



UMA META-HEURÍSTICA *ADAPTIVE LARGE NEIGHBORHOOD SEARCH*  
COM MECANISMOS DE PARALELISMO, DETECÇÃO DE ESTAGNAÇÃO E  
PERTURBAÇÕES PARA O PROBLEMA DE ROTEAMENTO DE VEÍCULOS  
COM FROTA HETEROGÊNEA, PERIÓDICO E *MULTI-TRIPS*

Bruno Salezze Vieira

Dissertação de Mestrado apresentada ao Programa de Pós-graduação em Engenharia de Transportes, COPPE, da Universidade Federal do Rio de Janeiro, como parte dos requisitos necessários à obtenção do título de Mestre em Engenharia de Transportes.

Orientador: Glaydston Mattos Ribeiro

Rio de Janeiro  
Fevereiro de 2017

UMA META-HEURÍSTICA *ADAPTIVE LARGE NEIGHBORHOOD SEARCH*  
COM MECANISMOS DE PARALELISMO, DETECÇÃO DE ESTAGNAÇÃO E  
PERTURBAÇÕES PARA O PROBLEMA DE ROTEAMENTO DE VEÍCULOS  
COM FROTA HETEROGÊNEA, PERIÓDICO E *MULTI-TRIPS*

Bruno Salezze Vieira

DISSERTAÇÃO SUBMETIDA AO CORPO DOCENTE DO INSTITUTO  
ALBERTO LUIZ COIMBRA DE PÓS-GRADUAÇÃO E PESQUISA DE  
ENGENHARIA (COPPE) DA UNIVERSIDADE FEDERAL DO RIO DE  
JANEIRO COMO PARTE DOS REQUISITOS NECESSÁRIOS PARA A  
OBTENÇÃO DO GRAU DE MESTRE EM CIÊNCIAS EM ENGENHARIA DE  
TRANSPORTES.

Examinada por:

---

Prof. Glaydston Mattos Ribeiro, D.Sc.

---

Prof. Carlos David Nassi, Dr.Ing.

---

Prof. Geraldo Regis Mauri, D.Sc.

RIO DE JANEIRO, RJ – BRASIL  
FEVEREIRO DE 2017

Vieira, Bruno Salezze

Uma Meta-heurística *Adaptive Large Neighborhood Search* com Mecanismos de Paralelismo, Detecção de Estagnação e Perturbações para o Problema de Roteamento de Veículos com Frota Heterogênea, Periódico e *Multi-Trips*/Bruno Salezze Vieira. – Rio de Janeiro: UFRJ/COPPE, 2017.

XVI, 75 p.: il.; 29, 7cm.

Orientador: Glaydston Mattos Ribeiro

Dissertação (mestrado) – UFRJ/COPPE/Programa de Engenharia de Transportes, 2017.

Referências Bibliográficas: p. 63 – 75.

1. Roteamento de veículos. 2. Frota Mista. 3. Roteamento de Veículos Periódico. 4. *Multi-Trip*. 5. Meta-heurística. 6. *Adaptive Large Neighborhood Search*. 7. ALNS. I. Ribeiro, Glaydston Mattos. II. Universidade Federal do Rio de Janeiro, COPPE, Programa de Engenharia de Transportes. III. Título.

*Dedico este trabalho aos meus pais, Afonso e Auxiliadora, por todo o apoio, confiança e dedicação, fundamentais para tornar realidade mais essa conquista.*

# Agradecimentos

Inicio meus agradecimentos por Deus, já que Ele colocou pessoas tão especiais a meu lado, sem as quais certamente não teria dado conta.

A meus pais, Afonso e Auxiliados, todo meu agradecimento. Sempre me apoiam e me incentivam a ter uma educação de qualidade, e sempre acreditando em minha capacidade. Obrigado.

Ao meu orientador, professor Glaydston Mattos Ribeiro, que sempre esteve próximo criticando e dando sugestões com muita paciência de como prosseguir.

À equipe do Projeto PNCT, professor Rômulo Dante Orrico Filho, Saul Quadros, Cristiane da Penha Bernardo e Gerusa Ravache, por toda amizade, companheirismo e ensinamentos. Participar desses projetos me proporcionou enorme aprendizado, pessoal e profissional, que levarei por toda a vida.

Aos colegas de mestrado e doutorado, especialmente, Tulio Silveira, Marcus Vinícius Oliveira Câmara, Lygia Bronneberg, Bruno Guida, Pedro Basilio, Igor Godeiro, Thayse Ferrari, Ewerton Torres, Victor Albino, Inaê Lucato, Breno Garcia, Pedro Geaquinto e Luiz Saldanha.

A Jane Correa de Souza e Maria Helena Santos Oliveira por todo apoio na secretaria do PET. Sua disposição em ajudar contribuíram para minha formação acadêmica.

Ao DNIT e ao COTER pelas informações e sugestões que contribuíram para o aperfeiçoamento da pesquisa.

À Coordenação de Aperfeiçoamento de Pessoal de Nível Superior – CAPES, pelo auxílio concedido.

Aos professores Carlos David Nassi e Geraldo Regis Mauri por aceitarem participar da banca e por toda contribuição para aprimoramento da pesquisa.

Resumo da Dissertação apresentada à COPPE/UFRJ como parte dos requisitos necessários para a obtenção do grau de Mestre em Ciências (M.Sc.)

UMA META-HEURÍSTICA *ADAPTIVE LARGE NEIGHBORHOOD SEARCH*  
COM MECANISMOS DE PARALELISMO, DETECÇÃO DE ESTAGNAÇÃO E  
PERTURBAÇÕES PARA O PROBLEMA DE ROTEAMENTO DE VEÍCULOS  
COM FROTA HETEROGÊNEA, PERIÓDICO E *MULTI-TRIPS*

Bruno Salezze Vieira

Fevereiro/2017

Orientador: Glaydston Mattos Ribeiro

Programa: Engenharia de Transportes

A atribuição e o planejamento de rotas de veículos são problemas importantes envolvidos nas cadeias de suprimentos. Em ambiente real é comum encontrar situações que envolvam uma quantidade muito grande de clientes ou de restrições que conseqüentemente fogem do alcance de métodos exatos. Neste contexto, este trabalho apresenta uma meta-heurística capaz de resolver algumas variantes do problema de roteamento de veículos (PRV) combinadas: o PRV capacitado com frota heterogênea, o PRV periódico e o PRV com *multi-trips*. A meta-heurística escolhida, denominada *Adaptive Large Neighborhood Search* (ALNS), combina a força de estratégias bem-sucedidas na literatura como busca em vizinhança ampla e mecanismos adaptativos e também novos mecanismos como paralelismo, detecção de estagnação e perturbações. O ALNS foi implementado de tal maneira que todas as variantes do PRV citadas pudessem ser resolvidas sem alterações de código. Os resultados obtidos, em diversas instâncias propostas na literatura foram satisfatórios, mostrando o bom desempenho do método proposto.

Abstract of Dissertation presented to COPPE/UFRJ as a partial fulfillment of the requirements for the degree of Master of Science (M.Sc.)

