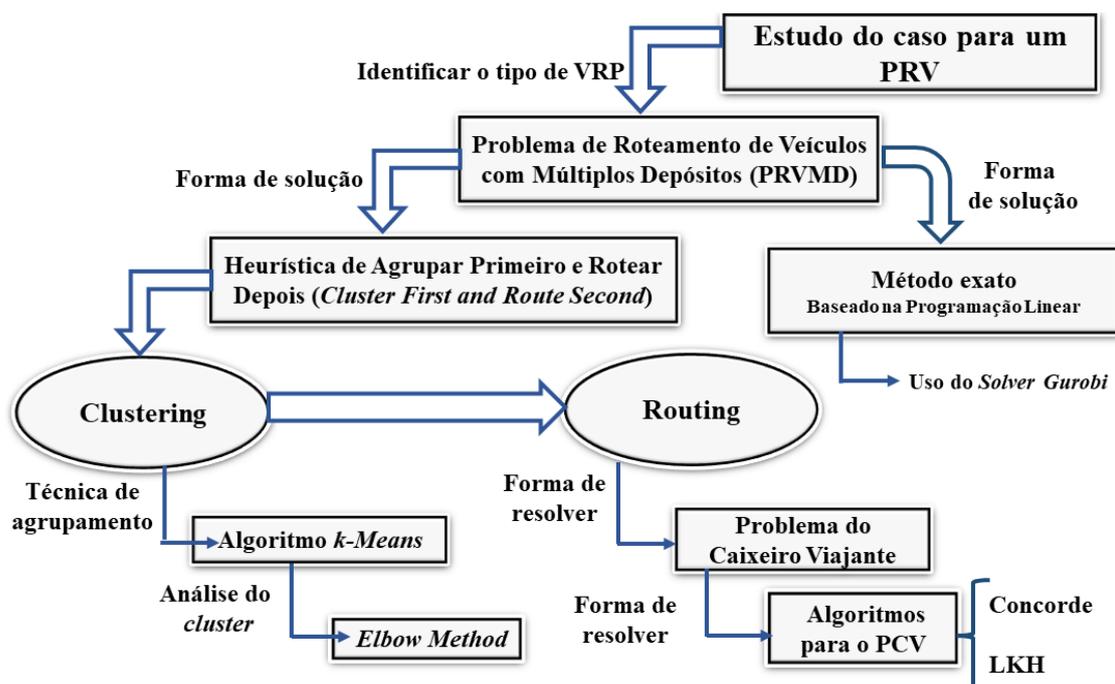


Figura 4.1: Esquema da metodologia para o estudo de caso

A heurística implementada pertence aquela que agrupar primeiro e rotear depois a qual foi escolhida após um estudo da literatura sobre alguns métodos aproximados que resolvem o problema com múltiplos depósitos, como por exemplo o algoritmo de varredura, algoritmo evolutivo, entre outras. Na primeira fase da heurística abordada, se constituiu *clusters* usando o algoritmo *k-means*, onde os vértices representam os clientes, os quais foram designados aos depósitos considerados como centros dos grupos formados. Após a designação, outros grupos foram criados para cada *clusters* de clientes alocados em um dos depósitos, respeitando a restrição de capacidade e demanda. Na segunda fase da heurística usou-se algoritmos baseados no PCV, da mesma forma foi utilizado o algoritmo Concorde e o algoritmo LKH, ou seja, o roteamento foi resolvido por diferentes técnicas. A metodologia proposta forneceu soluções para o problema com múltiplos depósitos como uma ferramenta para o cumprimento da distribuição das mercadorias. A Figura 4.1 apresenta um esquema para ajudar no entendimento da metodologia usada.

4.1 Informações Considerados para o Estudo de Caso

O banco de dados usados para o estudo de caso é fortemente baseado em dados reais de uma empresa, que por motivos de confidencialidade será chamada de EPL1. Esta pertence ao setor agroindustrial, elaborando e repartindo os produtos ao longo da cidade metropolitana de Lima. Os produtos com maior demanda que a empresa oferece para os clientes primários e secundários são: leite, iogurte, manteiga, queijo, sucos, conservas, aveias, panetone, etc. em diferentes apresentações. Chamam-se clientes primários aos mercados e supermercados, e clientes secundários as lojas e *minimarkets*.

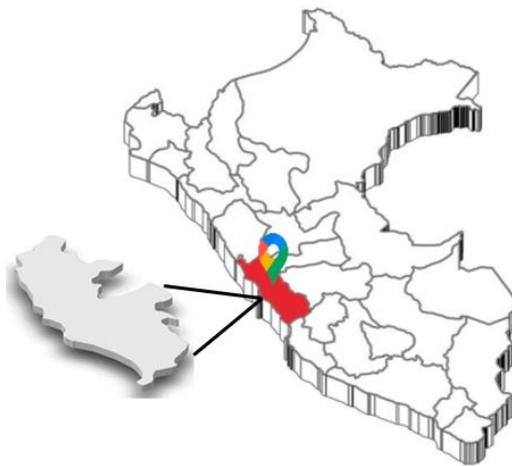


Figura 4.2: Localização geográfica da empresa em estudo

Informação da empresa em análise

A EPL1 conta com duas plantas de produção localizadas estrategicamente, cada uma delas dispõe de um depósito e uma ampla carteira de clientes os quais demanda produtos periodicamente (dia, semana, mês, etc.). A empresa dispõe de uma frota de 90 veículos de capacidade homogênea de 5 toneladas, dos quais 30 veículos são designados para um depósito e 60 para outro. Todos os veículos partem de um armazém, percorrem uma rota e retornam ao depósito de saída, se considera que todos os motoristas só podem trabalhar 8 horas por dia.

Distribuição logística atual da empresa

A EPL1 conta com um analista de distribuição e com dois coordenadores de depósito, os quais gerenciam as distribuições dos produtos para os clientes, criando rotas de despacho para os veículos alocados nos centros de distribuições. Por meio de um sistema ERP de distribuição (ERP do inglês *Enterprise Resource Planning*), o analista e os coordenadores são auxiliados para melhorar as entregas dos produtos na busca de otimizar o envio das mercadorias.

Com a crescente demanda por produtos que a empresa fabrica e comercializa, nos últimos 5 anos tornou-se mais difícil o controle da rede de distribuição das mercadorias, este descontrole trouxe conflitos para os clientes (reclamações por conta da demora da entrega) e outros problemas para a empresa. O problema central refere-se à definição de rotas de distribuição de modo a evitar custos indesejáveis na área de logística.

Problematização para o estudo de caso

Atualmente a empresa não dispõe de uma metodologia sofisticada para decidir quais centros de distribuições atendem quais clientes, esta tarefa é executada conforme a experiência dos gestores locais dos depósitos em coordenação com o analista de distribuição. Isto levou a problemas de gerenciamento operacional, surgindo dificuldades na criação de rotas e problemas de distribuição de mercadorias, além das frequentes reclamações sobre o serviço.

4.1.1 Caracterização da Procura de uma Solução

Com base nos estudos realizados neste trabalho e os dados fornecidos da empresa em análise, o objetivo deste estudo centrou-se em achar uma solução para os problemas de roteamento na rede de distribuição da empresa EPL1, implementando os algoritmos de otimização exata e algoritmos de otimização combinatória, visando diminuir os custos operacionais para a geração das rotas, a distribuição de produtos e dar solução as reclamações das mercadorias atrasadas.

A Figura 4.3 apresenta alguns dados sobre a região metropolitana de Lima e uma ideia da distribuição de 50 clientes escolhidos de forma aleatória para representação visual.

4.2 Desenvolvimentos Metodológicos para o Estudo de Caso

Nesta seção, descreve-se os métodos algorítmicos que foram empregados, como também o passo a passo do desenvolvimento metodológico para resolver o problema de distribuição de mercadorias para a empresa em estudo.

4.2.1 Componentes Sistemáticos Abordados no Estudo

Os componentes sistemáticos são as partes do método que são possíveis quantificar com precisão, nelas podemos incluir as coordenadas geográficas de latitude e longitude dos clientes e dos depósitos, a matriz de distâncias realizada com as localizações dos vértices e a quantidade das rotas para atender as demandas. Para este estudo de caso foram considerados todos os componentes mencionados.

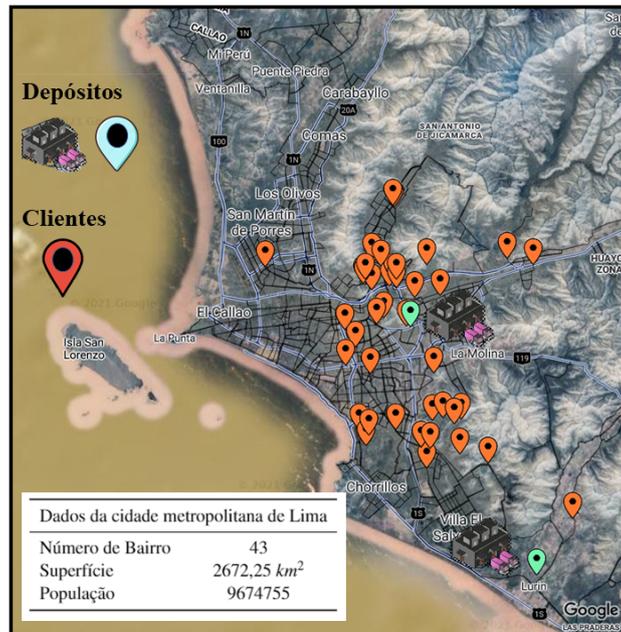


Figura 4.3: Localizações dos depósitos e uma amostra aleatória dos clientes

Localização geográfica dos vértices

No cálculo da matriz de distância, entre os clientes e os depósitos, é importante conhecer as coordenadas geográficas de Latitude ϕ e Longitude λ . A partir dos dados fornecidos pela empresa, acham-se as informações de endereços dos clientes, esses endereços são transformadas em coordenadas geográficas com o auxílio da ferramenta de extensão de Google Maps.

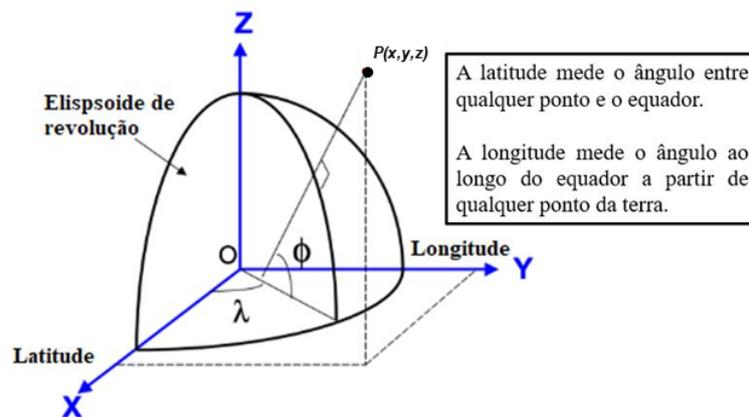


Figura 4.4: Representação das coordenadas geográficas

Elaboração da matriz de distância

Para o cálculo das distâncias dos vértices usou-se a fórmula de Haversine (Robusto 1957), obtendo as distâncias em quilômetro (km). A função é expressa em haversine, sendo definido como:

$$\text{haversine} \left(\frac{D}{R} \right) = \text{haversine} (\phi_1 - \phi_2) + \cos (\phi_1) \cdot \cos (\phi_2) \cdot \text{haversine} (\Delta\lambda)$$

Onde:

$$\text{haversine}(\theta) = \sin^2\left(\frac{\theta}{2}\right).$$

D = distância entre dois pontos.

R = raio da esfera = 6356.755288 Km.

ϕ_i = latitude do ponto i .

λ_i = longitude do ponto i .

$\Delta\phi$ = diferença de latitude = $\phi_1 - \phi_2$.

$\Delta\lambda$ = diferença de longitude = $\lambda_2 - \lambda_1$.

A partir da fórmula anterior temos que distância entre dois pontos é:

$$D = R \times 2 \arcsin\left(\sqrt{\left(\sin^2\left(\frac{\Delta\phi}{2}\right) + \cos\phi_S \cos\phi_f \sin^2\left(\frac{\Delta\lambda}{2}\right)\right)}\right)$$

Embora esta fórmula seja correta para a maioria das distâncias, ela pode ter alguns problemas para um determinado hemisfério. Para resolver este problema, temos as seguintes equações:

$$A = \sin^2\left(\frac{\Delta\phi}{2}\right) + \cos\phi_S \cos\phi_f \sin^2\left(\frac{\Delta\lambda}{2}\right)$$

$$C = 2 \times a \tan 2(\sqrt{A}, \sqrt{1-A})$$

Então a fórmula para achar a distância é: $D = R * C$

4.2.2 Componentes Empíricos Descrito no Estudo

Os componentes empíricos são todos os parâmetros dos métodos que não são medíveis com precisão, nelas temos as estimativas do tempo de engarrafamento em áreas específicas, o tempo médio da viagem entre clientes de diferentes setores, o tempo médio de serviço na entrega e a velocidade média dos veículos. Para este estudo de caso não serão considerados tais parâmetros na resolução do problema com múltiplos depósitos.

4.2.3 Problema de Roteamento com Múltiplos Depósitos Abordado no Estudo

Mediante as informações conhecida na Seção 4.1 com os dados da Tabela 4.1 e as diferentes formas do PRV estudadas no Capítulo 2, definimos o problema com múltiplos depósitos como uma forma de modelar e representar os problemas enfrentados na empresa em análise.

Dados da empresa EPL1		
Número de CD's (2)	Depósito 1	Depósito 2
Número de veículos	60	30
Horas de trabalho por veículo	8h	8h
Capacidade dos veículos	5 Ton	5 Ton
Número máximo de clientes por rota	80	80
Média de clientes atendidos por dia	4250	2280

Tabela 4.1: Informações da empresa em análise para o estudo de caso

Após representar a variável do problema com múltiplos depósitos, precisamos usar o modelo matemático, além dos algoritmos de Programação Linear Inteira e o método de *Branch and Bound* na obtenção de uma solução por meio de um solver comercial. A formulação do modelo empregado foi retirado do trabalho de Kulkarni & Bhave (1985), onde os autores apresentam uma descrição sequencial de modelos matemá-

ticos, iniciando com PCV (descrito neste trabalho na Seção 2.1), em seguida o PRVC (Seção 2.3.1), até chegar no modelo PRVMD que foi usado para este estudo e está descrito na Seção 2.3.8.

4.2.4 Algoritmo Exato na Procura de uma Solução

Conhecendo a origem da formulação matemática do PRVMD proposto por Kulkarni & Bhawe (1985) estudada na Seção 2.3.8, os parâmetros considerados na Seção 4.2.1 e por meio do uso de solver comercial Gurobi dentro da linguagem de programação Julia, com a finalidade de gerar solução para os problemas de distribuição de produtos e gerações de rotas. Com a informação disponibilizada do departamento de distribuição da empresa EPL1, que presta serviços de atendimento das demandas dos clientes, formou-se a Tabela 4.2 a qual fornece informações das quantidades das demandas e dos clientes atendidos diariamente nos dias 16 até 25 de junho de 2021. As 10 instâncias revelam de forma real a quantidade de clientes e a totalidade das demandas diárias a serem atendidas.

Instância	Dia	Quantidade de clientes	Demanda total (Ton)
1	16/06/2021	6804	300.41
2	17/06/2021	6929	307.51
3	18/06/2021	6413	296.14
4	19/06/2021	6357	246.68
5	20/06/2021	6061	179.09
6	21/06/2021	1038	118.40
7	22/06/2021	6835	431.48
8	23/06/2021	6431	358.84
9	24/06/2021	6293	304.65
10	25/06/2021	6469	239.32

Tabela 4.2: Serviço de distribuição desde o dia 16 até 25 de junho da empresa em análise

Implementação do algoritmo exato para o PRVMD

A realização de alguns testes computacionais para algumas instâncias na implementação do PRVMD foram importantes para validar o modelo e analisar o comportamento dos resultados conforme são acrescentados mais pontos ao problema. A criação das instâncias foi feita com o analisada a Tabela 4.2, especificamente o dado do dia 16 de junho onde se atenderam 300.41 toneladas de produtos para 6804 clientes os quais fizeram seus pedidos com 1 semana de antecedência. Dessa população de clientes, extraímos de forma aleatória instâncias de 50, 100, 150 e 200 vértices, repetimos a operação de aleatoriedade mais duas vezes para cada instância assim como se visualiza na Tabela 4.3. A repetição da aleatoriedade teve o objetivo de compreender melhor o comportamento das distâncias totais e os tempos investido na solução das instâncias para o modelo multi depósito.

Criação das instâncias para os testes da metodologia

As instâncias da Tabela 4.3 foram geradas com base no banco de dados da empresa, de tal forma que cada instância testada foi escolhida por meio de uma amostra de informações escolhida aleatoriamente neste banco de dados, na tentativa de reproduzir os dados reais da empresa. Os cálculos de alguns parâmetros,

como a quantidade e capacidade dos veículos a serem usados no desenvolvimento dos métodos foram definidos como segue.

Parâmetros das instâncias criadas para os testes computacionais				
Instância	Número de clientes	Demanda (Ton)	Número de veículos	Capacidade (Ton)
50a	50	1.18	13	0.09
50b	50	2.09	5	0.56
50c	50	6.25	3	3.22
100a	100	7.37	3	3.72
100b	100	12.25	4	3.87
100c	100	2.52	14	0.19
150a	150	11.10	4	3.84
150b	150	5.42	9	0.66
150c	150	5.53	11	0.57
200a	200	7.68	20	0.42
200b	200	6.25	9	0.78
200c	200	6.43	8	0.69

Tabela 4.3: Instâncias criadas para os testes da metodologia

- Quantidade máxima de clientes designados aos depósitos na formação dos *clusters*. Seja n o número de clientes em cada instância, o cálculo da quantidade máxima de clientes designados para o primeiro depósito é $(\frac{2}{3}n)$ e para o segundo depósito é $(\frac{1}{3}n)$.