AN METAHEURISTIC ADAPTIVE LARGE NEIGHBORHOOD SEARCH  
WITH MECHANICS OF PARALLELISM, DETECTION OF STAGNATION  
AND DISTURBANCES FOR THE HETEROGENEOUS VEHICLE ROUTING  
PROBLEM, PERIODIC AND MULTI-TRIPS

Bruno Salezze Vieira

February/2017

Advisor: Glaydston Mattos Ribeiro

Department: Transportation Engineering

The planning of vehicle routes is a major issue involved in supply chains. In real environment we can find situations involving a very large number of clients or constraints which indicate that exact methods should be avoided. In this context, this work presents a metaheuristic for solving some variants of the vehicle routing problem (VRP): Heterogeneous VRP, VRP Periodic and VRP with multi-trips. The metaheuristic chosen, called Adaptive Large Neighborhood Search (ALNS), combines the power of successful strategies in the literature as a large neighborhood search and adaptive mechanisms with new features such as parallelism, detection of stagnation and perturbations. Our ALNS was implemented in such a way that all variants of the VRP are solved without changes in the code. The results for several instances proposed in the literature are satisfactory, showing the good performance of the approach.

# Sumário

<b>Lista de Figuras</b>	<b>x</b>
<b>Lista de Tabelas</b>	<b>xi</b>
<b>Lista de Símbolos</b>	<b>xii</b>
<b>Lista de Abreviaturas</b>	<b>xiv</b>
<b>1 Introdução</b>	<b>1</b>
1.1 Caracterização do Problema . . . . .	3
1.2 Objetivos Geral e Específicos . . . . .	4
1.3 Motivação e Justificativa . . . . .	5
1.4 Metodologia . . . . .	5
1.5 Organização do Trabalho . . . . .	6
<b>2 Revisão Bibliográfica</b>	<b>7</b>
2.1 O Problema de Roteamento de Veículos . . . . .	7
2.2 Métodos Clássicos de Solução para o PRV . . . . .	8
2.2.1 Heurísticas Construtivas . . . . .	9
2.2.2 Heurísticas de Melhoria . . . . .	10
2.2.3 Meta-heurísticas . . . . .	10
2.2.4 Métodos Exatos . . . . .	12
2.3 Variantes Estudadas do PRV . . . . .	16
2.3.1 PRV com Frota Heterogênea . . . . .	16
2.3.2 PRV Periódico . . . . .	22
2.3.3 PRV <i>Multi-Trips</i> . . . . .	25
2.4 Considerações Finais . . . . .	27
<b>3 <i>Adaptive Large Neighborhood Search</i> (ALNS)</b>	<b>28</b>
3.1 Mecanismos e Funcionamento Geral do ALNS para o PRV-FH-P-MT	30
3.2 Descrição do ALNS para o PRV-FP-P-MT . . . . .	31



3.2.1	Função Objetivo e Representação das Variações do PRV estudadas . . . . .	31
3.2.2	Solução Inicial . . . . .	33
3.2.3	Heurísticas de Remoção e Inseção . . . . .	33
3.2.4	Camada Adaptativa . . . . .	36
3.2.5	Temperatura . . . . .	37
3.2.6	Estratégias de Paralelismo para Avaliar Estagnação e Perturbações . . . . .	38
3.2.7	Código Principal do ALNS . . . . .	39
3.3	Considerações Finais . . . . .	40
<b>4</b>	<b>Testes Computacionais</b>	<b>43</b>
4.1	Calibração dos Parâmetros . . . . .	44
4.1.1	Número de Elementos Removidos e Re-inseridos por Iteração . . . . .	45
4.1.2	Tamanho do Segmento . . . . .	45
4.1.3	Impacto dos Mecanismos de Perturbação . . . . .	46
4.1.4	Número de Soluções Simultâneas . . . . .	48
4.1.5	Taxas de Atualização dos Pesos das Penalidades . . . . .	49
4.1.6	Número Máximo de Segmentos e Iterações . . . . .	49
4.1.7	Resultado da Calibração . . . . .	50
4.2	Instâncias do PRV Clássico . . . . .	50
4.3	Instâncias do PRV Periódico . . . . .	52
4.3.1	Instâncias do PRV com Múltiplos Depósitos . . . . .	53
4.4	Instâncias do PRV com Frota Heterogênea . . . . .	55
4.5	Instâncias do PRV <i>Multi-trip</i> . . . . .	57
<b>5</b>	<b>Considerações Finais</b>	<b>61</b>
	<b>Referências Bibliográficas</b>	<b>63</b>

# Lista de Figuras

1.1	Exemplo de uma possível solução para um problema de roteamento de veículos clássico. . . . .	2
1.2	Ilustração de diversos tipos de veículos que realizam atividades de entregas ou coletas de produtos. . . . .	4
2.1	Exemplo de um problema sendo resolvido por <i>Branch and Bound</i> . . .	14
2.2	Evolução dos modelos e métodos para solução do PRVP (FRANCIS <i>et al.</i> [1]). . . . .	23
3.1	Comportamento oscilante das soluções correntes ao longo dos segmentos na Instância <i>p5</i> do PRVP. . . . .	39
4.1	Impacto dos intervalos de $\gamma$ ao longo dos segmentos na Instância <i>p5</i> do PRVP. . . . .	45
4.2	Impacto dos intervalos de $\gamma$ em relação ao tempo gasto e o custo das soluções na Instância <i>p5</i> do PRVP. . . . .	46
4.3	Impacto do tamanho do segmento $\psi$ em relação ao custo médio das soluções na Instância <i>p5</i> do PRVP. . . . .	47
4.4	Impacto dos mecanismos de perturbação em relação ao custo médio das soluções na Instância <i>p5</i> do PRVP. . . . .	47
4.5	Impacto de diferentes valores de $\eta$ em relação ao custo médio e tempo de solução na Instância <i>p5</i> do PRVP. . . . .	48
4.6	Impacto de $\alpha^-$ relação ao custo médio das soluções na Instância <i>p5</i> do PRVP. . . . .	49
4.7	Impacto de $\alpha^+$ em relação ao custo médio das soluções na Instância <i>p5</i> do PRVP. . . . .	50
4.8	Valor médio da melhor solução encontrada ao longo dos segmentos na Instância <i>p5</i> do PRVP. . . . .	50

# Lista de Tabelas

4.1	Valor utilizado dos parâmetros para todas as instâncias . . . . .	51
4.2	Instâncias do PRVC. . . . .	52
4.3	Primeiro Grupo de Instâncias do PRVP. . . . .	54
4.4	Segundo Grupo de Instâncias do PRVP. . . . .	54
4.5	Instâncias do PRV com Múltiplos Depósitos . . . . .	56
4.6	Instâncias do PRVFH. . . . .	58
4.7	Instâncias do PRVFH com Múltiplas Viagens . . . . .	60

# Lista de Símbolos

$A$	Conjunto Elite, p. 38
$K$	Número Máximo de Iterações, p. 38
$T$	Temperatura, p. 30
$T_0$	Temperatura Inicial, p. 30
$T_F$	Temperatura Final, p. 30
$\alpha^+$	Taxa de Atualização Viável de Peso de Penalidade, p. 32
$\alpha^-$	Taxa de Atualização Inviável de Peso de Penalidade, p. 32
$\alpha_T$	Taxa de Atualização da Temperatura, p. 30
$\alpha_A$	Taxa de Recálculo de Pesos, p. 36
$\epsilon$	Tamanho do Conjunto Elite, p. 38
$\eta$	Número de Execuções Paralelas do Algoritmo, p. 31
$\gamma$	Número de Requisições a serem removidas, p. 30
$\omega_0^q$	Peso Inicial da Penalidade de Sobrecarga, p. 32
$\omega_0^t$	Peso Inicial da Penalidade de Sobretempo, p. 32
$\omega_q$	Peso da Penalidade de Sobrecarga, p. 32
$\omega_t$	Peso da Penalidade de Sobretempo, p. 32
$\phi$	Constante entre 0 e 1, p. 34
$\psi$	Tamanho do Segmento, p. 31
$\rho$	Limiar de Estagnação, p. 38
$\tau_1$	Pontuação por Melhorar a Melhor Solução, p. 36

$\tau_2$	Pontuação por Melhorar a Solução Atual, p. 36
$\tau_3$	Pontuação por Passar no Critério do SA, p. 36
$K\text{-Regret}_i$	Valor do “k-arrependimento” para a requisição $i$ , p. 36
$\theta$	Número de Soluções Perturbadas, p. 31
$c(r)$	Custo da Rota $r$ , p. 32
$f'(s)$	Valor da Função Objetivo da solução $s$ , p. 32
$f(s)$	Valor da Função Objetivo Penalizada da solução $s$ , p. 30
$t(r)$	Tempo da Rota $r$ , p. 32
$w_0$	Valor Inicial dos Pesos Adaptativos, p. 31
$w_f$	Valor Residual em $K/3$ Iterações de $\omega_0$ , p. 31
$w_i$	Peso Adaptativo de Seleção da Heurística $i$ , p. 31

# Lista de Abreviaturas

ALNS	<i>Adaptive Large Neighborhood Search</i> , p. 3
HGSADC	<i>Hybrid Genetic Search with Adaptive Diversity Control</i> , p. 12
ILS	<i>Iterated Local Search</i> , p. 11
LB	Limitante Inferior ou <i>Lower Bound</i> , p. 14
LNS	<i>Large Neighborhood Search</i> , p. 28
PPI	Problema de Programação Inteira, p. 13
PRV-FH-P-MT	Problema de Roteamento de Veículos com Frota Heterogênea, Periódico e <i>Multi-Trips</i> , p. 2
PRVFH	Problema de Roteamento de Veículos com Frota Heterogênea, p. 4
PRVHFF	Problema de Roteamento de Veículos Heterogêneos com Frota Fixa, p. 17
PRVMDF	Problema de Roteamento de Veículos Misto e Dimensionamento de Frota, p. 17
PRVMD	Problema de Roteamento de Veículos com Múltiplos Depósitos, p. 12
PRVMT	Problema de Roteamento de Veículos <i>Multi-Trips</i> , p. 4
PRVP	Problema de Roteamento de Veículos Periódico, p. 4
PRV	Problema de Roteamento de Veículos, p. 1
SA	<i>Simulated Annealing</i> , p. 11
TSP	<i>Travelling Salesman Problem</i> , p. 8
TS	<i>Tabu Search</i> , p. 11

UB	Limitante Superior ou <i>Upper Bound</i> , p. 14
UHGS	<i>Unified Hybrid Genetic Search</i> , p. 12
VRP	<i>Vehicle Routing Problem</i> , p. 1

# Lista de Algoritmos

3.1	Heurística de Remoção Shaw baseada na Função Objetivo . . . . .	34
3.2	Heurística de Remoção do Pior . . . . .	35
3.3	Heurística de Inserção $k - Regret$ . . . . .	36
3.4	Adaptive Large Neighborhood Search . . . . .	41
3.5	<i>CalculaSolucoesPerturbadas()</i> . . . . .	42



# Capítulo 1

## Introdução

A questão do roteamento de veículos tem grande importância na literatura, e este vem sendo amplamente estudado por sua relevância para a indústria e economia. O seu estudo teve início com o Problema de Roteamento de Veículos (PRV ou VRP da sigla em inglês para *Vehicle Routing Problem*). Nesse problema, veículos de mesmas características e capacidades saem de um centro de distribuição (ou depósito) e fazem entregas ou coletas em locais específicos (que podem ser clientes ou fornecedores), ao menor custo possível. O custo neste caso está associado basicamente à distância percorrida (GOLDEN *et al.* [2]).

A Figura 1.1 ilustra uma possível solução de um PRV simples com quatro rotas resultantes sendo que cada veículo sai do depósito, atende a todos os clientes de sua rota e retorna ao ponto inicial.

Com a evolução do problema na literatura e tendo como objetivo aproximar-se mais da realidade vivida pelas indústrias e pelas transportadoras, o problema foi enriquecido com mais restrições, características de clientes, frotas e rotas, tornando-se muito complexo (TOTH e VIGO [3]).

Desta forma, existem diversas variantes do PRV clássico. De acordo com TOTH e VIGO [3], as mais encontradas são: adição de janelas de tempo para atendimento que envolvem clientes e centros de distribuição (LIU e SHEN [4]); jornada de trabalho dos motoristas (TOTH e VIGO [3]); frota de veículos heterogênea, cuja determinação do tamanho e composição é parte do problema (LIU e SHEN [4]); entregas periódicas por cliente (CORDEAU *et al.* [5]); múltiplos armazéns (VIDAL *et al.* [6]); entre outras características estudadas.

Em HOFF *et al.* [7], foi apresentado um extenso levantamento bibliográfico sobre os diferentes tipos de roteamento de veículos e metodologias utilizadas, terminando com uma análise crítica da produção científica. Apesar de citar positivamente essa adição de restrições e características reais ao problema, os autores salientam que a literatura específica ainda está longe de representar satisfatoriamente às situações reais.

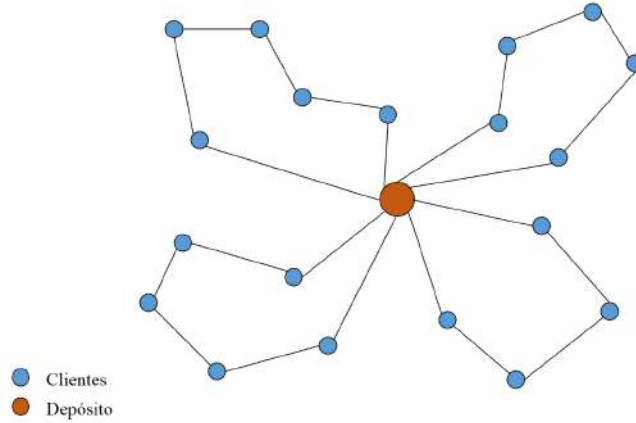


Figura 1.1: Exemplo de uma possível solução para um problema de roteamento de veículos clássico.