Para a instância 50a foram selecionado 50 clientes de forma aleatória onde foram designados como máximo $(\frac{2}{3}50) = 33.3 \approx 33$ para um depósito e $(\frac{1}{3}50) = 16.6 \approx 17$ para u outro.

- Cálculo da capacidade máxima dos veículos para cada instância. Entre os n clientes, procure-se achar a maior demanda existente (DM). Este valor é a capacidade máxima de cada veículo em carga.

A instância 50a conta com uma demanda total de 1.18 toneladas, nessa população a maior demanda realizada por um cliente foi de 0.09 toneladas de mercadoria, esse último número indica a capacidade máxima do veículo.

- Cálculo da quantidade de veículos no primeiro depósito. Conhecendo a demanda total dos clientes designados para o primeiro depósito (DT_1), podemos calcular a quantidade de veículos requeridos no atendimento (V_{DT_1}), esta expressão é definida como $V_{DT_1} = \frac{DT_1}{DM}$.

Na instância 50a, o depósito 1 conta com uma população máxima de 33 clientes designados, onde a demanda total deles é 0.88 toneladas e a demanda máxima de um clientes dessa instância é de 0.09 toneladas, então podemos calcular a quantidade de veículos para o depósito 1, este cálculo é igual a $\frac{0.88}{0.09} = 9.8 \approx 10$.

- Cálculo da quantidade de veículos no segundo depósito. Este calculo é realizada com os dados da demanda total dos clientes designados para o segundo depósito (DT_2) e a demanda máxima da instância (DM), portanto $V_{DT_2} = \frac{DT_2}{DM}$.

Na instância 50a, o depósito 2 conta com uma população de 17 clientes designados, onde a demanda total deles é 0.30 toneladas e a demanda máxima de um clientes dessa instância é de 0.09 toneladas, então podemos calcular a quantidade de veículos para o depósito 1, este cálculo é igual a $\frac{0.30}{0.09} = 3.4 \approx 3$.

- Número total dos veículos a serem usados nas instâncias. Este cálculo é realizado somando as quantidades de veículos do primeiro e segundo depósito V_{DT_1} e V_{DT_2} .

Para a instância 50a o valor de V_{DT_1} e V_{DT_2} são 10 e 3 respetivamente, então o total de veículos para essa instância é 13.

- Capacidade máxima de visitas dos veículos por rota. Este cálculo é produzido da quantidade de clientes por cada instância dividido pela quantidade de veículos totais: $\frac{n}{V_{DT_1} + V_{DT_2}} + 1$.

Para a instância 50a conta-se com 50 clientes e um total de 13 veículos, porem a quantidade máxima de clientes atendido por rota é $\frac{50}{10+3} + 1 = 4.84 \approx 5$.

Para cada instância da Tabela 4.3 se introduziram dois vértices adicionais que representam os depósitos, além disso, nesta tabela está presente as demandas por cada instância, a quantidade de veículos e as capacidades destes. Para o cálculo da quantidade e capacidade dos veículos foi realizado uma operação de proporcionalidade referente à demanda total, às quantidades e às capacidades dos veículos no dia 16 de Junho, baseado nos dados reais da Tabela 4.2. Uma vez criadas as instâncias e os parâmetros da Tabela 4.3, utilizou-se o modelo e os algoritmos propostos para este estudo, cujo os resultados computacionais aparecem na sequência.

4.2.5 Algoritmo Heurístico na Procura de uma Solução

Após ter implementado e aplicado o algoritmo exato, via solver, para o PRVMD e não conseguir soluções ótimas em um tempo computacional de 30 minutos para as instâncias 50, 100, 150 e 200, surgiu a necessidade de implementar um método aproximado de solução que consiga reduzir o tempo computacional. Das múltiplas heurísticas estudadas no Capítulo 3, foi escolhida aquela que conta com duas fases, especificamente a heurística que primeiro agrupa e depois roteia.

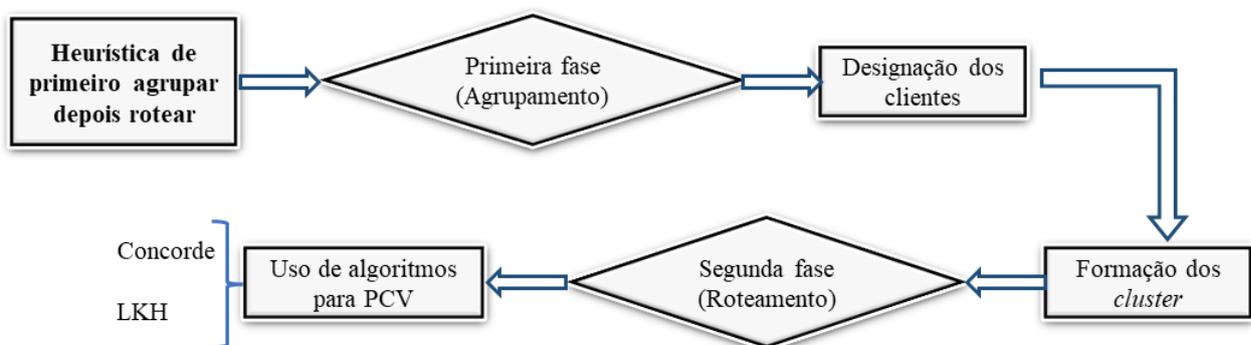


Figura 4.5: Representação da metodologia heurística

As instâncias e os parâmetros usados no desenvolvimento da heurística foram os mesmos usados para o método exato, fornecidos da Tabela 4.3. A Figura 4.5 apresenta um esquema metodológico da heurística de

duas fases implementada para o estudo de caso. Na primeira fase serão designados os clientes aos centros de distribuição formando dois *clusters*, cada um deles será visto como um problema de roteamento de veículo capacitado com um único depósito. Para cada depósito com os seus clientes designados, se formarão grupos independentes. Na segunda fase foi implementado e resolvido o roteamento dos grupos usando o modelo matemático visto na Seção 2.1 e o solver comercial Gurobi executado na linguagem de programação Julia. Além disso, usou-se dois algoritmos especializados para o PCV (Concorde e LKH), como uma forma de avaliar e tentar melhorar os resultados para as instâncias testadas. O Concorde resolve o PCV de forma exata, enquanto que o LKH é um método heurístico construtivo.

4.2.6 Primeira Fase da Heurística

Na primeira fase do desenvolvimento da heurística (agrupamento ou *clustering*) usou-se uma versão de agrupamento não supervisionado e particionado *k-means*, este algoritmo foi implementado na linguagem de programação Julia e testados para as instâncias da Tabela 4.3. Os depósitos foram considerados como centros dos *clusters* para a formação dos grupos, de modo a minimizar a soma das distâncias cliente-depósito enquanto satisfaz a restrição de capacidade e demanda. Designamos então, quantidades de clientes para cada depósito como se apresenta na Figura 4.6, dessa forma o PRVMD pode ser visto como dois PRVCs independentes tendo, cada um deles, um único depósito.

Algoritmo *k-means* para as formações dos *clusters*

O desenvolvimento do algoritmo *k-means* neste estudo de caso, foi baseado no modelo matemático estudado no trabalho de Seref et al. (2014) a qual foi descrito na seção 3.4.4. Este algoritmo começa designando todos os clientes à um dos depósitos, formando assim dois *clusters*. Os depósitos serão considerados como os centros dos *clusters* inalteráveis. Assim, após finalizar a designação dos clientes apenas uma vez, o primeiro processo de *clustering* é concluído, onde o PRVMD passará a ser dois PRVC independentes.

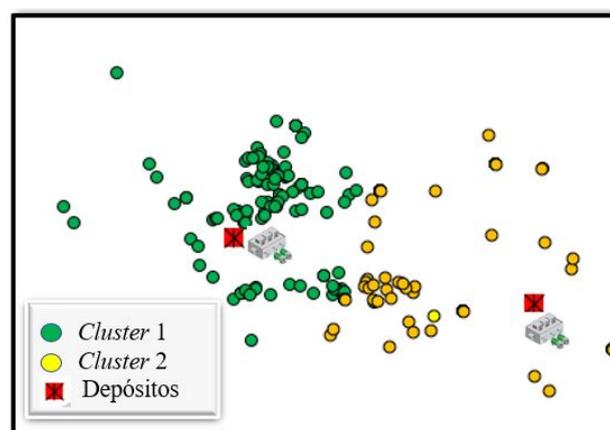


Figura 4.6: Representação da primeira fase da heurística – *Clustering*

A adaptação do modelo do algoritmo *k-means* foi realizada considerando as quantidades de veículo disponibilizados em cada depósito, o número de depósitos, estas informações trazida da Tabela 4.1, e da informação das quantidades de clientes por depósito apresentado na Seção 4.2.4. Esta adaptação surgiu pela necessidade de designar de forma proporcional os clientes aos depósitos, por exemplo para a instância

de 50a clientes e, implementando o modelo algorítmico *k-means* formulado por (Seref et al. 2014), o 90% dos clientes foram designado ao primeiro depósito, deixando apenas 5 para o segundo, gerando distorções nas quantidades de veículos designados aos depósitos. Porém as restrições 4.3 e 4.4 do modelo adaptado serviram para restringir a quantidade máxima de clientes designado para cada *cluster* considerando uma determinada frota de veículos, dessa forma a alocação dos clientes aos depósitos para a instância 50a foram 33 para o primeiro depósito e 17 para o segundo. Em continuação apresentamos o modelo matemático adaptado para o nosso estudo, que uma versão orientada a modelo do método *k-means*.

$$\text{Minimizar} \quad \sum_{i=1}^n \sum_{j=1}^n \delta_{ij} z_{ij} \quad (4.1)$$

$$\text{Sujeito a} \quad \sum_{i=n-1}^n z_{ij} = 1, \quad j = 1, \dots, n, \quad (4.2)$$

$$\sum_{j=1}^n z_{n-1j} \leq \left(\frac{2}{3}n\right), \quad (4.3)$$

$$\sum_{j=1}^n z_{nj} \leq \left(\frac{1}{3}n\right), \quad (4.4)$$

$$\sum_{j=1}^n z_{ij} \leq 0, \quad i = 1, \dots, n-2, \quad (4.5)$$

$$z_{ij} \in \{0, 1\}, \quad i = n-1, n, \quad j = 1, \dots, n. \quad (4.6)$$

A função objetivo trata de minimizar a soma das distâncias desde os vértices até os centros dos *clusters*. Para este caso os centros dos *clusters* é conhecido a “*priori*” as quais são representados pelos vértices dos depósitos $n-1$ e n . A restrição (4.2) garante que cada objeto seja atribuído exatamente para um centro de *cluster*. Considerado a totalidade dos veículos (90) e sabendo que no depósito 1 estão alocados 60 veículos e para o depósito 2 estão alocados 30 veículos, foram criados as restrições (4.3) e (4.4). A restrição (4.3) indica que no máximo $\frac{2}{3}$ do total de clientes das instâncias são atribuídos ao vértice $n-1$ (este valor é arredondado ao valor inteiro mais próximo), representado pelo depósito 1. A restrição (4.4) indica que no máximo $\frac{1}{3}$ do total de clientes são atribuídos ao vértice n (este valor é arredondado ao valor inteiro mais próximo), representado pelo depósito 2.

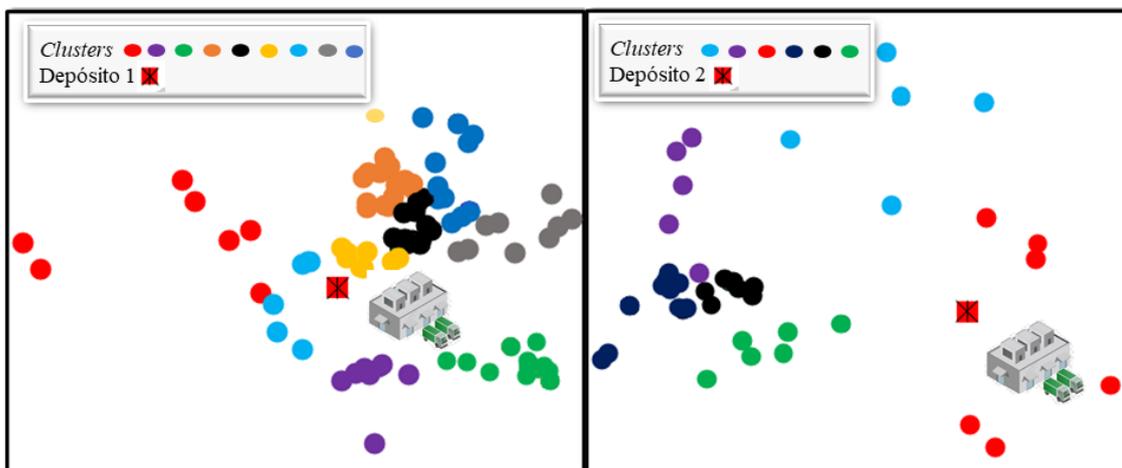


Figura 4.7: Designação dos clientes e formação dos *clusters*

Após a designação dos clientes como ilustrado na Figura 4.6, o PRVMD é transformado em dois PRVCs mono depósito. Então, para cada PRVC, são gerados novos *clusters* utilizando a distância como métrica de similaridade. Para isso, foi resolvido por meio do solver Gurobi o modelo estudado na Seção 3.4.4 (versão do método *k-means* baseado em modelo matemático). Um exemplo desses novos *clusters* é ilustrado na Figura 4.7, onde cada uma destes novos *clusters* inclui um depósito particular na formação de uma das rotas do PRVC.

4.2.7 Segunda Fase da Heurística

A segunda fase da heurística consiste no roteamento, onde as rotas para os novos *clusters* são criados, percorrendo dessa forma todos os pontos, a Figura 4.8 ilustra a segunda fase da heurística. O método para definir o percurso nos vértices está baseado nos algoritmos para o problema do caixeiro viajante. O modelo matemático descrito na Seção 2.1 desenvolvida por Kulkarni & Bhawe (1985) foi resolvido por meio do solver Gurobi na segunda fase da heurística.

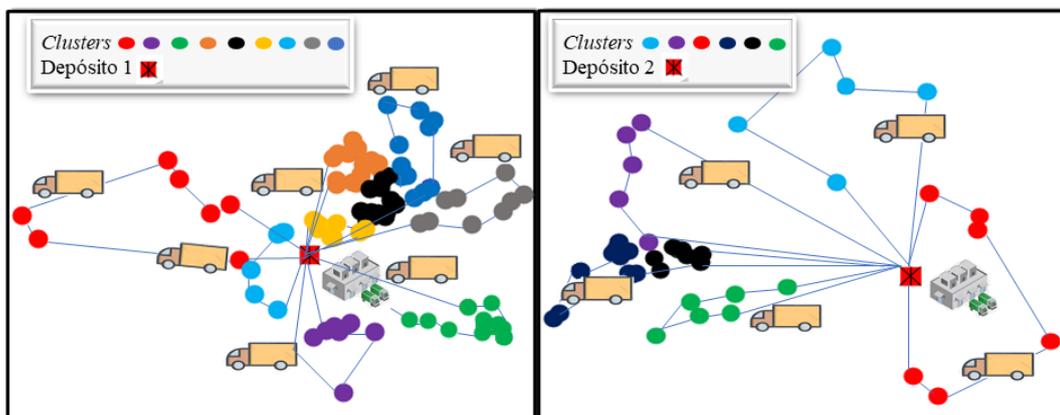


Figura 4.8: Representação da segunda fase da heurística – *Routing*

Uso de outros algoritmos para o roteamento dos *clusters*

Além de resolver a fase de roteamento com o algoritmo exato para o PCV, foi usado outros dois algoritmos que conseguiram resolver a segunda fase da heurística. Estes dois algoritmos serviram para comparar os custos e os tempos computacionais na procura de uma solução. Portanto, para o roteamento dos *clusters* foram usados os algoritmos Concorde e LKH, estes algoritmos são especializado para resolver os problemas do PCV de forma exata gerando boas soluções Matai et al. (2010), (para obter o algoritmo Concorde pode visitar o link (<https://www.math.uwaterloo.ca/tsp/concorde.html>) e LKH (para obter o algoritmo LKH pode visitar o link (<http://webhotel4.ruc.dk/~keld/research/LKH/>), que foram fornecidos pela plataforma de repositórios de solver online GitHub.

4.3 Ferramentas Utilizadas

4.3.1 Linguagem de Programação Julia

O desenvolvimento, a implementação e os testes computacionais da metodologia proposta para este estudo de caso foi através da linguagem de programação Julia e a linguagem de modelagem JuMP que é

uma coleção de pacotes de suporte para otimização matemática em Julia. O JuMP facilita a formulação e resolução de uma variedade de classes de problemas, incluindo programas lineares, programas inteiros, programas cônicos, programas semi-definidos e programas não lineares restritos. O Julia é uma linguagem de programação que busca melhorar o desempenho das linguagens Python e C+, em relação à produtividade e eficiência respectivamente. Além disso, essa linguagem tem sido frequentemente utilizada no desenvolvimento de disciplinas e pesquisas, também possui diversos artigos e publicações científicas que a descreve (Castellucci 2017).

4.3.2 Informações do Computador

Para a implementação dos algoritmos e os testes computacionais usou-se um computador modelo ACER Nitro 5 com processador Intel(R) Core(TM) i7-9750H CPU @ 2.60GHz 2.59 GHz com memória RAM de 8,00 GB (utilizável: 7,84 GB), com sistema operacional Windows 10 de 64 bits e um processador baseado em 64x.