Nesta linha de pensamento, essa dissertação aborda o PRV no qual o custo de transporte é proporcional à distância percorrida pelos veículos, somado aos custos fixos de cada um deles, sendo que os veículos, que possuem capacidades diferentes (frota heterogênea), possuem uma jornada de trabalho fixa por dia, podendo voltar ao depósito e atender mais clientes em um mesmo dia, característica esta denominada de *multi trips* (múltiplas viagens). Deve-se ainda programar coletas ao longo de um determinado período de dias e cada cliente possui requisições com frequências diferentes. Por exemplo, um cliente pode solicitar três coletas semanais sem determinar em quais dias úteis elas devem ocorrer exatamente. Sendo assim, o PRV abordado nesta dissertação de mestrado é denominado Problema de Roteamento de Veículos com Frota Heterogênea, Periódico e *Multi-Trips* (PRV-FH-P-MT).

Ao analisar a literatura sobre o problema foco desta pesquisa, verificou-se que não existem trabalhos que abordam todos esses elementos citados de maneira conjunta. O mais próximo encontrado foi o de COELHO *et al.* [8] no qual foi estudado o PRV com frota heterogênea e *multi trips*, sendo que o número máximo de viagens por dia foi restrito a dois.

PRVs são problemas difíceis de serem resolvidos de forma exata e, com isso, heurísticas são indicadas. O termo heurística provê do grego *heuriskein* que significa descobrir. Uma heurística é um procedimento algorítmico desenvolvido por meio de um modelo cognitivo, usualmente com regras baseadas na experiência dos desenvolvedores (CORDENONSI [9]). ZANAKIS *et al.* [10] justificam o uso de métodos heurísticos por sua habilidade de produzir rapidamente soluções próximas da ótima em complexos problemas de otimização com tempos computacionais reduzidos.

Sendo assim, esta dissertação de mestrado se propõe a implementar a meta-heurística *Adaptive Large Neighborhood Search* (ALNS) para o PRV-FH-P-MT, devido ao seu bom desempenho em PRVs e otimização combinatória em geral (ROPKE

e PISINGER [11]). O ALNS proposto possui mecanismos que, segundo a revisão da literatura, até então não haviam sido utilizados em implementações do método como função objetivo com penalidades dinâmicas, paralelismo, detecção de estagnação e perturbações ao longo do processo de busca.

## 1.1 Caracterização do Problema

Como mencionado anteriormente, o problema estudado é o de roteamento de veículos com várias características que o torna mais próximo da realidade de empresas que prestam serviços de coletas (ou entregas) utilizando uma frota, para uma região ao longo de um determinado período de tempo, normalmente, uma semana.

De maneira mais clara, o problema alvo desta pesquisa possui as seguintes características:

- Clientes dispersos geograficamente com posições pré-determinadas para a definição das rotas;
- Demandas periódicas não simétricas. Cada cliente pode possuir demandas individuais, com frequências também individuais; e
- Frota heterogênea de veículos que deixam um único depósito ou centro de distribuição.

Existem diferentes tipos de veículos, com diferentes capacidades e custos que podem ser empregados para realizar as coletas, como ilustrado na Figura 1.2. Neste problema, quanto maior o veículo, maior é o seu custo fixo (custo de utilização imediata do equipamento) (KOÇ *et al.* [12]). O centro de distribuição possui uma determinada quantidade de cada tipo de veículo e, sendo assim, deve-se definir as rotas de atendimento respeitando algumas características:

- Jornada de trabalho das equipes de campo: O atendimento deve respeitar os horários disponíveis para atendimento dos clientes (janelas de tempo), a janela de tempo do centro de distribuição, e o número máximo de horas que um motorista pode trabalhar diariamente;
- Múltiplas viagens de cada veículo por dia (*Multi Trips*): Um veículo pode deixar o centro de distribuição, realizar todas as coletas previstas e retornar ao ponto inicial em um espaço de tempo menor que o seu período útil (jornada de trabalho da equipe). Logo este mesmo veículo está apto a realizar uma nova viagem; e



Figura 1.2: Ilustração de diversos tipos de veículos que realizam atividades de entregas ou coletas de produtos.

- Programação periódica: A programação das rotas deve ser feita para um determinado horizonte de planejamento que, no caso mais comum, é de uma semana.

Assim, PRV-FH-P-MT abordado neste trabalho pode ser visto como uma junção dos seguintes problemas de roteamento: Problema de Dimensionamento e Roteamento de Veículos com Frota Heterogênea (*Fleet Size and Mix Vehicle Routing Problem*); Problema de Roteamento de Veículos Periódico (*Periodic Vehicle Routing Problem*); e Problema de Roteamento de Veículos com *Multi Trips* (*Multi Trip Vehicle Routing Problem*).

LIU e SHEN [4], CORDEAU *et al.* [5] e BRANDAO e MERCER [13] afirmam que cada um desses problemas, separadamente, é *NP-Hard*. Deste modo, por redução polinomial, a intercessão de todos pode ser definida também como *NP-Hard*, ou seja, é pouco provável que exista um algoritmo de complexidade polinomial que forneça uma solução ótima (FLORIAN *et al.* [14]).

## 1.2 Objetivos Geral e Específicos

O objetivo geral desta dissertação consiste em implementar uma meta-heurística ALNS que utiliza paralelismo, penalidades dinâmicas e perturbações para resolver o PRV-FH-P-MT. Como objetivos específicos, buscou-se:

1. Analisar novas estratégias de busca para o ALNS;
2. Gerar ou obter instâncias públicas para efeito de comparação com outros métodos heurísticos ou exatos; e

3. Fazer testes computacionais em instâncias conhecidas para analisar o desempenho do ALNS.

### 1.3 Motivação e Justificativa

Este trabalho iniciou por uma demanda de um problema real de uma grande fabricante de automóveis localizada no interior do Estado de Minas Gerais. Esta demanda apresentou exatamente as características citadas na Seção 1.1 com poucas restrições a mais como a de alguns clientes não podem ser atendidos por certos tipos de veículos. Desta maneira decidiu-se investigar o problema e propor uma estratégia de solução que unificasse as variantes do PRV, para posterior comparação com a literatura. Cabe destacar que a fabricante não disponibilizou dados referente ao problema.

O Problema de Roteamento de Veículos é um problema recorrente na área de logística do setor de transportes, que envolve diretamente custos. Busca-se assim, achar o equilíbrio ideal entre a utilização dos veículos e o atendimento dos clientes, pois sabe-se que rotas menores implicam positivamente na conservação dos veículos, produzem um menor consumo de combustível e uma menor emissão de gases do efeito estufa, mas por outro aumentam o número de veículos circulando que impactam negativamente na depreciação dos equipamentos e no fluxo nas cidades (TOTH e VIGO [3]).

Do ponto de vista científico, como evolução do estado da arte que envolve roteamento de veículos, busca-se nesta dissertação de mestrado propor um bom método de solução para um problema mais próximo da realidade vivenciado pelas empresas que possa ser empregado por elas.