4.4 Testes Computacionais e Análise dos Resultados

Esta seção apresenta os resultados obtidos usando as metodologias propostas para o estudo de caso. Primeiro, apresentam-se os resultados da metodologia exata a qual foi descrita na Seção 4.2.4, posteriormente, apresentam-se os resultados da heurística que primeiro agrupa e depois roteia, descrita na Seção 4.2.5, os quais serão analisados e interpretados, e para finalizar compara-se analisado os resultados das duas metodologias propostas.

4.4.1 Resultados e Análise dos Testes Computacionais para a Metodologia Exata

A Tabela 4.4 apresenta os resultados dos testes computacionais da metodologia exata para o PRVMD, usando um tempo máximo de 30 minutos. A tabela revela informações sobre as quantidades de veículos a serem usados em cada depósito e para cada instância, as capacidades e visitas dos veículo, os custos mínimos achados pelo algoritmo, os tempos na busca de uma solução em segundos, os números de ramos gerados na árvore de busca e as porcentagens dos GAPs.

Os custos totais, em termo de distâncias, para percorrer as rotas das instâncias 50a, 50b, 50c, 100a, 100b, 100c, 150a, 150b, 150c, 200a, 200b, e 200c são: 1.18, 2.09, 6.25, 7.37, 12.25, 2.52, 4.26, 6.64, 5.88, 11.64 e $9.60 \times 10^3 km$, respectivamente. Note-se que os resultados obtidos não expressam as soluções ótimas, isso foi concluído visualizando os GAP elevados. Qualquer diferença de valores que fornece uma medida do quão longe estamos do valor ótimo da solução é chamada de GAP de otimalidade, quer dizer, a tolerância absoluta obtida pela diferença entre o melhor nó inteiro e o valor da função objetiva dos melhores nós restantes, quanto menor seja esse valor mais próximo fica da solução ótima. O melhor GAP fornecido nos testes computacionais foi de 9.77 % onde o número de ramos ou nós formados na árvore de busca foi de 17824, da instância 50c. O pior GAP gerado foi de 87.25 % para a instância 150b onde se formaram 47 ramos.

Não foi possível a obtenção de uma solução na instância 200c, para um tempo de 30 minutos, isso pode ser causado pelo pouco tempo computacional, mas infere-se que é produto da complexidade das instâncias

para o computador usado. Analisando os GAP de todas as instâncias e, sabendo que os números de clientes para o caso real da empresa EPL1 ultrapassam os 6 mil por dia, podemos concluir que a metodologia proposta não fornecerá uma solução ótima em um tempo aceitável.

	Exemplar	50a	50b	50c	100a	100b	100c
Número de veículos	Total	11	5	3	3	4	12
	D1	8	4	2	2	3	8
	D2	3	1	1	1	1	4
Demanda total (Ton)		1.18	2.09	6.25	7.37	12.25	2.52
Capacidade máxima	Carga (Ton)	0.09	0.56	3.22	3.72	3.87	0.19
	Visitas	5	12	19	35	27	9
Distância (10^3 km)	Total	1.93	1.28	1.57	2.33	2.48	4.89
	D1	1.61	1.28	0.82	1.38	1.64	3.53
	D2	0.31	0.00	0.75	0.95	0.84	1.36
Tempo(s)		1800	1800	1800	1800	1800	1800
Número de nós		59535	81781	179824	24329	10207	702
Gap(%)		60.51	31.55	9.77	43.05	43.77	45.56

	Exemplar	150a	150b	150c	200a	200b	200c
Número de veículos	Total	4	9	11	19	9	–
	D1	2	6	8	14	6	–
	D2	2	3	3	5	3	–
Demanda total (Ton)		11.10	5.42	5.53	7.68	6.25	–
Capacidade máxima	Carga (Ton)	3.84	0.66	0.57	0.42	0.78	–
	Visitas	39	18	15	12	24	–
Distância (10^3 km)	Total	4.26	6.64	5.88	11.64	9.60	–
	D1	1.61	4.50	4.17	8.32	6.57	–
	D2	2.65	2.14	1.71	3.32	3.03	–
Tempo(s)		1800	1800	1800	1800	1800	–
Número de nós		5802	47	1232	0	1	–
Gap(%)		66.47	87.25	76.14	87.02	86.73	–

Tabela 4.4: Resultados computacionais obtidos via solver para 50, 100, 150 e 200 vértices

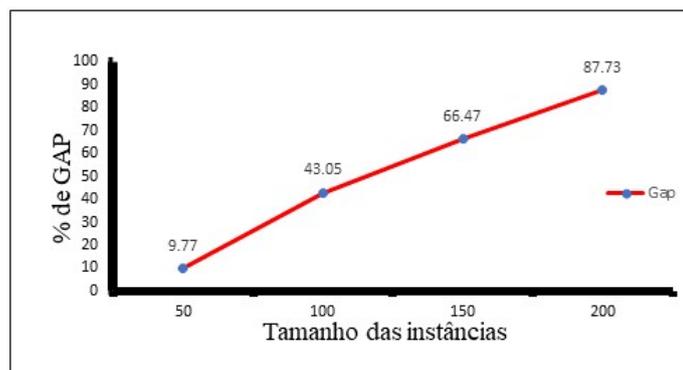


Figura 4.9: GAP das instâncias testadas via solver

A Figura 4.9 foi criada empregando os GAP de menor valor das instâncias de 50, 100, 150, e 200. Analisando os GAP, especialmente da instância 200, podemos inferir que quando cresce o número de vértices (clientes) o GAP também aumenta ou pode ser que o GAP não apareça como aconteceu na instância 200c, então para instâncias com grande número de vértices será difícil achar a solução ótima em um tempo de 30 minutos. Porém, requerem-se outros algoritmos ou métodos que consigam fornecer melhores soluções do que as achadas na Tabela 4.4 dentro do tempo estabelecido.

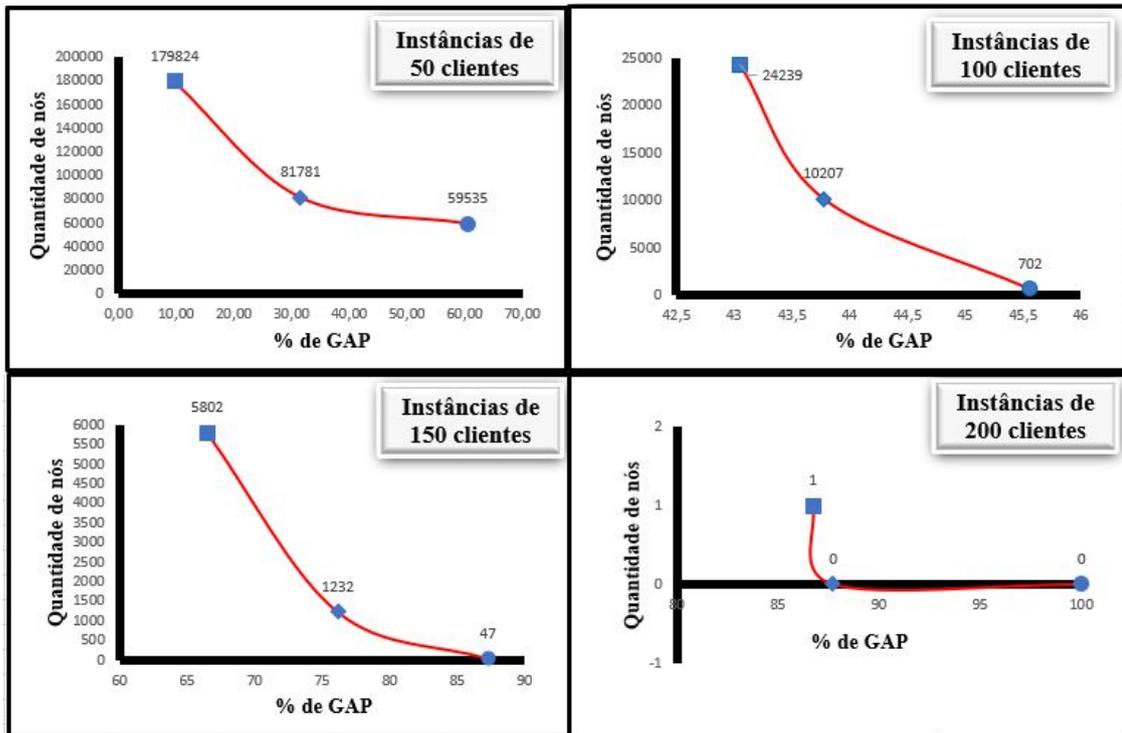


Figura 4.10: Comportamento dos GAPs das instâncias testadas

A Figura 4.10 foi criada para visualizar a quantidade de ramificações que foram feitas pelo algoritmo na procura de uma solução inteira próxima do ótimo para um tempo de 30 minutos. Para as instâncias onde se têm 50 vértices, observe-se que quanto mais ramificações são criadas a porcentagem de GAP diminui, esse comportamento acontece de forma similar para as instâncias de 100, 150 e 200 vértices. Observe-se também, que o número de ramificações diminui quando o número de clientes aumenta, então podemos inferir para este trabalho, que quanto mais ramificações são criadas a solução tende a se aproximar do ótimo.

4.4.2 Resultados e Análise dos Testes Computacionais para a Metodologia Heurística

Resultados da heurística usando algoritmo exato na segunda fase

As Tabelas 4.5 e 4.6 apresentam os resultados dos testes computacionais fornecidos pela heurística desenvolvida para o PRVMD, usando um tempo máximo de 30 minutos. Na segunda fase da heurística (roteamento) implemento-se e usou-se o algoritmo exato para o PCV segundo o modelo desenvolvido no trabalho de Kulkarni & Bhawe (1985) descrito na Seção 2.1. As tabelas revelam informações sobre as demandas totais para as instâncias em toneladas, as quantidades de veículos a serem usados em cada depósito e para cada instância, as capacidades dos veículos em toneladas, as visitas máximas para cada veículo, os custos mínimos achados pelo algoritmo e os tempos usados na busca de uma solução em segundos.

Na Tabela 4.5 pode ser observada duas análises para as instâncias de 50 e 100 vértices, a primeira análise diferencia-se da segunda pelas quantidades de mais um veículo a ser usado em cada depósito, quer dizer, para a instância 50a da primeira análise conta-se com 13 veículos a serem usados e dois depósitos, na segunda análise da mesma instância conta-se com mais um veículo por depósito (15 veículos no total). Essa escolha na quantidade de veículos foi com a intenção de diminuir os custos e os tempos na procura das soluções.

		Exemplar	50a	50b	50c	100a	100b	100c
Número de veículos	Total		13	5	3	3	4	14
	D1		10	4	2	2	3	9
	D2		3	1	1	1	1	5
		Demanda total (Ton)	1.18	2.09	6.25	7.37	12.25	2.52
Capacidade máxima	Carga (Ton)		0.09	0.56	3.22	3.72	3.87	0.19
	Visitas		5	12	19	35	27	9
Distância (10^3 km)	Total		2.58	1.49	1.81	1.85	2.45	4.91
	D1		1.62	0.95	1.12	1.27	1.91	3.58
	D2		0.96	0.54	0.69	0.58	0.54	1.33
		Tempo(s)	31	14.2	2.3	917	488	95.2
Adição de um veículo por depósito								
Número de veículos	Total		15	7	5	5	6	16
	D1		11	5	3	3	4	10
	D2		4	2	2	2	2	6
		Demanda total (Ton)	1.18	2.09	6.25	7.37	12.25	2.52
Capacidade máxima	Carga (Ton)		0.09	0.56	3.22	3.72	3.87	0.19
	Visitas		5	12	19	35	27	9
Distância (10^3 km)	Total		2.65	1.69	2.16	2.11	2.45	5.09
	D1		1.62	0.94	1.24	1.30	1.69	3.67
	D2		1.03	0.76	0.92	0.80	0.76	1.42
		Tempo(s)	1.0	1.0	1.0	602	296.8	95.0

Tabela 4.5: Resultados computacionais obtidos via heurística para 50 e 100 vértices

Observa-se nas Tabelas 4.5 e 4.6 os diferentes custos de todas as instâncias em termos de distância. Para a instância 50a da primeira análise se teve uma distância de $2.58 \times 10^3 km$ no percurso de todos os vértices. Esta distância é diferente a instância 50b ($1.49 \times 10^3 km$) porque as localizações dos clientes para a instância 50a são mais longas entre eles e com os depósitos do que os clientes da instância 50b. Porém, temos distâncias diferentes para cada instância, além disso observa-se para algumas instâncias, uma maior distâncias percorridas com menores números de clientes. Por exemplo o percurso na instâncias 50a ($2.58 \times 10^3 km$) é maior do que a instâncias 100a ($1.81 \times 10^3 km$) porque os clientes da instância 100a estão mais próximas aos depósitos e entre eles.

A Tabela 4.6 apresenta os resultados das instâncias de 150 e 200 como foi feita na tabela 4.5, quer dizer, na instância 150a da primeira análise, conta-se com 4 veículos a serem usados e, na segunda análise da mesma instância conta-se com 6 veículos no total.

		Exemplar	150a	150b	150c	200a	200b	200c
Número de veículos	Total		4	9	11	20	9	8
	D1		2	6	8	15	6	6
	D2		2	3	3	5	3	2
Demanda total (Ton)			11.10	5.42	5.53	7.68	6.25	6.43
Capacidade máxima	Carga (Ton)		3.84	0.66	0.57	0.42	0.78	1.01
	Visitas		39	18	15	12	24	26
Distância (10 ³ km)	Total		1.95	2.72	4.03	4.97	2.72	2.62
	D1		1.27	1.72	2.38	2.92	1.51	1.73
	D2		0.68	1.00	1.65	2.05	1.21	0.90
Tempo(s)			942.4	234.9	381.5	103.7	104.4	970.2

Adição de um veículo por depósito									
Número de veículos	Total			6	11	13	22	11	10
	D1			3	7	9	16	7	7
	D2			3	4	4	6	4	3
Demanda total (Ton)			11.10	5.42	5.53	7.68	6.25	6.43	
Capacidade máxima	Carga (Ton)			3.84	0.66	0.57	0.42	0.78	1.01
	Visitas			39	18	15	12	24	26
Distância (10 ³ km)	Total			2.41	2.80	4.50	5.04	2.97	2.98
	D1			1.52	1.72	2.57	3.02	1.68	1.86
	D2			0.89	1.08	1.92	2.02	1.29	1.12
Tempo(s)			492.4	39.3	371.1	103.1	31.1	765.4	

Tabela 4.6: Resultados computacionais obtidos via heurística para 150 e 200 vértices

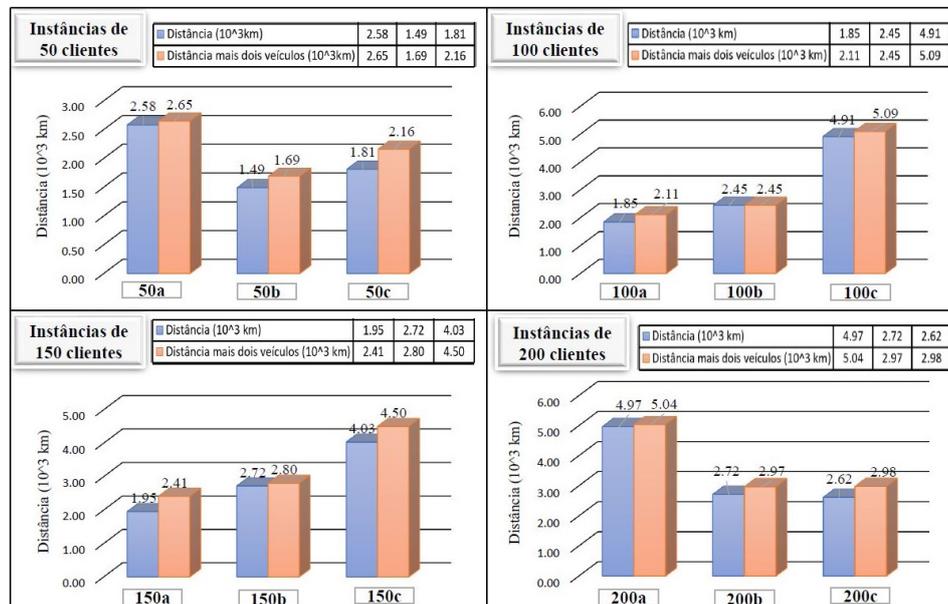


Figura 4.11: Análise dos custos obtidos via heurística para 50, 100, 150 e 200 vértices

A Figura 4.11 apresenta os resultados de todas as instâncias fornecidas pelas Tabelas 4.5 e 4.6. Observa-se que o custo em termo de distância das instâncias com 50 vértices da primeira análise e os custos das mesmas instâncias adicionando um veículo por depósito na segunda análise, são diferentes. Por exemplo, a distância da instância 50a da primeira análise foi de $2.58 \times 10^3 km$ e para a segunda análise da mesma instância com mais dois veículos foi de $2.65 \times 10^3 km$, esse crescimento acontece para todas as instâncias do problema. Assim para a instância de 200b a distância na primeira análise foi de $2.72 \times 10^3 km$ e para a segunda análise foi de $2.79 \times 10^3 km$. Analisando os resultados das distâncias com as quantidades de veículos disponíveis no depósito, podemos concluir para este estudo de caso que eles aumentam quanto mais veículos forem usados.

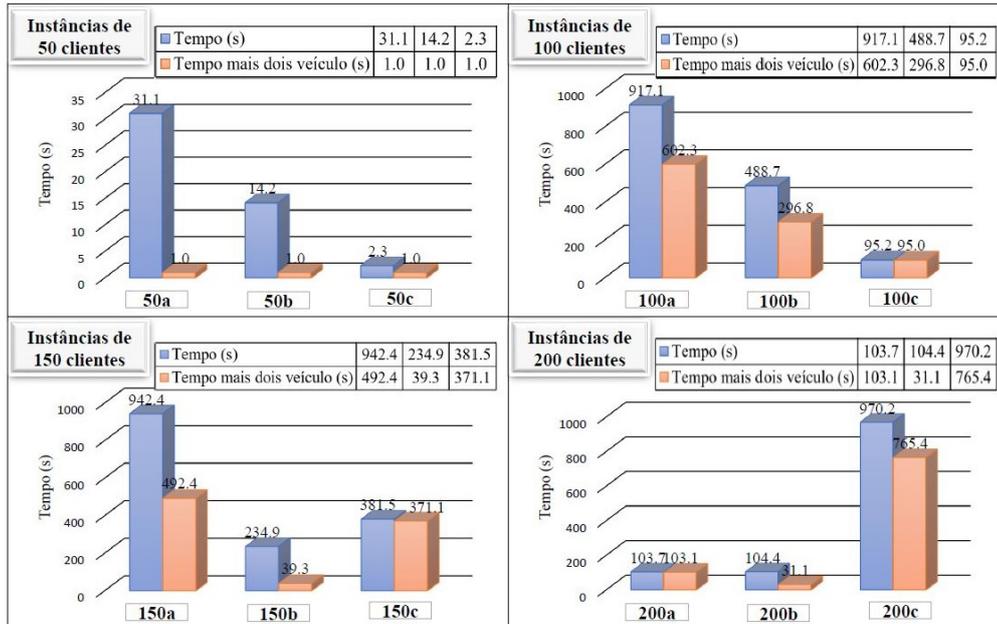


Figura 4.12: Análise dos tempos obtidos via heurística para 50, 100, 150 e 200 vértices

A Figura 4.12 ilustra os tempos usados pela heurística para a procura das soluções. Os tempos investidos em resolver o PRVMD são diferentes em cada instância, também observa-se irregularidade nos tempos para algumas instâncias, por exemplo para as instâncias 100a (917s) e 200a (103.1s), normalmente a primeira precisaria de menos tempo investido do que a instância com 200 clientes, mas nos testes revelaram que para a instância com 100a o tempo em resolver o PRVMD foi maior do que a instância 200a, isso é por que os clientes da instância 100a conta com parâmetros diferente do que da instância 200a, além disso estão alocados em zonas mais distantes dos depósitos e entre eles, porém na primeira fase da heurística vai ser mais demorado formar os *clusters* e designar os clientes aos depósitos.

Observa-se também que os tempos para as instâncias com 50 vértices da primeira análise, quanto para as mesmas instâncias acrescentando mais dois veículos da segunda análise, diferem entre eles. Por exemplo, a instância 50a da primeira análise usou 31.1 segundos para resolver o problema e o tempo usado na segunda análise para a mesma instância foi de 1 segundo, essa queda de tempo acontece para todas as instâncias testadas, assim para a instância 150b o tempo na primeira análise foi de 234.9 segundos e o tempo que demorou a segunda análise da mesma instância em resolver o problema foi de 39.3 segundos. Então, podemos concluir neste estudo de caso, que o tempo gasto pela heurística em resolver o PRVMD diminui quanto mais veículos são disponibilizados, embora gere um aumento no custo.

Resultados usando o algoritmo Concorde na segunda fase da heurística

As Tabelas 4.7 e 4.8 apresentam os resultados dos testes computacionais fornecidos pela heurística de duas fases desenvolvida para o PRVMD, usando um tempo máximo de 30 minutos. Para a segunda fase da heurística (roteamento) usou-se o algoritmo Concorde que é um algoritmo especializado para o PCV, este algoritmo não foi retirado do GitHub e trazido ao ambiente Julia como um solver. As tabelas revelam informações sobre as demandas totais para as instâncias em toneladas, as quantidades de veículos a serem usados em cada depósito e para cada instância, as capacidades dos veículos em toneladas, as visitas máximas para cada veículo, os custos mínimos achados pelo algoritmo e os tempos usados na busca de uma solução em segundos.

Na Tabela 4.7 pode ser observada duas análises para as instâncias de 50 e 100 clientes, a primeira análise diferencia-se da segunda pelas quantidades de mais um veículo a ser usado em cada depósito, quer dizer, para a instância 50c da primeira análise conta-se com 3 veículos a serem usados e dois depósitos, na segunda análise da mesma instância conta-se com mais um veículo por depósitos, no qual faz um total de 5 veículos. Esse crescimento das quantidades de veículos foram feitos para diminuir os custos, os tempos na procura das soluções e também para formar mais rotas.

	Exemplar	50a	50b	50c	100a	100b	100c
Número de veículos	Total	13	5	3	3	4	14
	D1	10	4	2	2	3	9
	D2	3	1	1	1	1	5
Quantidade	Demanda total (Ton)	1.18	2.09	6.25	7.37	12.25	2.52
	Carga (Ton)	0.09	0.56	3.22	3.72	3.87	0.19
	Visitas	8	12	19	35	27	9
Distância (10 ³ km)	Total	2.6	1.51	1.84	1.87	2.45	4.96
	D1	1.63	0.97	1.14	1.28	1.89	3.62
	D2	0.97	0.54	0.7	0.59	0.56	1.34
	Tempo(s)	8.0	2.2	11.0	6.1	11.3	443.7
Adição de um veículo por depósito							
Número de veículos	Total	15	7	5	5	6	16
	D1	11	5	3	3	4	10
	D2	4	2	2	2	2	6
Quantidade	Demanda total (Ton)	1.18	2.09	6.25	7.37	12.25	2.52
	Carga (Ton)	0.09	0.56	3.22	3.72	3.87	0.19
	Visitas	8	12	19	35	27	9
Distância (10 ³ km)	Total	2.67	1.72	2.18	2.15	2.49	5.15
	D1	1.64	0.96	1.25	1.33	1.72	3.71
	D2	1.03	0.76	0.93	0.82	0.77	1.44
	Tempo(s)	2.0	1.0	0.7	8.1	9.3	443.2

Tabela 4.7: Resultados computacionais obtidos via algoritmo Concorde para 50 e 100 vértices

Na Tabela 4.8 apresenta-se os resultados fornecidos pela heurística das instâncias de 150 e 200, com as

mesmas análises da Tabela 4.7. Para a instância 150a da primeira análise conta-se com 4 veículos a serem usados gerando assim uma distância de $2.0 \times 10^3 km$ e, na segunda análise da mesma instância pode ser observado 6 veículos no total gerando uma nova distância de $2.47 \times 10^3 km$.

		Exemplar	150a	150b	150c	200a	200b	200c
Número de veículos	Total		4	9	11	20	9	8
	D1		2	6	8	15	6	6
	D2		2	3	3	5	3	2
		Demanda total (Ton)	11.10	5.42	5.53	7.68	6.25	6.43
Quantidade	Carga (Ton)		3.84	0.66	0.57	0.42	0.78	1.01
	Visitas		39	18	15	12	24	26
Distância ($10^3 km$)	Total		2.0	2.78	4.10	5.01	2.79	2.63
	D1		1.31	1.77	2.44	2.94	1.55	1.76
	D2		0.69	1.01	1.66	2.07	1.24	0.87
		Tempo(s)	70.1	360.8	14.3	444.1	209.8	45.9
Adição de um veículo por depósito								
Número de veículos	Total		6	11	13	22	11	10
	D1		3	7	9	16	7	7
	D2		3	4	4	6	4	3
		Demanda total (Ton)	11.10	5.42	5.53	7.68	6.25	6.43
Quantidade	Carga (Ton)		3.84	0.66	0.57	0.42	0.78	1.01
	Visitas		39	18	15	12	24	26
Distância ($10^3 km$)	Total		2.47	2.85	4.58	5.13	3.05	3.04
	D1		1.56	1.74	2.64	3.08	1.73	1.91
	D2		0.91	1.1	1.94	2.05	1.32	1.13
		Tempo(s)	26.6	29.5	12.5	276.4	49.1	25.9

Tabela 4.8: Resultados computacionais obtidos via algoritmo Concorde para 150 e 200 vértices

Observa-se nas Tabelas 4.7 e 4.8 as diferentes custos de todas as instâncias em termos de distância. Para a instância 100c da primeira análise se teve uma distância de $4.96 \times 10^3 km$ no percurso dos clientes. Esta distância é diferente a instância 200c ($2.63 \times 10^3 km$) porque as localizações dos clientes para a instância 100c são mais longas entre eles e com os depósitos do que os clientes da instância 200c, além disso, cada instância conta com diferentes parâmetros de entrada. Porém, observa-se para algumas instâncias, maior distâncias percorridas com menores números de clientes. Por exemplo o percurso na instâncias 150c ($4.10 \times 10^3 km$) é maior do que a instâncias 200c ($2.63 \times 10^3 km$).

A Figura 4.13 apresenta os custos em termos de distâncias de todas as instâncias das Tabelas 4.7 e 4.8. Observa-se que os custos da instâncias com 50 vértices da primeira análise e os custos das mesmas instâncias adicionando um veículo por depósito na segunda análise são diferentes. Por exemplo, a distância

da instância 50c da primeira análise foi de $1.84 \times 10^3 km$ e da segunda análise para a mesma instância com mais dois veículos foi de $2.18 \times 10^3 km$, esse crescimento se percebe para todas as instâncias do problema. Assim, para a instância de 200c a distância na primeira análise foi de $2.63 \times 10^3 km$ e para a segunda análise foi de $3.04 \times 10^3 km$. Analisando as distâncias com a quantidade de veículos disponível, podemos concluir com esses resultados que a distância aumenta quanto mais veículos são usados.

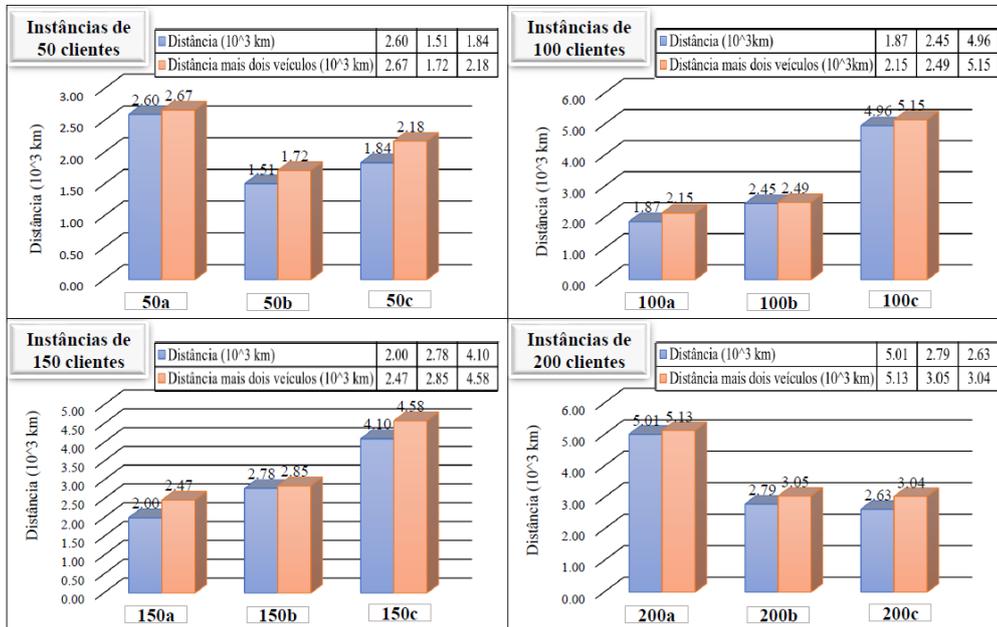


Figura 4.13: Análise dos custos obtidos via algoritmo Concorde para 50, 100, 150 e 200 vértices

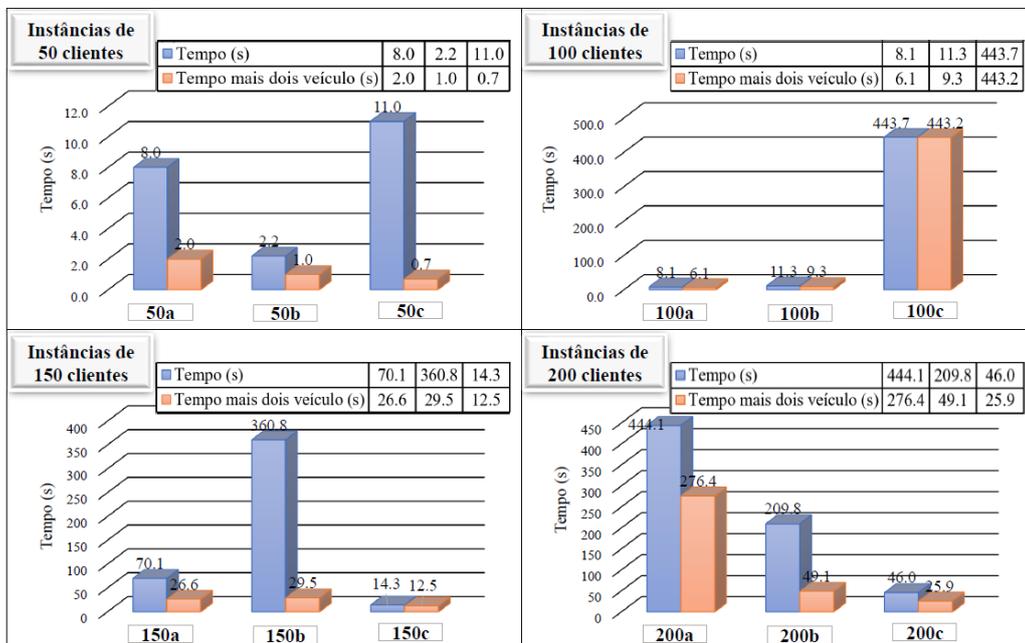


Figura 4.14: Análise dos tempos obtidos via algoritmo Concorde para 50, 100, 150 e 200 vértices

A Figura 4.14 foi desenhada para apresentar de forma visual os tempos gastos na procura das soluções do problema com múltiplos depósitos resolvido pela heurística de duas fases, usando o Concorde para resolver a segunda fase da heurística. Observa-se algumas irregularidade nos tempos para algumas instâncias, por exemplo para as instâncias 150b e 200b, normalmente a primeira precisaria de menos tempo investido do

que a instância com 200 clientes, mas nos testes revelaram que para a instância 150b o tempo em resolver o PRVMD foi de 360.8 segundos e para a instância 200b se investiu 208.8 segundos, isso é cada instância conta com parâmetros diferente, além disso as alocações mais próxima dos clientes os depósitos estão na instância 150b, porém na primeira fase da heurística vai ser mais demorado formar os *clusters* e designar clientes ao depósitos, este efeito aconteceu também nos tempos da heurística implementada.

Observa-se também os tempos para as instâncias com 50 vértices da primeira análise quanto para as mesmas instâncias acrescentando mais dois veículos da segunda análise, diferem entre eles. Por exemplo, a instância 50c da primeira análise usou 11 segundos para resolver o problema e o tempo usado na segunda análise para a mesma instância foi de 0.7 segundos, essa queda de tempo acontece para todas as instâncias testadas. Assim, para a instância 200c o tempo na primeira análise foi de 45.97 segundos e o tempo que demorou a segunda análise da mesma instância em resolver o problema foi de 25.9 segundos. Então podemos concluir neste estudo de caso, que o tempo que gasta a heurística em resolver o PRVMD diminui quanto mais veículos são disponibilizados, embora gere aumento no custo.

Resultados usando o algoritmo LKH na segunda fase da heurística

	Exemplar	50a	50b	50c	100a	100b	100c
Número de veículos	Total	13	5	3	3	4	14
	D1	10	4	2	2	3	9
	D2	3	1	1	1	1	5
	Demanda total	1.18	2.09	6.25	7.37	12.25	2.52
Quantidade	Carga	0.09	0.56	3.22	3.72	3.87	0.19
	Visitas	8	12	19	35	27	9
Distância (10 ³ km)	Total	2.60	1.51	1.83	1.87	2.43	4.95
	D1	1.63	0.96	1.13	1.28	1.88	3.62
	D2	0.97	0.55	0.70	0.59	0.55	1.33
	Tempo(s)	18.5	2.5	1.9	6.8	12.8	407.2
Adição de um veículo por depósito							
Número de veículos	Total	15	7	5	5	6	16
	D1	11	5	3	3	4	10
	D2	4	2	2	2	2	6
	Demanda total	1.18	2.09	6.25	7.37	12.25	2.52
Quantidade	Carga	0.09	0.56	3.22	3.72	3.87	0.19
	Visitas	5	12	19	35	27	9
Distância (10 ³ km)	Total	2.67	1.72	2.18	2.15	2.49	5.14
	D1	1.63	0.96	1.25	1.33	1.72	3.71
	D2	1.04	0.76	0.93	0.82	0.77	1.43
	Tempo(s)	3.3	1.6	1.0	8.9	8.5	443.5

Tabela 4.9: Resultados computacionais obtidos via algoritmo LKH para 50 e 100 vértices

Os resultados dos testes computacionais fornecidos pela heurística de duas fases, de agrupar primeiro e rotear depois desenvolvida para o PRVMD usando um tempo máximo de 30 minutos, as quais são visualizadas nas Tabelas 4.9 e 4.10. Para o desenvolvimento da segunda fase da heurística (roteamento) usou-se o algoritmo LKH. As tabelas revelam informações das demandas totais para as instâncias em toneladas, das quantidades de veículos a serem usados em cada depósito e instância, das capacidades dos veículos em toneladas, das visitas máximas para cada veículo, dos custos mínimos achados pelo algoritmo e dos tempos usados na busca de solução em segundos.

A Tabela 4.9, apresenta duas análises das instâncias de 50 e 100 clientes, a primeira análise diferencia-se da segunda pelas quantidades de mais um veículo usado em cada depósito, quer dizer, para a instância 50b da primeira análise conta-se com 5 veículos a serem usados e dois depósitos, na segunda análise da mesma instância se disponibiliza um veículo adicional nos depósitos, quer dizer 7 veículos no total.