### 1.4 Metodologia

A pesquisa realizada é classificada, sob o ponto de vista do delineamento adotado, como experimental, uma vez que experimentos são realizados para testar os algoritmos manipulando-se variáveis para criar situações de interesse com o propósito de analisar e avaliar os métodos desenvolvidos. A pesquisa é de natureza quantitativa, uma vez que as variáveis envolvidas foram quantificadas segundo uma escala de intervalo ou de razão. Do ponto de vista do seu objetivo, o presente estudo pode ser classificado como uma pesquisa explicativa, o qual se pretende aprofundar no conhecimento dos fenômenos, identificar relações de causa e efeito, descobrir o motivo de dificuldades e propor soluções.

Os procedimentos técnicos envolveram um estudo sobre o estado da arte, baseado em pesquisa bibliográfica sobre o tema em estudo, e em outras etapas, descritas a

seguir:

1. Estudos sobre os PRVs com ênfase no PRV com frota heterogênea, periódico e *multi-trips*;
2. Estudos com foco nos métodos existentes para resolver as variantes de interesse do problema;
3. Análise e definição dos aspectos do ALNS; e
4. Implementação e teste do ALNS.

A ferramenta de programação escolhida para implementar o ALNS foi a linguagem de programação C, dado o conhecimento prévio do autor em questão, e pelo fato de ser uma linguagem de alto nível com bom desempenho. Além disso, este autor possui experiência com o ALNS, conforme pode ser visualizado em RIBEIRO *et al.* [15].

## 1.5 Organização do Trabalho

Neste trabalho, o Capítulo 2 faz uma revisão bibliográfica começando por detalhar o que é o Problema de Roteamento de Veículos na Seção 2.1. Em seguida, detalha os principais métodos de solução para o PRV (Seção 2.2) e também detalha as variantes estudadas do PRV (Seção 2.3). O Capítulo 3 apresenta todo o funcionamento do ALNS proposto neste trabalho, com os mecanismos de funcionamento na Seção 3.1 e os detalhes da implementação na Seção 3.2. O Capítulo 4 apresenta os testes computacionais. Inicialmente é apresentada a calibração dos parâmetros do ALNS (Seção 4.1) e, em seguida, todos os resultados obtidos, para instâncias da literatura, são exibidos nas Seções 4.2, 4.3, 4.3.1 e 4.4. Com os resultados das instâncias propostas de PRVMT na Seção 4.5. Finalmente, o Capítulo 5 discute as considerações finais deste trabalho e destaca possibilidades de trabalhos futuros.

# Capítulo 2

## Revisão Bibliográfica

Na literatura, o tema de roteamento veículos é bastante estudado e de grande relevância por estar relacionado à logística, transportes e toda a cadeia de suprimentos, com aplicações na indústria, rede bancária, governos, exércitos, etc. Portanto, é relevante para diversas áreas e possui muitas aplicações práticas (TOTH e VIGO [3]).

Da mesma forma, o uso de heurísticas para a solução de problemas da área de logística já se tornou prática comumente encontrada, uma vez que a complexidade crescente dos problemas faz com que não possa ser encontrada a solução ótima em um tempo de execução aceitável (GOLDEN *et al.* [2]).

Sendo assim, apresenta-se a seguir alguns conceitos básicos sobre o PRV, e os primeiros métodos utilizados para resolvê-lo. Em seguida, apresenta-se uma revisão bibliográfica dos principais métodos de solução propostos para PRVs na literatura entre heurísticas, meta-heurísticas e métodos exatos. Finalmente, detalha-se os principais PRVs citados neste trabalho no qual são destacados os principais métodos de solução utilizados na literatura para cada um deles.

### 2.1 O Problema de Roteamento de Veículos

Como descrito no Capítulo 1, o PRV busca encontrar um conjunto de rotas para uma frota homogênea visando o atendimento de um conjunto de clientes (pontos de demanda) ao menor custo possível. As principais restrições neste caso são:

- Capacidade dos veículos;
- Cada rota deve iniciar e terminar no centro de distribuição; e
- Todos os clientes devem ser atendidos uma única vez.

A primeira abordagem foi feita por DANTZIG e RAMSER [16] para um problema real de distribuição de combustível de um terminal para diferentes postos.

Neste artigo é proposto um modelo de programação linear inteira, com base no utilizado para o Problema do Caixeiro Viajante (*Travelling Salesman Problem* (TSP)) com adições de restrições de capacidade de transporte.

Dada a dificuldade do PRV, a experiência com heurísticas para solucioná-lo tem como marco a heurística das economias (*Savings Heuristics*), proposta por CLARKE e WRIGHT [17]. Essa heurística ainda é bastante utilizada por ser flexível o suficiente para permitir adaptações e inclusão de restrições para solução de problemas mais complexos.

A heurística das economias gera rotas que respeitam as restrições citadas anteriormente. Ela inicia no cenário em que cada veículo sai do centro de distribuição para atender um único cliente e retorna. A partir daí, passa a buscar economias unindo rotas e diminuindo a distância e tempo percorridos. Existem dois tipos de procedimentos para fazer isso: o paralelo e o sequencial. No método proposto em CLARKE e WRIGHT [17], busca-se a maior economia possível considerando todas as rotas existentes no problema. Assim, em cada iteração, são unidas em par as rotas que juntas não extrapolem a capacidade do veículo e que gerem a maior economia. Já no trabalho de DANTZIG e RAMSER [16], uma rota é estendida até seu limite, para depois se analisar outra rota.

Os trabalhos de DANTZIG e RAMSER [16] e CLARKE e WRIGHT [17] também representam as duas linhas de pensamento heurístico dentro da solução de problemas de roteamento de veículos mais recorrentes. A primeira de DANTZIG e RAMSER [16] é chamada de “agrupar primeiro e rotear depois”, e a segunda de CLARKE e WRIGHT [17], é chamada de “rotear primeiro e dividir depois”. Estas questões foram largamente discutidas em LAPORTE e SEMET [18] e depois amplamente analisada por PRINS *et al.* [19]. PRINS *et al.* [19] concluíram que, atualmente, os métodos de roteamento mais eficientes por tempo de processamento são os de natureza “rotear primeiro e dividir depois”.

Uma vez definido o PRV, bem como seus primeiros métodos de solução, na Seção 2.2, são apresentados com maiores detalhes os métodos de solução heurísticos e meta-heurísticos e exatos que surgiram ao longo do tempo para problemas de roteamentos de veículos em geral. E nas seções 2.3.1, 2.3.2 e 2.3.3 são detalhados os melhores métodos de solução propostos, também ao longo do tempo, para cada uma das variações do PRV abordadas neste trabalho.

## 2.2 Métodos Clássicos de Solução para o PRV

De acordo com CORDENONSI [9], uma heurística é um procedimento algorítmico que busca boas soluções sem o compromisso de obter a melhor solução possível para um dado problema. Ao contrário dos métodos exatos, que buscam encontrar



uma forma algorítmica de achar a melhor solução (conhecida como solução ótima) por meio da combinação ou busca de todas as soluções possíveis, as heurísticas normalmente tendem a apresentar um certo grau de conhecimento acerca do comportamento do problema, gerando um número muito menor de soluções.

Atualmente, mesmo com vários algoritmos sofisticados de decomposição matemática encontrando as soluções ótimas de muitos PRVs, apenas problemas pequenos com até uma centena de clientes são resolvidos, após isso, mesmo com a adição de um único cliente, o tempo de solução pode crescer demais, tornando inviável o seu uso. No entanto, PRVs reais podem possuir milhares de clientes e assim, torna-se necessário o desenvolvimento de heurísticas e meta-heurísticas (TOTH e VIGO [3]).