Observa-se nas Tabelas 4.9 e 4.10 os diferentes custos de todas as instâncias em termos de distância. Para a instância 100c da primeira análise se teve uma distância de $4.95 \times 10^3 km$. Esta distância é diferente a instância 150c ($4.10 \times 10^3 km$) porque as localizações dos clientes para a instância 100c são mais longas entre eles e com os depósitos do que os clientes da instância 150c, além disso, cada instância conta com diferentes parâmetros de entrada. Para algumas instâncias conta-se com maior distâncias percorridas com menores números de clientes. Por exemplo o percurso na instâncias 50a ($2.60 \times 10^3 km$) é maior do que a instâncias 100a ($1.87 \times 10^3 km$).

A Tabela 4.10 apresenta os resultados da heurística para o PRVMD com a mesma análise realizada na Tabela 4.9 para as instâncias de 150 e 200 vértices. A instância 150b da primeira análise conta com 9 veículos a serem usados e na segunda análise da mesma instância com 11 veículos no total. Esse crescimento das quantidades de veículos foram feitos para gerar maior número de rotas em todas as instâncias, com a finalidade de diminuir os custos e os tempos na busca das soluções.

	Exemplar	150a	150b	150c	200a	200b	200c
Número de veículos	Total	4	9	11	20	9	8
	D1	2	6	8	15	6	6
	D2	2	3	3	5	3	2
Quantidade	Demanda total (Ton)	11.10	5.42	5.53	7.68	6.25	6.43
	Carga (Ton)	3.84	0.66	0.57	0.42	0.78	1.01
	Visitas	39	18	15	12	24	26
Distância ($10^3 km$)	Total	2.00	2.77	4.10	5.05	2.78	2.65
	D1	1.30	1.76	2.44	2.98	1.54	1.77
	D2	0.70	1.01	1.66	2.07	1.24	0.88
	Tempo(s)	71.1	342.6	15.7	445.4	196.7	47.0
Adição de um veículo por depósito							
Número de veículos	Total	6	11	13	22	11	10
	D1	3	7	9	16	7	7
	D2	3	4	4	6	4	3

Continua na próxima página

		Exemplar	150a	150b	150c	200a	200b	200c
Quantidade	Demanda total (Ton)		11.10	5.42	5.53	7.68	6.25	6.43
	Carga (Ton)		3.84	0.66	0.57	0.42	0.78	1.01
	Visitas		39	18	15	12	24	26
	Total		2.46	2.85	4.57	5.12	3.05	3.04
Distância (10^3 km)	D1		1.56	1.75	2.63	3.08	1.73	1.91
	D2		0.90	1.10	1.94	2.04	1.32	1.13
	Tempo(s)		26.6	25.7	13.7	272.0	44.7	26.9

Tabela 4.10: Resultados computacionais obtidos via algoritmo LKH para 150 e 200 vértices

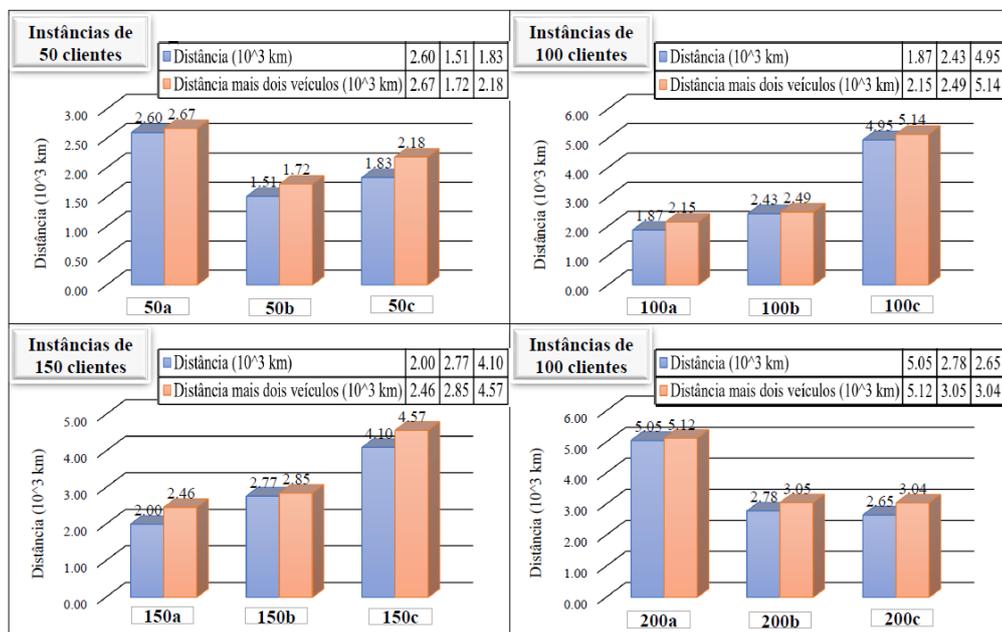


Figura 4.15: Análise dos custos obtidos via algoritmo LKH para 50, 100, 150 e 200 vértices

A Figura 4.15 foi desenhada para visualizar os resultados de resolver o PRVMD de todas as instâncias via algoritmo LKH. Observa-se que as distâncias das instâncias com 100 clientes da primeira análise e a distância da mesma instância adicionando um veículo por depósito, na segunda análise são diferentes. Por exemplo, a distância da instância 100c da primeira análise foi de 4.95×10^3 km e da segunda análise para a mesma instância com mais dois veículos foi de 5.14×10^3 km, esse crescimento acontece para todas as instâncias do problema. Assim, para a instância 200b da primeira análise se teve uma distância de 2.78×10^3 km e para a segunda análise foi de 3.05×10^3 km. Analisando estes resultados com as quantidades de veículos disponíveis no depósito podemos concluir, para este estudo de caso que as distâncias aumentam quanto mais veículos forem usados.

A Figura 4.16, apresenta os tempos usados pela heurística para a procura das soluções em todas as instâncias testadas usando o algoritmo de duas fases implementado. Nesta parte, a fase de roteamento foi resolvida pelo algoritmo LKH. Observa-se algumas disparidades nos tempos para algumas instâncias, por exemplo para as instâncias 100c e 150c, normalmente a primeira precisaria de menos tempo investido do que a instância com 150c clientes, mas nos testes revelaram que para a instância 100c o tempo em resolver

o PRVMD foi de 407.0 segundos e para a instância 200b se investiu 15.7 segundos, isso é por que os dados de entrada são diferentes, porém na primeira fase da heurística vai ser mais demorado formar os *clusters* e designar clientes ao depósitos na instância 100c.

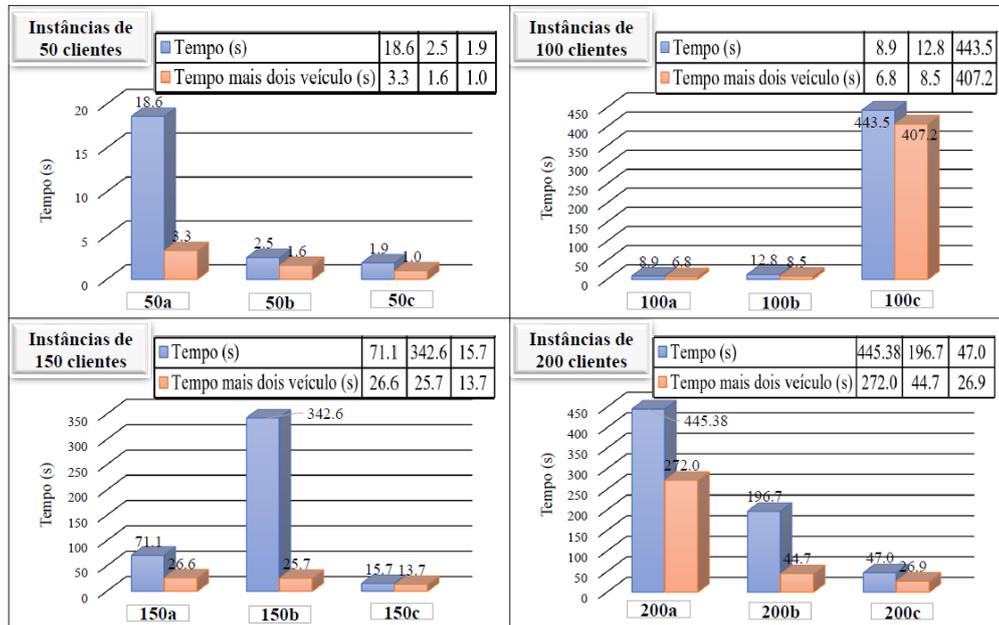


Figura 4.16: Análise dos tempos obtidos via algoritmo LKH para 50, 100, 150 e 200 vértices

Observa-se que os tempos para as instâncias com 50 clientes da primeira análise e a mesma instância acrescentando mais dois veículos na segunda análise não são iguais. Por exemplo, a instância 100c da primeira análise usou 443.5 segundos em resolver o problema e para a segunda análise da mesma instâncias foi de 407.2 segundos, essa queda de tempo acontece para todas as instâncias testadas. Assim, para a instância 200b o tempo em achar a solução na primeira análise foi de 196.7 segundos e o tempo na segunda análise da mesma instância foi de 44.7 segundos. Então podemos concluir para este estudo de caso, que o tempo que gasta o algoritmos LKH em resolver o PRVMD diminui quanto mais veículos são disponibilizados, este comportamento nos tempos aconteceu também usando o Concorde como algoritmo para resolver o PCV na heurística implementada.

4.4.3 Análise dos Resultados dos Testes Computacionais para as Metodologias

Nesta seção serão analisados os resultados dos testes computacionais das instâncias 50, 100, 150 e 200 e as suas repetições em “a”, “b” e “c”. Nas seguintes tabelas e figuras serão apresentados o resumo dos resultados para o PRVMD via heurística, algoritmo Concorde e algoritmo LKH, para um tempo máximo de 30 minutos usando um computador descrito em 4.3.2. Mediante esta análise será definido o algoritmo de maior eficiência na resolução de instâncias com maior número de clientes.

Resultados dos algoritmos para as instância 50a, 100a, 150a e 200a

A Tabela 4.11 e a Figura 4.17 apresentam os resultados dos custos em termos de distância dos algoritmos implementados, da mesma forma os tempos computacionais investido em resolver o PRVMD via heurística para as instâncias 50a, 100a, 150a e 200a. Observa-se que os resultados, usando qualquer algoritmo testado,

não dependem exclusivamente das quantidades de clientes a serem atendidos. Estas variabilidades depende das demandas das instâncias, das capacidades e das quantidades dos veículos disponíveis nos depósitos e pelas visitas máximas por rotas. Por exemplo, para o algoritmo heurístico na instância 50a gerou-se uma distância de $2.58 \times 10^3 km$ utilizando 13 veículos de 0.09 toneladas de capacidade cada um deles, uma demanda total de 1.18 toneladas e 5 visitas como máximo por rota. Para a instância 100a gerou-se uma distância de $1.85 \times 10^3 km$, esta distância é inferior a instância 50a porque as condições dos veículos são diferentes. A instância 100a conta com 3 veículos de 3.72 toneladas de capacidade, uma demanda total de 7.37 e 35 visitas como máximo por rota.

Exemplar	Método Heurístico		Algoritmo Concorde		Algoritmo LKH	
	Distância (10 ³ km)	Tempo (s)	Distância (10 ³ km)	Tempo (s)	Distância (10 ³ km)	Tempo (s)
50a	2.58	31.10	2.60	8.00	2.60	18.60
100a	1.85	917.10	1.87	8.10	1.87	8.90
150a	1.95	942.40	2.00	70.10	2.00	71.10
200a	4.97	103.70	5.01	444.10	5.05	445.38

Adição de um veículo por depósito						
Exemplar	Método Heurístico		Algoritmo Concorde		Algoritmo LKH	
	Distância (10 ³ km)	Tempo (s)	Distância (10 ³ km)	Tempo (s)	Distância (10 ³ km)	Tempo (s)
50a	2.65	1.00	2.67	2.00	2.67	3.30
100a	2.11	602.30	2.15	6.10	2.15	6.80
150a	2.41	492.40	2.47	26.60	2.46	26.60
200a	5.04	103.10	5.13	276.40	5.12	272.00

Tabela 4.11: Resultados dos testes computacionais das instâncias 50a, 100a, 150a e 200a

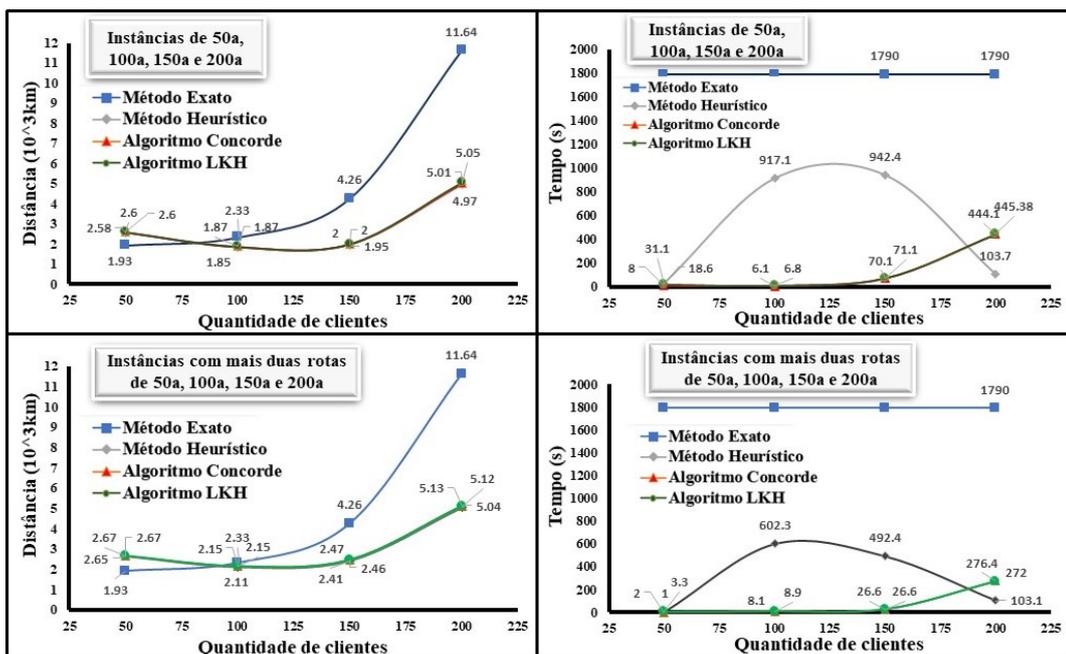


Figura 4.17: Análise dos resultados dos testes computacionais das instâncias 50a, 100a, 150a e 200a

Para a instância de 50 clientes, o melhor resultado fornecido para este estudo de caso foi produzido pelo método exato ($1.93 \times 10^3 km$), embora tenha usado um maior tempo computacional do que os outros algoritmos (1800 segundos). Observe-se na Figura 4.17, que conforme cresce a quantidade de clientes o método exato tem dificuldade em resolver o PRVMD por causa da complexidade do problema, do recurso do computador e do tempo limitado. Contrariamente acontece com o uso da heurística, quando as instâncias são pequenas o uso deste método oferece soluções de menor qualidade que as ofertadas pelo método exato, mas quando o tamanho dos clientes aumenta, a eficiência das heurísticas em resolver o PRVMD também cresce.

A Tabela 4.11 e a Figura 4.17 também apresentam os resultados das distâncias e os tempos investidos em resolver o PRVMD das instâncias 50a, 100a, 150a e 200a com mais um veículo por depósito. As análises destes resultados são similares com aqueles analisadas anteriormente. Observa-se na Figura 4.17 as distâncias fornecidos pelos algoritmos LKH, Concorde e o método heurístico são próximas entre eles. Note-se nesta instância que o tempo investido pelos algoritmos Concorde e LKH, são superiores ao método heurístico, este fenômeno pode ser causado pela complexidade do problema e a formação de sub rotas. Esta anomalia ainda persiste, em menor grau, na instância 200a da segunda análise com mais dois veículos.

Os resultados fornecidos pelos algoritmos de maior eficiência, em termos de distância e tempo para este estudo, são produzidos pelo algoritmo Concorde e o algoritmo LKH. Um dos trabalhos que pode dar sustentação à análise para esta abordagem é o estudo de [Baniyadi et al. \(2018\)](#), onde os autores trabalharam o PCV com até 1000 pontos usando os algoritmos Concorde, LKH, Chained Lin-Kernighan (CLK) e Snakes and Ladders Heuristic (SLH), os resultados de maior eficiência foram fornecidos pelos algoritmos LKH e o algoritmo Concorde. No trabalho de [Rodríguez & Ruiz \(2009\)](#) foram analisados os algoritmos Concorde e LKH para o PCV, nesse estudo foram analisados um máximo de 550 pontos. Os resultados concluíram que a heurística LKH investe menos tempo em alcançar soluções de maior eficiência do que o algoritmo Concorde para instâncias superiores a 350 pontos, para as instâncias de 50, 100, 200 e 250 os resultados achados não diferem entre eles.

Resultados dos algoritmos para as instâncias 50b, 100b, 150b e 200b

A Tabela 4.12 e a Figura 4.18 apresentam os resultados em termos de distância dos algoritmos implementados, da mesma forma os tempos investidos nos testes das instâncias 50b, 100b, 150b e 200b para resolver o problema multi-depósito. Observa-se que os resultados fornecidos pelos algoritmos nas instâncias testadas não dependem das quantidades de clientes a serem atendidos. Estas variabilidades dependem das demandas dos clientes, das capacidades e as quantidades dos veículos disponíveis nos depósitos e pelas visitas máximas por rotas. Por exemplo, para o algoritmo Concorde na instância 150b gerou-se uma distância de $2.78 \times 10^3 km$ utilizando 9 veículos de 0.66 toneladas de capacidade, uma demanda total de 5.42 toneladas e 18 visitas como máximo por rota. Para a instância 200b gerou-se $2.79 \times 10^3 km$, esta distância é ligeiramente superior do que a instância 150b causada pelos diferentes parâmetros. A instância 200b conta com 9 veículos de 0.78 toneladas de capacidade, uma demanda total de 6.25 e 24 visitas como máximo por rota.