A história das heurísticas para o PRV é tão antiga quanto o problema em si, desde DANTZIG e RAMSER [16] que desenharam uma simples heurística baseada na união de vértices por meio da solução de problemas lineares e da eliminação de soluções fracionadas. Desde então, uma grande variedade de heurísticas construtivas e de melhoria (busca local) foram propostas, culminando nos anos mais recentes no desenvolvimento de poderosas e flexíveis meta-heurísticas capazes de encontrar soluções a no máximo um por cento do que os melhores resultados conhecidos GOLDEN *et al.* [2].

As heurísticas podem ser divididas, em termos teóricos, em construtivas, de melhorias e meta-heurísticas, sendo que as duas primeiras deram origem ao surgimento da terceira que tem sido foco de pesquisa nos últimos 10 anos (PRINS *et al.* [19]).

### 2.2.1 Heurísticas Construtivas

Uma heurística construtiva, ou míope, para o PRV, consiste em tentar encontrar uma boa rota, considerando, a cada iteração, somente o próximo passo, ou seja, o critério de escolha é basicamente no melhor candidato daquele passo como dito em CAMPELLO e MACULAN [20]. Ela parte de uma solução vazia e constrói a rota, inserindo sempre um cliente de cada vez, até obter uma rota completa. Algoritmos construtivos não possuem nenhum esquema de *backtracking*, ou seja, após inserir um cliente, não é possível retirá-lo da rota (CORDENONSI [9]). A heurística das economias de CLARKE e WRIGHT [17], citada anteriormente, trata-se de uma heurística construtiva com boa aceitação devido à sua simplicidade (LAPORTE [21]).

Atualmente, as heurísticas de construção são aplicadas para a obtenção de soluções iniciais que passam então pelas heurísticas de melhoria que, por sua vez, procuram melhorá-la.

## 2.2.2 Heurísticas de Melhoria

As heurísticas de melhoria, por sua vez, iniciam seu processo de busca a partir de uma solução inicial. Uma rota é considerada viável se ela satisfaz todas as restrições presentes no problema. Desta forma, a heurística de melhoria trabalha no aprimoramento da solução atual, por meio da realização de passos sucessivos de melhoria (CORDENONSI [9]).

Usualmente, estes passos realizam a exclusão e inclusão de arcos ou vértices à rota em questão, de forma a pesquisar a vizinhança da solução em busca de uma rota de melhor qualidade. O termo vizinhança se refere a rotas que se encontram próximas no espaço de busca das soluções, que podem ser alcançadas por meio de um movimento. Por movimento, em um espaço de busca, entende-se a aplicação de uma regra ou função que altere a solução atual, gerando uma nova solução. Um dos movimentos mais simples é o *swap*, representado pela simples troca de posição entre dois clientes. Por exemplo, em um problema de quatro clientes, as rotas  $\{A, B, C, D, A\}$  e  $\{A, B, D, C, A\}$  são consideradas vizinhas, pois é possível obter a segunda rota pela simples troca das posições dos clientes C e D (*swap*) da primeira rota. No entanto, as rotas  $\{A, B, C, D, A\}$  e  $\{A, D, C, B, A\}$  exigem um maior número de trocas para que sejam consideradas vizinhas por meio do movimento *swap*. É importante observar que, para movimentos diferentes, a estrutura de vizinhança também se altera (CORDENONSI [9]).

As heurísticas de melhorias param, usualmente, quando nenhuma outra troca possível melhora o resultado atual, o que é considerado um ótimo local. Este ótimo local pode, eventualmente, ser o ótimo global, ou seja, a solução ótima para o problema, mas não há garantias em relação a este fato, como em todos os procedimentos heurísticos (CORDENONSI [9]).

## 2.2.3 Meta-heurísticas

Assim como as heurísticas construtivas e de melhorias (heurísticas clássicas), as meta-heurísticas têm como objetivo explorar apenas uma parte do espaço de solução. Entretanto, geralmente, isto é feito de um modo mais abrangente, fazendo com que as soluções encontradas sejam de melhor qualidade. Contrariamente às heurísticas clássicas, as meta-heurísticas são mais gerais e projetadas para escaparem de ótimos locais.

São muitas as meta-heurísticas aplicadas ao PRV. Algumas das principais são: *Simulated Annealing* (SA) (HWANG [22]), *Deterministic Annealing* (DA) (ROSE [23]), *Tabu Search* (TS) (GLOVER e LAGUNA [24]), *Genetic Algorithms* (GA) (GOLDBERG e HOLLAND [25]), *Ant Systems* (AS) (EYCKELHOF e SNOEK [26]), *Greedy Randomized Adaptive Search Procedures* (GRASP) (FEO e RESENDE

[27]), *Iterated Local Search* (ILS) (LOURENÇO *et al.* [28]) e *Variable Neighborhood Descent* (VND) (HERTZ e MITTAZ [29]).

Normalmente as meta-heurísticas utilizam heurísticas construtivas ou até mesmo de melhorias para compor uma solução inicial. No entanto, a maioria das meta-heurísticas atuais são robustas o suficiente para que o resultado final não dependa da solução inicial (TOTH e VIGO [3]).

Desta forma, as meta-heurísticas podem ser divididas entre baseadas em Busca Local, baseadas em populações, métodos híbridos e métodos unificados.

### 2.2.3.1 Meta-heurísticas Baseadas em Busca Local

São métodos que partem de uma solução inicial e a cada iteração geram uma nova solução vizinha. As soluções são comparadas por algum critério (normalmente verifica-se se a nova solução é melhor do que a anterior) e uma nova iteração é executada.

Como exemplos dessa categoria, tem-se as seguintes meta-heurísticas: *Deterministic Annealing* (DA), *Tabu Search* (TS), *Iterated Local Search* (ILS) e *Variable Neighborhood Search* (VNS).

### 2.2.3.2 Meta-heurísticas Baseadas em Populações

As meta-heurísticas baseadas em populações empregam, normalmente, conceitos naturais como, por exemplo, a evolução das espécies ou o comportamento social dos insetos. Estes métodos implementam comportamentos complexos em diferentes estruturas de memória tais como matrizes de feromônios, cromossomos ou redes neurais (TOTH e VIGO [3]).

Estes não são paralelos aos de busca local mas sim uma evolução, pois praticamente todos utilizam algum tipo de busca local entre as iterações de controle de população (memória).

Como exemplos dessa categoria, tem-se como os mais comuns: *Ant Colony Optimization* (ACO), *Genetic Algorithms* (GA), *Scatter Search* (SS) e *Path Relinking* (PR).

### 2.2.3.3 Meta-heurísticas Híbridas

Como citado anteriormente, métodos híbridos surgiram da combinação das meta-heurísticas até então desenvolvidas com métodos chamados de larga vizinhança (PISINGER e ROPKE [30]), inteligência coletiva, perturbações, programação inteira, busca em árvores, mineração de dados, computação paralela, entre outros (TOTH e VIGO [3]).