O melhor resultado fornecido para a instância de 50 clientes ($1.28 \times 10^3 km$) foi produzido pelo método exato, embora tenham usado um maior tempo computacional do que os outros algoritmos (1800 segundos).

Podemos observar na Figura 4.18 um crescimento das distâncias por parte da metodologia exata a qual é produto pelo aumento na quantidade de clientes, da complexidade do problema, do recurso do computador e do tempo limitado. Contrariamente acontece com a heurística, quando o tamanho da instância cresce a eficiência da heurística para resolver o problema torna-se uma melhor opção.

Exemplar	Método Heurístico		Algoritmo Concorde		Algoritmo LKH	
	Distância (10 ³ km)	Tempo (s)	Distância (10 ³ km)	Tempo (s)	Distância (10 ³ km)	Tempo (s)
50b	1.49	14.20	1.51	2.20	1.51	2.50
100b	2.45	488.70	2.45	11.30	2.43	12.80
150b	2.72	234.90	2.78	360.80	2.77	342.60
200b	2.72	104.40	2.79	209.80	2.78	196.70

Adição de um veículo por depósito						
Exemplar	Método Heurístico		Algoritmo Concorde		Algoritmo LKH	
	Distância (10 ³ km)	Tempo (s)	Distância (10 ³ km)	Tempo (s)	Distância (10 ³ km)	Tempo (s)
50b	1.69	1.00	1.72	1.00	1.72	1.60
100b	2.45	296.80	2.49	9.30	2.49	8.50
150b	2.80	39.30	2.85	29.50	2.85	25.70
200b	2.97	31.10	3.05	49.10	3.05	44.70

Tabela 4.12: Resultados dos testes computacionais das instâncias 50b, 100b, 150b e 200b

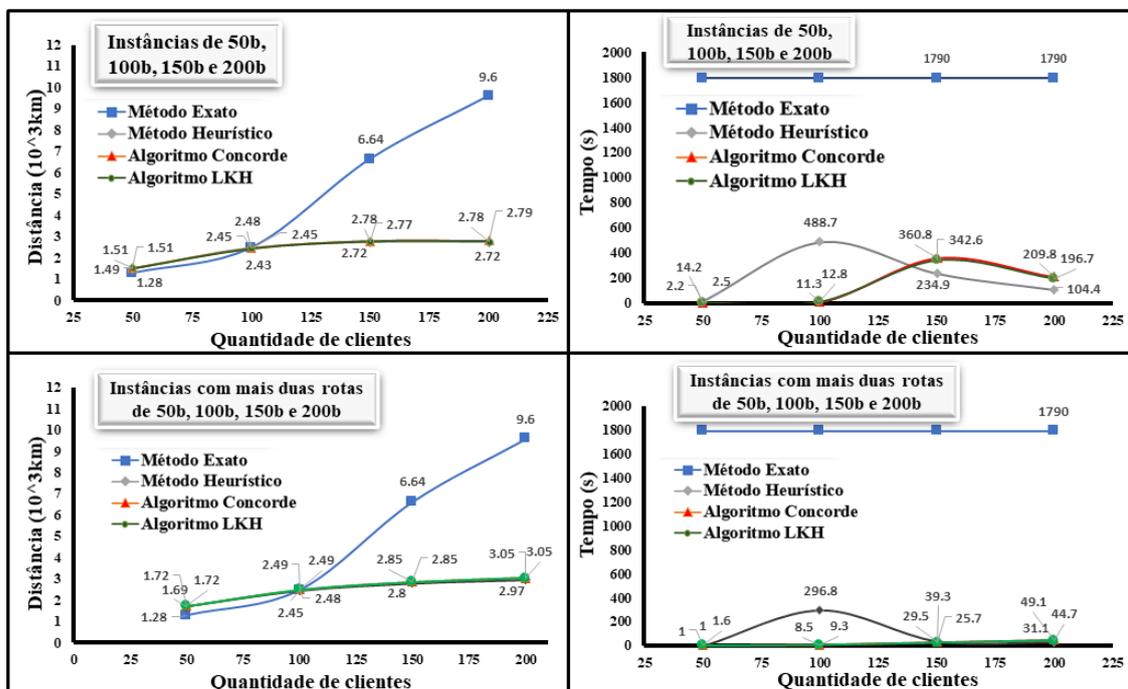


Figura 4.18: Análise dos resultados dos testes computacionais das instâncias 50b, 100b, 150b e 200b

A Tabela 4.12 e a Figura 4.18 também apresenta os resultados e os tempos em resolver o PRVMD para uma segunda análise das instâncias 50b, 100b, 150b e 200b com mais um veículo por depósito. Os resultados desta segunda análise não são tão diferentes quando as ofertadas pela primeira análise.

Os melhores resultados em termos de distância total fornecidos pela heurística e pelos algoritmos Concorde e LKH nas instâncias 150b e 200b apresentada na Figura 4.18 são próximas entre eles, $2,78$ e $2,79 \times 10^3 km$ com o algoritmo Concorde e, $2,77$ e $2,78 \times 10^3 km$ com o algoritmo LKH respectivamente. Nestas mesmas instâncias, os tempos investidos pelos algoritmos Concorde (360.80 e 209.80 segundos) e LKH (342.60 e 196 segundos), são superiores ao método heurístico (234 e 104 segundos), este fenômeno pode ser causado pela complexidade do problema e a formação de sub rotas. Esta anomalia é dissipada acrescentando mais dois veículo por depósito para as instâncias 150b e 200b na segunda análise.

Os resultados fornecidos pelos algoritmos de maior eficiência, em termos de distância mínima e tempo, são produzidos pelo algoritmo Concorde e o algoritmo LKH como aconteceu nas instâncias 50a, 100a, 150a e 200a. Podemos citar mais alguns trabalhos que validam a efetividade dos algoritmos LKH e Concorde, por exemplo, Fischer et al. (2005) fizeram provas para o PCV usando o algoritmo Concorde e o algoritmo LKH para instâncias de até 1000 pontos, concluindo que o Concorde resolveu o problema com menor tempo do que o LKH. No trabalho de Dubois-Lacoste et al. (2015) analisaram a efetividade do tempo em resolver o PCV do algoritmo LKH comparada com a técnica construtiva, os resultados mostraram a efetividade do algoritmo LKH em fornecer melhores soluções para um máximo de 1500 pontos comparados com a heurística construtiva.

Resultados dos algoritmos para as instância 50c, 100c, 150c e 200c

Exemplar	Método Heurístico		Algoritmo Concorde		Algoritmo LKH	
	Distância (10^3 km)	Tempo (s)	Distância (10^3 km)	Tempo (s)	Distância (10^3 km)	Tempo (s)
50c	1.81	2.30	1.84	11.00	1.83	1.90
100c	4.91	95.20	4.96	443.70	4.95	443.50
150c	4.03	381.50	4.10	14.30	4.10	15.70
200c	2.62	970.20	2.63	45.97	2.65	47.00

Adição de um veículo por depósito						
Exemplar	Método Heurístico		Algoritmo Concorde		Algoritmo LKH	
	Distância (10^3 km)	Tempo (s)	Distância (10^3 km)	Tempo (s)	Distância (10^3 km)	Tempo (s)
50c	2.16	1.00	2.18	0.70	2.18	1.00
100c	5.09	95.00	5.15	443.20	5.14	407.20
150c	4.50	371.10	4.58	12.50	4.57	13.70
200c	2.98	765.40	3.04	25.90	3.04	26.90

Tabela 4.13: Resultados dos testes computacionais das instâncias 50c, 100c, 150c e 200c

A Tabela 4.13 e a Figura 4.19 apresentam os resultados das distâncias mínimas e os tempos investidos em resolver o PRVMD para as instâncias 50c, 100c, 150c e 200c usando as metodologias descritas nas Seções anteriores. O uso do algoritmo LKH na instância 100c forneceu uma distância de $4.95 \times 10^3 km$ utilizando 14 veículos de 0.19 toneladas de capacidade, uma demanda total de 2.52 toneladas e 9 visitas como máximo por rota. Para a instância 200c se obteve uma distância de $2.65 \times 10^3 km$, esta distância é inferior a instância

100c pelos diferentes valores nos parâmetros. A instância 200c conta com 8 veículos de 1.01 toneladas de capacidade, uma demanda total de 6.43 toneladas e 26 visitas como máximo por rota.

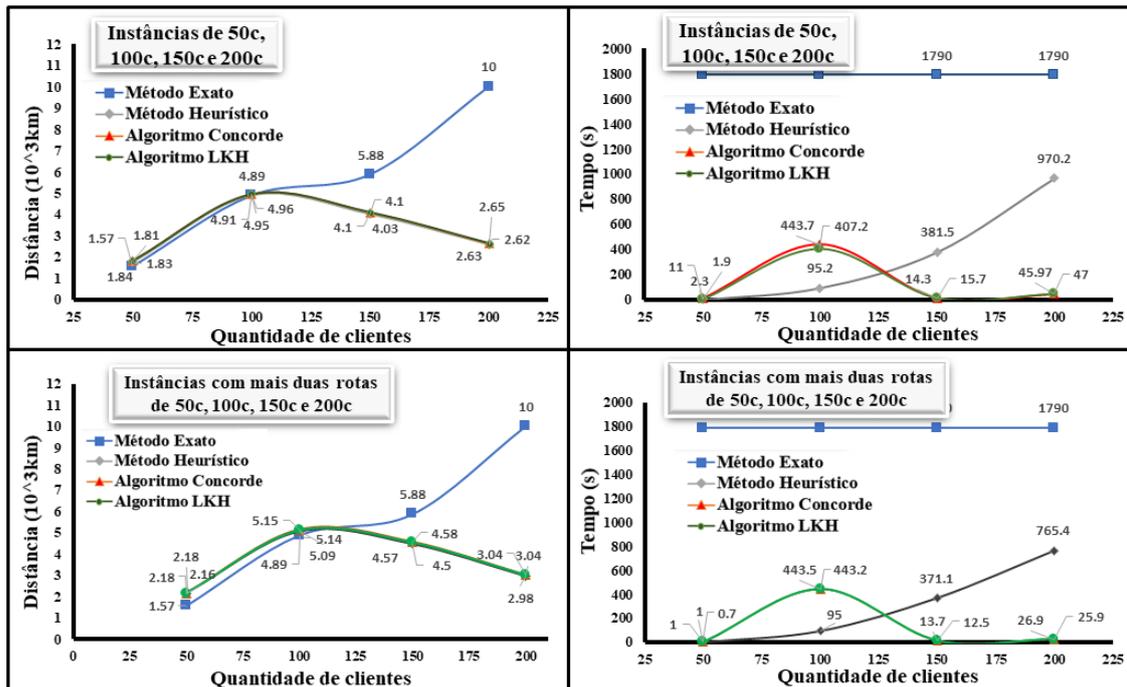


Figura 4.19: Análise dos resultados dos testes computacionais das instâncias 50c, 100c, 150c e 200c

Na Figura 4.19 observa-se os melhores resultados para resolver o PRVMD, estes foram fornecidos pelo algoritmo Concorde e pelo algoritmo LKH nas instâncias 150c e 200c. Com base nos resultados, os menores tempos usados nas procuras das soluções dos testes computacionais foram os algoritmos Concorde e LKH para estas instâncias. Na Tabela 4.13 e na Figura 4.19 apresentam também uma segunda análise das instâncias 50c, 100c, 150c e 200c com mais um veículo por depósito, os resultados obtidos são similares com os resultados da primeira análise.

4.4.4 Uso da Heurística LKH em Instâncias com Maior Número de Clientes

Conforme as análises dos resultados para os testes computacionais apresentados anteriormente na resolução do PRVMD, foram testadas as instâncias de 500, 1000, 1500, 2000, 2500 e 6804 clientes, esta última representa aos clientes atendidos no dia 16 de junho como se apresenta na instância 1 da Tabela 4.2. As instâncias de 500 até 2500 foram criadas acrescentando 500 clientes de forma gradual com a finalidade de observar melhor o comportamento dos resultados.

Um dos objetivos do estudo de caso foi resolver o PRVMD mediante a heurística que agrupa primeiro e roteia depois, a qual foi selecionada depois da revisão da literatura para diferentes formas de resolver o PRVMD. A primeira fase foi resolvida usando uma versão do algoritmo *k-means*, adaptando o modelo matemático de Seref et al. (2014) para designar quantidades de clientes aos depósitos formando dois *clusters* e, posteriormente usando este mesmo modelo para a formação dos sub-*clusters*. Com base nos testes preliminares das seções anteriores a segunda fase da heurística foi resolvida usando o algoritmo LKH, este algoritmo foi escolhido por ser mais eficiente na obtenção dos resultados em termos de distância mínima com o menor tempo computacional do que aqueles fornecidos pelo método exato e pelo algoritmo Concorde.

	Exemplar	500	1000	1500	2000	2500	6804
Número de veículos	Total	16	14	18	22	27	-
	D1	10	8	10	12	16	-
	D2	6	6	8	10	11	-
Demanda total		12.26	46.46	62.72	67.25	83.51	-
Capacidade máxima	Carga	1.07	3.72	3.72	3.72	3.22	-
	Visitas	65	120	197	246	149	-
Custo	Total	5.05	10.0	14.83	19.35	24.31	-
	D1	3.32	6.64	8.82	12.74	16.14	-
	D2	1.71	3.36	6.01	6.61	8.17	-
Tempo(s)		39.6	141.1	1450.0	2624.0	9185.0	-

Tabela 4.14: Resultados computacionais das instâncias de 500, 100, 1500, 200, 2500 e 6804 clientes

O tempo máximo estabelecido para a execução da heurística foi 3 horas, nesse tempo pretende-se solucionar o PRVMD para a instância com 6804 clientes a qual corresponde á media de clientes diariamente atendido pela empresa EPL1. A Tabela 4.14 fornece informações das demandas totais em toneladas das instâncias, as quantidades de veículos a serem usados em cada depósito e para cada instância, das capacidades em toneladas dos veículos, das visitas máximas para cada veículo por rota, dos custos mínimos achados pela algoritmo LKH e dos tempos em segundos em resolver o problema.

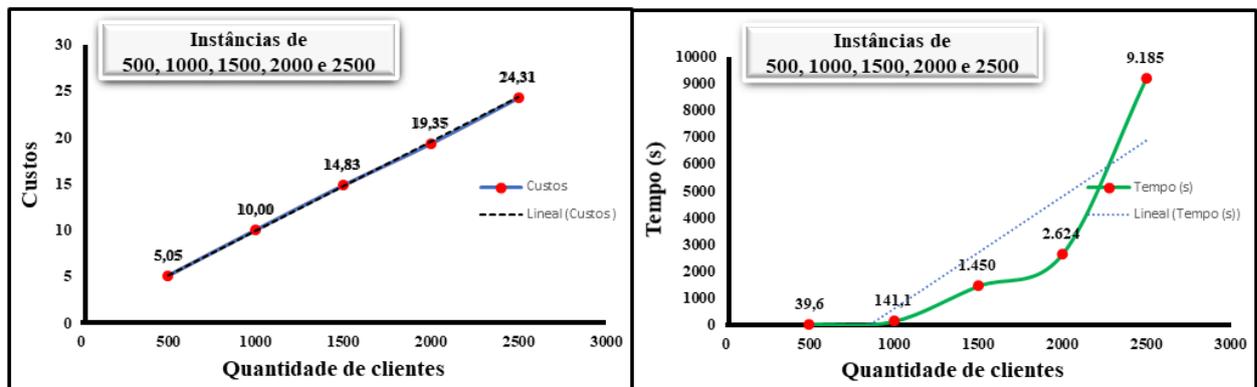


Figura 4.20: Análise dos resultados da heurística em instâncias de 500, 1000, 1500, 200 e 2500 clientes

Segundo a Figura 4.20, se observa que os custos e os tempos investidos na procura das soluções são maiores conforme aumenta a quantidade dos clientes no atendimento. Por exemplo, na instâncias com 1000 clientes a distância mínima foi de $10.0 \times 10^3 km$, esta solução é conseguida em um tempo 141 segundos. Da mesma forma o tempo investido para a instância com 2000 clientes foi de 2624.0 segundos e uma distância mínima de $19.35 \times 10^3 km$.

Os resultados da heurística para a instância de 6804 clientes apresentado na Tabela 4.14 não foram conseguidos dentro das 3 horas estabelecidas, isso pode ser causada pela complexidade do problema na primeira fase da heurística onde usou-se o método exato via solver Gurobi. Considera-se oportuno a implementação de algoritmos aproximados na primeira fase da heurística onde se designa os clientes ao depósitos, quer dizer a implementação de heurísticas ou meta-heurísticas para a formação dos grupos.

Conclusões e Próximos Passos

5.1 Conclusões Gerais

Este trabalho apresenta um estudo sobre o Problema de Roteamento de Veículos (PRV) e as suas diferentes extensões, começando com as restrições de capacidade (PRVC), passando pela variante que apresenta restrições de janelas de tempo (PRVJT), outras com restrições de demandas específicas (PRVE), com múltiplos depósitos (PRVMD), entre outras. Além disso, foi abordado os métodos e algoritmos que resolvem as variantes do PRV, começando pelos métodos exatos os quais provêm soluções ótimas, até chegar aos algoritmos heurísticos e meta-heurísticos. Podemos concluir, a partir da literatura, que os PRVs são razoavelmente fáceis de descrever, porém muito difíceis de resolver. Muitos pesquisadores usaram conceitos do PRV em aplicações reais como uma forma interessante de planejar o transporte nas indústrias e em outras áreas com a finalidade de ajudar no gerenciamento de rotas com baixo custo.

O PRV é de grande importância e tem aplicações relevantes no sistema de transporte de mercadorias. Diversas pesquisas desenvolveram conceitos práticos e teóricos, além de apresentarem uma série de métodos de resolução para problemas de distribuição. Embora diferentes autores tenham apresentado métodos de soluções exatas para encontrar uma solução ótima do problema, tais problemas requerem um alto custo computacional. Outros pesquisadores resolveram usar algoritmos heurísticos e meta-heurísticos para resolverem problemas de grande porte. Estes métodos de otimização combinatória, frequentemente, são capazes de encontrar soluções de boa qualidade em baixo tempo computacional do que os métodos exatos. A combinação destes métodos (exatos e aproximados) geram métodos híbridos, que permitem superar a dificuldade com a complexidade computacional do problema e aprimorar as soluções. Os métodos aproximados são as abordagens mais estudadas nos últimos anos e, da mesma forma, as inclusões de novas restrições apresentam novas variantes e novos desafios no estudo dos PRVs. Em particular, a partir da revisão da literatura foi possível escolher um caminho para resolver o PRVMD por meio do uso e implementação de uma metodologia híbrida de duas fases. A qualidade das soluções determinadas em detrimento da dificuldade de atuar métodos exatos foi o motivo da escolha, visto da perspectiva da complexidade e tempo computacional na resolução de exemplares de elevada complexidade.

5.2 Conclusões para o Estudo de Caso

Como forma de proposta para se refletir sobre o aprendizado e validar alguns dos métodos estudados, desenvolveu-se um estudo de caso, com base em dados reais de uma empresa Peruana chamada como EPL1. Alguns dos objetivos deste estudo de caso foi refletir o aprendizado sobre o PRV, assim também das formas de como resolvê-lo, gerando soluções por meio dos algoritmos exatos e aproximados, onde se considerou um caminho para auxiliar no planejamento da distribuição das mercadorias e na formação das rotas. Utilizando os dados fornecidos pela empresa EPL1 foi possível diferenciar e identificar a variante do PRV como mais adequada para os interesses dessa aplicação. Após a definição do PRVMD, optou-se por resolvê-lo de forma exata usando conceitos de Programação Linear Inteira e do algoritmo *Branch and Bound* mediante o solver Gurobi, usando a formulação matemática trazida da pesquisa de [Kulkarni & Bhave \(1985\)](#) para o PRVMD. Os resultados encontrados mostraram que o uso de um algoritmo exato para resolver o PRVMD forneceu soluções não ótimas para o tempo computacional limite de 30 minutos, verificados pelos tamanhos dos GAP's achados na resolução do problema. Por conta disso, foi utilizado um algoritmo heurístico na procura de melhores resultados em um tempo igual ou inferior ao fornecido no método exato.

Neste estudo, a heurística de agrupar primeiro e rotear depois apresentou uma série de características que favorecem a sua escolha para resolver o PRVMD. Com a finalidade de melhorar a eficiência na procura das soluções em menor tempo, se optou pelo auxílio dos algoritmos Concorde e LKH na segunda fase da heurística. Foram resolvidas instâncias com 50, 100, 150 e 200 clientes tanto para o método exato quanto para a heurística de duas fases. A primeira fase da heurística, foi feita usando método exato de *clustering* partindo do modelo de [Seref et al. \(2014\)](#), designando os clientes para cada centro de distribuição os quais foram considerados como centroides. A segunda fase foram criadas as rotas onde os percursos destas foram testadas com os algoritmos para o PCV, além disso usou-se os algoritmos Concorde e LKH. Os resultados com maior eficiência para estas instâncias, em termos de custo e tempo, foram os fornecidos pelo algoritmo LKH. Também foram testados instâncias de 500, 1000, 1500, 2000 2500 e 6804 clientes para um tempo máximo de 3 horas, onde a segunda fase da heurística foi resolvida pelo algoritmo LKH e se conseguiram resultados até 2500 clientes.

Não foi possível fornecer resultados para a instância de 6804 clientes em um tempo máximo de 3 horas, esta instância representa a quantidade de clientes atendido pela empresa em análise no dia 16 de junho de 2021. Se concluiu que não foi possível resolver o PRVMD com a heurística implementada para a media de 6000 clientes que atende a empresa EPL1 dentro do período de tempo estabelecido. Os resultados achados nos testes computacionais garante que é possível gerar soluções para instâncias pequenas de até 2500 clientes em um tempo de 3 horas para algumas empresas com menor número de clientes.

Os resultados encontrados com os algoritmos exatos, comparado com os resultados fornecidos pela heurística, mostraram que é melhor optar pelo uso e implementação deste último, desde que o tempo computacional para a heurística seja inferior aos algoritmos exatos e também considerando a redução dos custos. Podemos concluir com o estudo do PRV, as suas extensões e os métodos de soluções abordados nesta dissertação foram parcialmente aplicados e comprovados. Além disso, foi possível testar e criar rotas de veículo no processo de distribuição dos produtos para as instâncias surgidas com os dados da empresa EPL1. Desta forma, comprovamos a possibilidade de maximizar os lucros, reduzir os custos das viagens e aumentar o nível de serviço aos clientes.

5.3 Próximos Passos

É conveniente procurar outras formas de heurísticas para resolver o PRVMD, como o algoritmo de varredura. Da mesma forma, a implementação de meta-heurísticas como aquela baseada no comportamento das formigas pode melhorar os resultados computacionais. Também é interessante considerar a implementação de algoritmos híbridos que recentemente vêm se mostrando eficientes em resolver os PRVs.

Torna-se aplicável a metodologia proposta em empresas que atendam um menor número de clientes do que a EPL1, de modo a reduzir uma porcentagem de perdas geradas por uma péssima gestão na distribuição das mercadorias e na formação das rotas. Por consequência, essas soluções poderiam melhorar consideravelmente a formação de rotas de distribuição, à formação de grupos de clientes que serão atendidos por cada depósito, assim, satisfariam as demandas e requerimentos dos clientes.

É possível aprofundar o estudo do PRV, especificamente da variante com demanda estocástica, por conta de ser uma extensão do PRV que reflete melhor a realidade dos problemas no transporte de mercadorias. Nesse contexto, a abordagem de meta-heurísticas evolutivas seria uma excelente escolha para resolvê-lo. Esta ideia pretende ser desenvolvida nos próximos anos. Partindo desta pesquisa e de outros trabalhos, é possível fornecer soluções aos problemas empresariais no gerenciamento e desenho de rotas na rede de distribuição logística, usando algoritmos de otimização exata ou combinatórias para otimizar o tempo de entrega dos produtos, reduzir os custos de transporte, satisfazer a demandas dos clientes com melhor qualidade e evitar possíveis reclamações pela demora na entrega das mercadorias.

Bibliografía

- Aarts, E., Aarts, E. H. & Lenstra, J. K. (2003), *Local search in combinatorial optimization*, Princeton University Press.
- Anbuudayasankar, S., Ganesh, K., Koh, S. L. & Ducq, Y. (2012), 'Modified savings heuristics and genetic algorithm for bi-objective vehicle routing problem with forced backhauls', *Expert Systems with Applications* **39**(3), 2296–2305.
- Applegate, D., Bixby, R., Chvátal, V. & Cook, W. (2003), 'Implementing the dantzig-fulkerson-johnson algorithm for large traveling salesman problems', *Mathematical programming* **97**(1), 91–153.
- Applegate, D. L., Bixby, R. E., Chvátal, V. & Cook, W. J. (2011), *The traveling salesman problem*, Princeton university press.
- Archetti, C., Feillet, D., Gendreau, M. & Speranza, M. G. (2011), 'Complexity of the VRP and SDVRP', *Transportation Research Part C: Emerging Technologies* **19**(5), 741–750.
- Archetti, C. & Speranza, M. G. (2008), The split delivery vehicle routing problem: A survey, in 'The vehicle routing problem: Latest advances and new challenges', Springer, pp. 103–122.
- Augerat, P., Naddef, D., Belenguer, J., Benavent, E., Corberan, A. & Rinaldi, G. (1995), 'Computational results with a branch and cut code for the capacitated vehicle routing problem'.
- Balas, E., Ceria, S., Cornuéjols, G. & Natraj, N. (1996), 'Gomory cuts revisited', *Operations Research Letters* **19**(1), 1–9.
- Baldacci, R., Toth, P. & Vigo, D. (2007), 'Recent advances in vehicle routing exact algorithms', *4OR* **5**(4), 269–298.
- Balinski, M. L. & Quandt, R. E. (1964), 'On an integer program for a delivery problem', *Operations research* **12**(2), 300–304.
- Ballou, R. H. & de Lemus, P. R. (1991), *Logística empresarial: control y planificación*, Díaz de Santos.
- Baniasadi, P., Ejov, V., Haythorpe, M. & Rossomakhine, S. (2018), 'A new benchmark set for traveling salesman problem and hamiltonian cycle problem', *Cornell University* .

- Barnhart, C., Johnson, E. L., Nemhauser, G. L., Savelsbergh, M. W. & Vance, P. H. (1998), 'Branch-and-price: Column generation for solving huge integer programs', *Operations research* **46**(3), 316–329.
- Beasley, J. E. (1983), 'Route first—cluster second methods for vehicle routing', *Omega* **11**(4), 403–408.
- Belenguer, J.-M., Martinez, M. & Mota, E. (2000), 'A lower bound for the split delivery vehicle routing problem', *Operations research* **48**(5), 801–810.
- Beltrami, E. J. & Bodin, L. D. (1974), 'Networks and vehicle routing for municipal waste collection', *Networks* **4**(1), 65–94.
- Berkhin, P. (2006), A survey of clustering data mining techniques, in 'Grouping multidimensional data', Springer, pp. 25–71.
- Bertsimas, D. J. (1992), 'A vehicle routing problem with stochastic demand', *Operations Research* **40**(3), 574–585.
- Bochtis, D. & Sorensen, C. G. (2009), 'The vehicle routing problem in field logistics part I', *Biosystems engineering* **104**(4), 447–457.
- Boyd, S. & Mattingley, J. (2007), 'Branch and bound methods', *Notes for EE364b, Stanford University* pp. 2006–07.
- Bramel, J. & Simchi-Levi, D. (1995), 'A location based heuristic for general routing problems', *Operations research* **43**(4), 649–660.
- Bräysy, O. & Gendreau, M. (2002), 'Tabu search heuristics for the vehicle routing problem with time windows', *Top* **10**(2), 211–237.
- Brimberg, J. & Love, R. F. (1998), 'Solving a class of two-dimensional uncapacitated location-allocation problems by dynamic programming', *Operations Research* **46**(5), 702–709.
- Castellucci, P. B. (2017), 'Julia e jump: Novas ferramentas para programação matemática', *Pesquisa Operacional para o Desenvolvimento* **9**(2), 48–61.
- Christofides, N. (1979), *The vehicle routing problem*, Combinatorial optimization.
- Christofides, N. & Beasley, J. E. (1984), 'The period routing problem', *Networks* **14**(2), 237–256.
- Christofides, N., Mingozzi, A. & Toth, P. (1981), 'Exact algorithms for the vehicle routing problem, based on spanning tree and shortest path relaxations', *Mathematical programming* **20**(1), 255–282.
- Clarke, G. & Wright, J. W. (1964), 'Scheduling of vehicles from a central depot to a number of delivery points', *Operations research* **12**(4), 568–581.
- Cooper, L. & Cooper, M. W. (2016), *Introduction to Dynamic Programming: International Series in Modern Applied Mathematics and Computer Science, Volume 1*, Vol. 1, Elsevier.
- Cordeau, J.-F., Gendreau, M., Laporte, G., Potvin, J.-Y. & Semet, F. (2002), 'A guide to vehicle routing heuristics', *Journal of the Operational Research society* **53**(5), 512–522.

- Cordeau, J.-F., Laporte, G., Savelsbergh, M. W. & Vigo, D. (2007), 'Vehicle routing', *Handbooks in operations research and management science* **14**, 367–428.
- Crevier, B., Cordeau, J.-F. & Laporte, G. (2007), 'The multi-depot vehicle routing problem with inter-depot routes', *European journal of operational research* **176**(2), 756–773.
- Crowder, H. & Padberg, M. W. (1980), 'Solving large-scale symmetric travelling salesman problems to optimality', *Management Science* **26**(5), 495–509.
- Dantzig, G. B. & Ramser, J. H. (1959), 'The truck dispatching problem', *Management science* **6**(1), 80–91.
- De Backer, B., Furnon, V., Prosser, P., Kilby, P. & Shaw, P. (1997), Local search in constraint programming: Application to the vehicle routing problem, in 'Proc. CP-97 Workshop Indust. Constraint-Directed Scheduling', Schloss Hagenberg Austria, pp. 1–15.
- Dedong, W., Qijun, C. et al. (2008), Stochastic vehicle routing problems and their solution algorithm, in '2008 IEEE International Conference on Automation and Logistics', IEEE, pp. 958–962.
- Desrochers, M., Desrosiers, J. & Solomon, M. (1992), 'A new optimization algorithm for the vehicle routing problem with time windows', *Operations research* **40**(2), 342–354.
- Desrosiers, J., Soumis, F. & Desrochers, M. (1984), 'Routing with time windows by column generation', *Networks* **14**(4), 545–565.
- Dondo, R. & Cerdá, J. (2007), 'A cluster-based optimization approach for the multi-depot heterogeneous fleet vehicle routing problem with time windows', *European journal of operational research* **176**(3), 1478–1507.
- Dorigo, M. (1992), 'Optimization, learning and natural algorithms', *Ph. D. Thesis, Politecnico di Milano*.
- Dorigo, M., Birattari, M. & Stützle, T. (2006), 'Ant colony optimization', *IEEE computational intelligence magazine* **1**(4), 28–39.
- Dorigo, M. & Di Caro, G. (1999), Ant colony optimization: a new meta-heuristic, in 'Proceedings of the 1999 congress on evolutionary computation-CEC99 (Cat. No. 99TH8406)', Vol. 2, IEEE, pp. 1470–1477.
- Dorigo, M., Maniezzo, V. & Coloni, A. (1996), 'Ant system: optimization by a colony of cooperating agents', *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)* **26**(1), 29–41.
- Dorigo, M. & Stützle, T. (2019), 'Ant colony optimization: overview and recent advances', *Handbook of metaheuristics* pp. 311–351.
- Dror, M. & Trudeau, P. (1989), 'Savings by split delivery routing', *Transportation Science* **23**(2), 141–145.
- Drucker, P. F. (1962), 'The economy's dark continent', *Fortune* **65**(103), 265–270.
- Dubois-Lacoste, J., Hoos, H. H. & Stützle, T. (2015), On the empirical scaling behaviour of state-of-the-art local search algorithms for the euclidean tsp, in 'Proceedings of the 2015 Annual Conference on Genetic and Evolutionary Computation', pp. 377–384.

- Eilon, S., Watson-Gandy, C. D. T., Christofides, N. & de Neufville, R. (1974), 'Distribution management-mathematical modelling and practical analysis', *IEEE Transactions on Systems, Man, and Cybernetics* pp. 589–589.
- Eksioglu, B., Vural, A. V. & Reisman, A. (2009), 'The vehicle routing problem: A taxonomic review', *Computers & Industrial Engineering* **57**(4), 1472–1483.
- Everitt, B. & Hothorn, T. (2011), *An introduction to applied multivariate analysis with R*, Springer Science & Business Media.
- Feo, T. A. & Resende, M. G. (1989), 'A probabilistic heuristic for a computationally difficult set covering problem', *Operations research letters* **8**(2), 67–71.
- Feo, T. A. & Resende, M. G. (1995), 'Greedy randomized adaptive search procedures', *Journal of global optimization* **6**(2), 109–133.
- Fischer, T., Stützle, T., Hoos, H. & Merz, P. (2005), An analysis of the hardness of tsp instances for two high performance algorithms, in 'Proceedings of the Sixth Metaheuristics International Conference', pp. 361–367.
- Fischetti, M., Toth, P. & Vigo, D. (1994), 'A branch-and-bound algorithm for the capacitated vehicle routing problem on directed graphs', *Operations Research* **42**(5), 846–859.
- Fisher, M. L. (1994), 'Optimal solution of vehicle routing problems using minimum k-trees', *Operations research* **42**(4), 626–642.
- Fisher, M. L. & Jaikumar, R. (1978), *A decomposition algorithm for large-scale vehicle routing*, Department of Decision Sciences, Wharton School, University of Pennsylvania.
- Fisher, M. L. & Jaikumar, R. (1981), 'A generalized assignment heuristic for vehicle routing', *Networks* **11**(2), 109–124.
- Franceschini, F. & Rafele, C. (2000), *Quality evaluation in logistic services*, MCB UP Ltd.
- Francis, P. M., Smilowitz, K. R. & Tzur, M. (2008), The period vehicle routing problem and its extensions, in 'The vehicle routing problem: latest advances and new challenges', Springer, pp. 73–102.
- Francis, P., Smilowitz, K. & Tzur, M. (2005), 'The period vehicle routing problem with service choice; 2004'.
- Gaskell, T. (1967), 'Bases for vehicle fleet scheduling', *Journal of the Operational Research Society* **18**(3), 281–295.
- Gavish, B. & Graves, S. C. (1978), *The travelling salesman problem and related problems*, Massachusetts Institute of Technology, Operations Research Center.
- Geetha, S., Vanathi, P. & Poonthalir, G. (2012), 'Metaheuristic approach for the multi-depot vehicle routing problem', *Applied artificial intelligence* **26**(9), 878–901.