#### 2.2.3.4 Meta-heurísticas Unificadas

Finalmente, os métodos unificados são meta-heurísticas híbridas que surgiram com o propósito de resolverem vários PRVs ao mesmo tempo, com o objetivo de se ter uma rápida adaptação de uma variação para outra. Como podem existir uma infinidade de variações de PRVs, houve a necessidade de se criarem métodos genéricos o suficiente para que sejam aplicados em quase todo tipo de PRV. Começando pela Busca Tabu Unificada de CORDEAU *et al.* [31], posteriormente ela foi melhorada com um processo paralelo e iterativo em MAISCHBERGER e CORDEAU [32] e CORDEAU e MAISCHBERGER [33].

Dentre os métodos unificados, está o ALNS que foi proposto por ROPKE e PISINGER [34] e posteriormente foi aplicada a outros PRVs como em RIBEIRO e LAPORTE [35] no chamado PRV Cumulativo e em RIBEIRO *et al.* [15] no chamado Problema de Sondas de Intervenção com Horizonte de Planejamento que também é uma variação do PRV, entre outras implementações separadas. Atualmente, o método com os melhores resultados em uma grande variedade de PRVs é de VIDAL *et al.* [6] denominado *Hybrid Genetic Search with Adaptive Diversity Control* (HG-SADC). Posteriormente esta meta-heurística foi adaptada para uma gama maior de PRVs em VIDAL *et al.* [36], no qual passou a ser chamada *Unified Hybrid Genetic Search* (UHGS).

#### 2.2.4 Métodos Exatos

Como dito em TOTH e VIGO [3], métodos heurísticos e meta-heurísticos são conhecidos por proverem soluções de boa qualidade em curtos períodos de processamento. Mas métodos heurísticos sozinhos não podem garantir que soluções ótimas sejam encontradas. Desta maneira métodos exatos não existem só para competir com os métodos heurísticos, mas com técnicas que permitem a produção de limitantes inferiores (para problemas de minimização) ou superiores (maximização).

Assim, segundo NEMHAUSER e WOLSEY [37] a grande maioria dos métodos exatos recorrem a solução de um modelo matemático que normalmente se encontra da forma das Equações (2.1) e (2.2), no qual  $z$  é um escalar ( $z \in \mathbb{R}$ ) e consiste no valor a ser minimizado,  $x$  é um vetor coluna das variáveis de decisão cujo o valor se deseja descobrir,  $c$  é um vetor linha chamado de vetor de custos,  $b$  é um vetor coluna chamado de vetor de recursos e  $A$  é uma matriz com os coeficientes das restrições de  $x$ .

$$\text{Min } z = cx \tag{2.1}$$

$$Ax = b \tag{2.2}$$

Se todas as variáveis do vetor  $x$  assumem valores reais ( $x_i \in \mathbb{R}, \forall x_i \in X$ ), o modelo matemático das Equações 2.1 e 2.2 é facilmente resolvível em tempo polinomial por vários métodos conhecidos como Simplex, Simplex revisado, método dos pontos interiores. Entretanto, o PRV e todas as suas variantes são classificadas como Problemas de Programação Inteira (PPI), que leva ao desenvolvimento e aplicação de métodos exatos mais elaborados, como descrito nas subseções a seguir.

#### 2.2.4.1 *Branch and Bound*

Como o PRV e todos as suas variantes são classificadas como Problemas de Programação Inteira (PPI), pode-se avaliar as rotas, por meio de estruturas do tipo árvore que são ramificadas a partir de um nó raiz. Segundo NEMHAUSER e WOLSEY [37], o primeiro método exato proposto para resolver PPIs é chamado de *Branch and Bound*.

O método *Branch and Bound* é usualmente representado por meio de uma árvore binária, tal que em cada nó é resolvido o seu problema na forma real (chamado de solução da relaxação linear). Para questão de ilustração, considere o nó raiz de um PPI de minimização como sendo Nó  $A$ . Caso alguma variável  $x_i$  tenha o seu valor ótimo não inteiro, por exemplo  $x_3 = 5,15$ , são criados dois nós filhos (*Branch*),  $B_1$  e  $B_2$  tal que cada um deles possui uma restrição a mais, que limita (*Bound*) o valor da variável inteira em cada nó filho aos dois valores inteiros mais próximos, no caso do exemplo  $x_3 \leq 5$  e  $x_3 \geq 6$ . A união dos espaços de busca destes dois nós possui exatamente o mesmo espaço de busca inteiro do problema original para a variável  $x_3$ . O mesmo processo aplicado ao nó  $A$  é realizado nos seus nós filhos, e assim sucessivamente, até que o critério de parada seja atendido, detalhado a seguir.

Reforçando o processo de ramificação, a Figura 2.1 ilustra o problema descrito acima com o nó raiz  $A$ , que é ramificado nos nós  $B_1$  e  $B_2$ , limitando a variável  $x_3$  com as restrições  $x_3 \leq 5$  e  $x_3 \geq 6$  respectivamente. Em seguida tem-se uma ramificação a partir de  $B_1$  que gera  $C_1$  e  $C_2$  a qual limita a variável  $x_k \in X$  com as restrições  $x_k \leq R$  e  $x_k \geq R + 1$ , para  $R \in \mathbb{Z}$ .

Cada solução, para qualquer subproblema de  $A$ , tem o seu valor da relaxação linear chamado de limitante inferior ou *Lower Bound* (LB), pois é um valor o qual a solução ótima nunca será menor, já que essa é a solução ótima de um espaço de busca que contém todo o espaço de busca do problema com as variáveis inteiras. E consequentemente, cada solução da relaxação linear de qualquer nó filho de  $A$  que tenha todos os valores de  $x$  inteiros, é chamada de **viável**, pois respeita todas as restrições do problema original, e seu valor da relaxação inteira é chamado de limitante superior ou *Upper Bound* (UB), pois o valor de  $z$  da solução ótima do problema não pode ser maior a esse valor encontrado. Podas são realizadas quando qualquer nó possuir uma solução de relaxação linear menor que o maior LB encontrado até

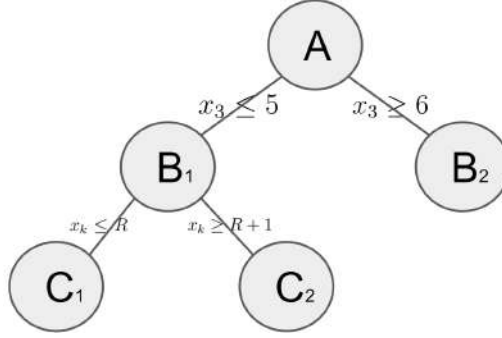


Figura 2.1: Exemplo de um problema sendo resolvido por *Branch and Bound*.

o momento. O mesmo não é necessário para qualquer UB pois quando um UB é encontrado, nenhum ramo é criado a partir do mesmo.