- Gendreau, M., Hertz, A., Laporte, G. & Stan, M. (1998), 'A generalized insertion heuristic for the traveling salesman problem with time windows', *Operations Research* **46**(3), 330–335.
- Gillett, B. E. & Miller, L. R. (1974), 'A heuristic algorithm for the vehicle-dispatch problem', *Operations research* **22**(2), 340–349.
- Glover, F. (1986), 'Future paths for integer programming and links to artificial intelligence', *Computers & operations research* **13**(5), 533–549.
- Goldbarg, M. C. & Luna, H. P. L. (2005), *Otimização combinatória e programação linear: modelos e algoritmos*, Elsevier.
- Golden, B. L., Raghavan, S. & Wasil, E. A. (2008), *The vehicle routing problem: latest advances and new challenges*, Vol. 43, Springer Science & Business Media.
- Gomory, R. E. (2010), Outline of an algorithm for integer solutions to linear programs and an algorithm for the mixed integer problem, in '50 Years of Integer Programming 1958-2008', Springer, pp. 77–103.
- Hadjiconstantinou, E., Christofides, N. & Mingozzi, A. (1995), 'A new exact algorithm for the vehicle routing problem based on q-paths and k-shortest paths relaxations', *Annals of Operations Research* **61**(1), 21–43.
- Hansen, K. H. & Krarup, J. (1974), 'Improvements of the held – karp algorithm for the symmetric traveling salesman problem', *Mathematical Programming* **7**(1), 87–96.
- Held, M. & Karp, R. M. (1971), 'The traveling-salesman problem and minimum spanning trees: Part ii', *Mathematical programming* **1**(1), 6–25.
- Helsgaun, K. (2000), 'An effective implementation of the lin–kernighan traveling salesman heuristic', *European journal of operational research* **126**(1), 106–130.
- Ho, W., Ho, G. T., Ji, P. & Lau, H. C. (2008), 'A hybrid genetic algorithm for the multi-depot vehicle routing problem', *Engineering applications of artificial intelligence* **21**(4), 548–557.
- Holland, J. H. (1962), 'Outline for a logical theory of adaptive systems', *Journal of the ACM (JACM)* **9**(3), 297–314.
- Ignizio, J. P. & Cavalier, T. M. (1994), *Linear programming*, Prentice-Hall, Inc.
- Jaillet, P., Qi, J. & Sim, M. (2016), 'Routing optimization under uncertainty', *Operations research* **64**(1), 186–200.
- Jain, A. K. & Dubes, R. C. (1988), *Algorithms for clustering data*, Prentice-Hall, Inc.
- Jain, A. K., Murty, M. N. & Flynn, P. J. (1999), 'Data clustering: a review', *ACM computing surveys (CSUR)* **31**(3), 264–323.
- Johnson, J. C., Wood, D. F. et al. (1990), *Contemporary logistics*, Macmillan.

- Jourdan, L., Basseur, M. & Talbi, E.-G. (2009), 'Hybridizing exact methods and metaheuristics: A taxonomy', *European Journal of Operational Research* **199**(3), 620–629.
- Jünger, M. & Thienel, S. (1998), 'Introduction to abacus—a branch-and-cut system', *Operations Research Letters* **22**(2-3), 83–95.
- Karaoglan, I., Altıparmak, F., Kara, I. & Dengiz, B. (2012), 'The location-routing problem with simultaneous pickup and delivery: Formulations and a heuristic approach', *Omega* **40**(4), 465–477.
- Kaufman, L. & Rousseeuw, P. J. (2009), *Finding groups in data: an introduction to cluster analysis*, Vol. 344, John Wiley & Sons.
- Kendall, M. G. (1966), 'Discrimination and classification', *Multivariate analysis* **1**.
- Kodinariya, T. M. & Makwana, P. R. (2013), 'Review on determining number of cluster in k-means clustering', *International Journal* **1**(6), 90–95.
- Kotler, P. & Turner, R. E. (1997), *Marketing management: Analysis, planning, implementation, and control*, Vol. 9, Prentice hall Upper Saddle River, NJ.
- Kulkarni, R. & Bhave, P. R. (1985), 'Integer programming formulations of vehicle routing problems', *European journal of operational research* **20**(1), 58–67.
- Kwon, B. C., Eysenbach, B., Verma, J., Ng, K., De Filippi, C., Stewart, W. F. & Perer, A. (2017), 'Cluster-vision: Visual supervision of unsupervised clustering', *IEEE transactions on visualization and computer graphics* **24**(1), 142–151.
- Lacerda, L. (2002), 'Logística reversa: uma visão sobre os conceitos básicos e as práticas operacionais', *Rio de Janeiro: COPPEAD/UFRJ* **6**.
- Laguna, M. & Glover, F. (1993), 'Bandwidth packing: A tabu search approach', *Management science* **39**(4), 492–500.
- Lance, G. N. & Williams, W. T. (1967), 'A general theory of classificatory sorting strategies: 1. hierarchical systems', *The computer journal* **9**(4), 373–380.
- Laporte, G. (1984), 'Optimal solutions to capacitated multidepot vehicle routing problems', *Congressus Nemerantium* **4**, 283–292.
- Laporte, G. (1992), 'The vehicle routing problem: An overview of exact and approximate algorithms', *European journal of operational research* **59**(3), 345–358.
- Laporte, G., Louveaux, F. V. & Van Hamme, L. (2002), 'An integer l-shaped algorithm for the capacitated vehicle routing problem with stochastic demands', *Operations Research* **50**(3), 415–423.
- Laporte, G., Mercure, H. & Nobert, Y. (1986), 'An exact algorithm for the asymmetrical capacitated vehicle routing problem', *Networks* **16**(1), 33–46.

- Laporte, G. & Nobert, Y. (1987), Exact algorithms for the vehicle routing problem, in 'North-Holland Mathematics Studies', Vol. 132, Elsevier, pp. 147–184.
- Laporte, G., Nobert, Y. & Desrochers, M. (1985), 'Optimal routing under capacity and distance restrictions', *Operations research* **33**(5), 1050–1073.
- Lenstra, J. K. & Kan, A. R. (1981), 'Complexity of vehicle routing and scheduling problems', *Networks* **11**(2), 221–227.
- Lim, A. & Wang, F. (2005), 'Multi-depot vehicle routing problem: A one-stage approach', *IEEE transactions on Automation Science and Engineering* **2**(4), 397–402.
- Lin, S. (1965), 'Computer solutions of the traveling salesman problem', *Bell System Technical Journal* **44**(10), 2245–2269.
- Lin, S. & Kernighan, B. W. (1973), 'An effective heuristic algorithm for the traveling-salesman problem', *Operations research* **21**(2), 498–516.
- Little, J. D., Murty, K. G., Sweeney, D. W. & Karel, C. (1963), 'An algorithm for the traveling salesman problem', *Operations research* **11**(6), 972–989.
- Lüer, A., Benavente, M., Bustos, J. & Venegas, B. (2009), El problema de rutas de vehículos: Extensiones y métodos de resolución, estado del arte., in 'EIG'.
- MacQueen, J. et al. (1967), Some methods for classification and analysis of multivariate observations, in 'Proceedings of the fifth Berkeley symposium on mathematical statistics and probability', Vol. 1, Oakland, CA, USA, pp. 281–297.
- Marinakis, Y., Marinaki, M. & Dounias, G. (2010), 'A hybrid particle swarm optimization algorithm for the vehicle routing problem', *Engineering Applications of Artificial Intelligence* **23**(4), 463–472.
- Matai, R., Singh, S. P. & Mittal, M. L. (2010), 'Traveling salesman problem: an overview of applications, formulations, and solution approaches', *Traveling salesman problem, theory and applications* **1**.
- Mateos, D. I. G., Gómez, A. G., Martino, D. D. A., García, J. P. & Fernández, N. G. (2013), 'Desarrollo de un método híbrido para la resolución del MDVRP', *Revista de la Escuela Jacobea de Posgrado* pp. 45–64.
- Mazzeo, S. & Loiseau, I. (2004), 'An ant colony algorithm for the capacitated vehicle routing', *Electronic Notes in Discrete Mathematics* **18**, 181–186.
- Medaglia, A. (2005), 'Combinatoria para logística', *Coloquio en Optimización Combinatoria Sesión Avanzada, Universidad de los Andes* pp. 1–10.
- Mejía, S. E., de Ocampo, L. S. R. & Silva, P. P. B. (2010), 'Análisis de los costos logísticos en la administración de la cadena de suministro', *Scientia et technica* **16**(45), 272–277.
- Melián, B., Pérez, J. A. M. & Vega, J. M. M. (2003), 'Metaheurísticas: Una visión global', *Inteligencia Artificial. Revista Iberoamericana de Inteligencia Artificial* **7**(19), 1–10.

- Miller, C. E., Tucker, A. W. & Zemlin, R. A. (1960), 'Integer programming formulation of traveling salesman problems', *Journal of the ACM (JACM)* **7**(4), 326–329.
- Miller, D. L. (1995), 'A matching based exact algorithm for capacitated vehicle routing problems', *ORSA Journal on Computing* **7**(1), 1–9.
- Mole, R. & Jameson, S. (1976), 'A sequential route-building algorithm employing a generalised savings criterion', *Journal of the Operational Research Society* **27**(2), 503–511.
- Montoya-Torres, J. R., Franco, J. L., Isaza, S. N., Jiménez, H. F. & Herazo-Padilla, N. (2015), 'A literature review on the vehicle routing problem with multiple depots', *Computers & Industrial Engineering* **79**, 115–129.
- Muningsih, E. & Kiswati, S. (2018), 'Sistem aplikasi berbasis optimasi metode elbow untuk penentuan clustering pelanggan', *Joutica* **3**(1), 117–124.
- Nainggolan, R., Perangin-angin, R., Simarmata, E. & Tarigan, A. F. (2019), Improved the performance of the k-means cluster using the sum of squared error (sse) optimized by using the elbow method, in 'Journal of Physics: Conference Series', Vol. 1361, IOP Publishing, pp. 12–15.
- Niazy, N. S. & Badr, A. (2012), 'Complexity of capacitated vehicles routing problem using cellular genetic algorithms', *International Journal of Computer Science and Network Security (IJCSNS)* **12**(2), 1–5.
- Niknejad, A. & Petrovic, D. (2014), 'Optimisation of integrated reverse logistics networks with different product recovery routes', *European Journal of Operational Research* **238**(1), 143–154.
- Nurchahyo, G. W., Alias, R. A., Shamsuddin, S. M. & Sap, M. N. M. (2002), 'Sweep algorithm in vehicle routing problem for public transport', *Jurnal Antarabangsa Teknologi Maklumat* **2**, 51–64.
- Osman, I. H. (1993), 'Metastrategy simulated annealing and tabu search algorithms for the vehicle routing problem', *Annals of operations research* **41**(4), 421–451.
- Ostfeld, A. (2011), *Ant Colony Optimization: methods and applications*, BoD–Books on Demand.
- Petersen, B., Pisinger, D. & Spoorendonk, S. (2008), Chvátal-gomory rank-1 cuts used in a dantzig-wolfe decomposition of the vehicle routing problem with time windows, in 'The vehicle routing problem: latest advances and new challenges', Springer, pp. 397–419.
- Potvin, J.-Y. (1996), 'Genetic algorithms for the traveling salesman problem', *Annals of Operations Research* **63**(3), 337–370.
- Ramírez, A. C. (2009), *Manual de la gestión logística del transporte y distribución de mercancías*, Universidad del Norte.
- Reeves, C. R. (1993), Using genetic algorithms with small populations., in 'ICGA', Vol. 590, p. 92.
- Renaud, J., Laporte, G. & Boctor, F. F. (1996), 'A tabu search heuristic for the multi-depot vehicle routing problem', *Computers & Operations Research* **23**(3), 229–235.

- Riveros, D. P. B. & Silva, P. P. B. (2008), 'Importancia de la administración logística', *Scientia et technica* **1**(38).
- Robusto, C. C. (1957), 'The cosine-haversine formula', *The American Mathematical Monthly* **64**(1), 38–40.
- Rochat, Y. & Semet, F. (1994), 'A tabu search approach for delivering pet food and flour in switzerland', *Journal of the Operational Research Society* **45**(11), 1233–1246.
- Rodríguez, A. & Ruiz, R. (2009), El impacto de la asimetría en la resolución de problemas de distribución y rutas, in 'XIII Congreso de Ingeniería de Organización', pp. 1645–1654.
- Rushton, A., Croucher, P. & Baker, P. (2014), *The handbook of logistics and distribution management: Understanding the supply chain*, Kogan Page Publishers.
- Ryan, D. M., Hjorring, C. & Glover, F. (1993), 'Extensions of the petal method for vehicle routeing', *Journal of the Operational Research Society* **44**(3), 289–296.
- San, O. M., Huynh, V.-N. & Nakamori, Y. (2004), 'An alternative extension of the k-means algorithm for clustering categorical data', *International journal of applied mathematics and computer science* **14**, 241–247.
- Sanches, D., Whitley, D. & Tinós, R. (2017), Improving an exact solver for the traveling salesman problem using partition crossover, in 'Proceedings of the Genetic and Evolutionary Computation Conference', pp. 337–344.
- Savelsbergh, M. (1997), 'A branch-and-price algorithm for the generalized assignment problem', *Operations research* **45**(6), 831–841.
- Secomandi, N. (2000), 'Comparing neuro-dynamic programming algorithms for the vehicle routing problem with stochastic demands', *Computers & Operations Research* **27**(11-12), 1201–1225.
- Selim, S. Z. & Ismail, M. A. (1984), 'K-means-type algorithms: A generalized convergence theorem and characterization of local optimality', *IEEE Transactions on pattern analysis and machine intelligence* **1**, 81–87.
- Seref, O., Fan, Y.-J. & Chaovalitwongse, W. A. (2014), 'Mathematical programming formulations and algorithms for discrete k-median clustering of time-series data', *INFORMS Journal on Computing* **26**(1), 160–172.
- Sobh, T. M. & Vacic, V. (2004), *Vehicle Routing Problems with Time Windows*, Ternopil Academy of National Economy, Institute of Computer Information.
- Solak, S., Scherrer, C. & Ghoniem, A. (2014), 'The stop-and-drop problem in nonprofit food distribution networks', *Annals of Operations Research* **221**(1), 407–426.
- Steinbach, M., Karypis, G. & Kumar, V. (2000), *A comparison of document clustering techniques*, University of Minnesota.

- Sujatha, S. & Sona, A. S. (2013), 'New fast k-means clustering algorithm using modified centroid selection method', *vol 2*, 1–9.
- Syakur, M., Khotimah, B., Rochman, E. & Satoto, B. D. (2018), Integration k-means clustering method and elbow method for identification of the best customer profile cluster, *in 'IOP Conference Series: Materials Science and Engineering'*, Vol. 1, IOP Publishing, pp. 12–17.
- Tan, K. C., Cheong, C. Y. & Goh, C. K. (2007), 'Solving multiobjective vehicle routing problem with stochastic demand via evolutionary computation', *European Journal of operational research* **177**(2), 813–839.
- Tan, P.-N., Steinbach, M. & Kumar, V. (2016), *Introduction to data mining*, Pearson Education India.
- Tansini, L., Urquhart, M. E. & Viera, O. (2001), 'Comparing assignment algorithms for the multi-depot VRP', *Reportes Técnicos 01-08*.
- Tavares, L. G., Lopes, H. S. & Lima, C. R. E. (2009), Construction and improvement heuristics applied to the capacitated vehicle routing problem, *in '2009 World Congress on Nature & Biologically Inspired Computing (NaBIC)'*, IEEE, pp. 690–695.
- Tillman, F. A. (1969), 'The multiple terminal delivery problem with probabilistic demands', *Transportation Science* **3**(3), 192–204.
- Toth, P. & Vigo, D. (1996), A heuristic algorithm for the vehicle routing problem with backhauls, *in 'Advanced methods in transportation analysis'*, Springer, pp. 585–608.
- Toth, P. & Vigo, D. (1997), 'An exact algorithm for the vehicle routing problem with backhauls', *Transportation science* **31**(4), 372–385.
- Toth, P. & Vigo, D. (2002), *The vehicle routing problem*, SIAM.
- Toth, P. & Vigo, D. (2003), 'The granular tabu search and its application to the vehicle-routing problem', *Inform Journal on computing* **15**(4), 333–346.
- Toth, P. & Vigo, D. (2014), *Vehicle routing: problems, methods, and applications*, SIAM.
- Van Breedam, A. (1995), 'Improvement heuristics for the vehicle routing problem based on simulated annealing', *European Journal of Operational Research* **86**(3), 480–490.
- Vance, P. H., Atamturk, A., Barnhart, C., Gelman, E., Johnson, E. L., Krishna, A., Mahidhara, D., Nemhauser, G. L. & Rebello, R. (1997), 'A heuristic branch-and-price approach for the airline crew pairing problem', *Massachusetts Institute of Technology* **3**(10), 215–355.
- Vera, C. A. C. (2005), 'Formulación y solución de un problema de ruteo de vehículos con demanda variable en tiempo real, trasbordos y ventanas de tiempo', *Memoria para optar al título de ingeniero civil matemático, Departamento de Ingeniería Matemática, Universidad de Chile, Santiago de Chile, Chile*.
- Villalobos, A. R. (2006), Grafos: herramienta informática para el aprendizaje y resolución de problemas reales de teoría de grafos., *in 'X Congreso de Ingeniería de Organización'*.

Wren, A. & Carr, J. (1971), 'Computers in transport planning and operation'.

Wren, A. & Holliday, A. (1972), 'Computer scheduling of vehicles from one or more depots to a number of delivery points', *Journal of the Operational Research Society* **23**(3), 333–344.

Yellow, P. (1970), 'A computational modification to the savings method of vehicle scheduling', *Journal of the Operational Research Society* **21**(2), 281–283.

Zhou, L., Baldacci, R., Vigo, D. & Wang, X. (2018), 'A multi-depot two-echelon vehicle routing problem with delivery options arising in the last mile distribution', *European Journal of Operational Research* **265**(2), 765–778.