Desta maneira, uma vez que se tem valores limitantes inferiores, e superiores, o valor ótimo desejado tem que estar entre eles, gerando o escalar *gap* que é definido pela Equação (2.3) e consiste em percentual que quantifica a distância da melhor solução conhecida (Limitante Superior) em relação ao melhor limitante inferior.

$$gap = \frac{LB - UB}{LB} \times 100 \quad (2.3)$$

Desta maneira, vale ressaltar que qualquer solução resultante de um método heurístico, que respeite todas as restrições do problema inteiro também é uma solução viável, conseqüentemente um limitante superior (UB). E métodos heurísticos que, de alguma forma, também geram algum limitante inferior são chamados de métodos aproximativos. Logo se qualquer método heurístico encontra uma solução com valor exatamente igual ao maior limitante inferior conhecido, ele está garantindo o valor ótimo. Esta é uma estratégia comum empregada nos mais variados métodos exatos (NEMHAUSER e WOLSEY [37]).

Finalmente, o critério de parada para garantia de uma solução ótima  $z^*$  em qualquer método exato é representado na Equação (2.4) na qual uma solução ótima  $z^*$  é obtida caso  $UB - LB$  seja menor que a menor variação de  $z$  ( $\Delta z$ ) possível para quaisquer duas soluções viáveis com valores de  $z$  distintas. Como não é uma tarefa trivial encontrar o  $\min\{\Delta z | \Delta z > 0\}$  para algum PPI, o critério mais comum para parada de um método exato é representado pela Equação (2.5), que é um caso particular de (2.4) quando  $\min\{\Delta z | \Delta z > 0\} = \lim_{x \rightarrow 0^+} x$  e sabendo que não é possível  $LB > UB$ .

$$z = z^* \Leftrightarrow UB - LB < \min\{\Delta z | \Delta z > 0\} \quad (2.4)$$

$$z = z^* \Leftrightarrow UB = LB \quad (2.5)$$

Esta técnica se mostrou útil para resolver PPIs, mas como ainda é um problema NP-*Hard*, qualquer método exato aumenta muito rápido o tempo de solução em problemas mais complexos. Logo sistemas de melhorias no método são necessárias para solução exata de problemas maiores como dito em NEMHAUSER e WOLSEY [37].

#### 2.2.4.2 *Branch and Bound and Cut*

Desta forma, segundo NEMHAUSER e WOLSEY [37] a técnica de solução *Branch and Bound and Cut* foi desenvolvida sabendo que um PPI. Tem a restrição de que as variáveis são inteiras. Considere o conjunto  $P$  de pontos de coordenadas inteiras que são viáveis para o problema. A envoltória convexa é definida como o menor polígono que contém  $P$ . Para problemas NP-*Hard*, não se pode encontrar todas as desigualdades que definem a envoltória convexa em tempo polinomial a menos que  $P = NP$ , já que do contrário seria possível resolver o PPI sem necessidade de *Branch and Bound*.

Entretanto, só o fato de aproximar a formulação da envoltória convexa já é interessante para o algoritmo de *Branch and Bound*. Aproximar a formulação da envoltória, tende a aumentar o valor da relaxação linear nos nós, que por sua vez tende a aumentar o número de podas por limitantes, melhorando o desempenho do algoritmo. Para isso faz-se necessário encontrar desigualdades válidas e que aproximem o modelo da envoltória convexa. Porém, o número dessas desigualdades pode ser muito grande, tornando a resolução da relaxação linear muito lenta, sem contar que algumas das desigualdades podem ser irrelevantes para diminuir o valor do LB. A ideia então é adicionar tais desigualdades apenas por demanda, através de vários algoritmos de plano de corte como os detalhados em NEMHAUSER e WOLSEY [37].

De forma que o objetivo geral é encontrar famílias de desigualdades válidas, ao resolver a relaxação linear em cada nó do *Branch and Bound*, verifica-se se a solução ótima encontrada viola alguma dessas desigualdades. Em caso positivo, insere-se uma nova restrição no modelo e a relaxação linear é resolvida novamente, repetindo o processo enquanto o limite inferior for melhorado. Ao mudar de nó no algoritmo, descarta-se os planos de corte adicionados.

Um corte pode não ser muito eficaz, porém se a desigualdade remover apenas a solução da relaxação linear, este corte não foi muito eficiente em aproximar a formulação da envoltória convexa. Dado um plano de corte, ele pode ser classificado de acordo com a dimensão da sua interseção com a envoltória convexa. Como não existe um método unificado de cortes que aproxime muito a envoltória convexa, logo para cada tipo de problema, existem trabalhos como YAMAN [38], BALDACCI *et al.* [39] e BALDACCI *et al.* [40] que se concentram em fornecer desigualdades

válidas para algum problema específico, a fim de aproximar a envoltória convexa e mostrar a efetividade das desigualdades propostas.

### 2.2.4.3 Geração de Colunas e *Branch and Price*

Atualmente, como relatado por TOTH e VIGO [3], os métodos exatos mais eficientes na solução de PRVs e muitos PPIs envolvem geração de colunas. Tal método tem como ideia primordial, segundo DESAULNIERS *et al.* [41], que muitos PPIs são muito grandes para se considerar todas as variáveis explicitamente. Desde que a maioria das variáveis serão não básicas e terão valor nulo na solução ótima, somente um pequeno conjunto de variáveis necessitam ser consideradas para resolver o problema. Geração de colunas emprega este conceito e utiliza somente as variáveis as quais possuem potencial para melhorar o custo da função objetivo, a fim de achar as variáveis com custo reduzido negativo (assumindo sempre um problema de minimização).

*Branch and price* é um método de *Branch and Bound* em que a cada nó da árvore de busca, aplica-se um método de geração de colunas na sua respectiva relaxação de programação linear. Assim, se planos de corte são utilizados para melhorar as relaxações de programação linear com um algoritmo de *Branch and Price*, este método é então chamado de *Branch and Price and Cut* (DESROSIERS e LÜBBECKE [42]). Como relatado por FEILLET [43], a maioria dos algoritmos de *Branch and Price* são específicos para cada problema, pois cada método de solução deve ser formulado de maneira que regras específicas de ramificação são projetadas para obtenção de uma solução em tempo razoável do problema de *Pricing*. Para maiores detalhes sobre geração de colunas e *Branch and Price* recomendada-se a leitura de DESAULNIERS *et al.* [41].

## 2.3 Variantes Estudadas do PRV

Com a descrição dos principais métodos de solução exatos e heurísticos para o PRV apresentada na Seção 2.2, nesta seção, são detalhados os trabalhos mais relevantes com suas respectivas estratégias de solução para o PRV de Frota Heterogênea (Subseção 2.3.1), para o PRV Periódico (Subseção 2.3.2) e para o PRV *Multi-Trips* (Subseção 2.3.3).

### 2.3.1 PRV com Frota Heterogênea

A partir do clássico PRV, foram agregadas características que o aproximasse de casos reais: a primeira sendo uma frota composta de veículos com diferentes capacidades



e diferentes custos de retirada e de deslocamento e a segunda sendo o dimensionamento da frota. Assim, os veículos utilizados no roteamento não são mais idênticos como no problema originalmente proposto por DANTZIG e RAMSER [16], mas de diferentes tipos. Cada tipo de veículo possui uma capacidade de transporte e custos de aquisição e de manutenção próprios, além de diferentes custos variáveis (por unidade de carga transportada ou distância percorrida). Desta maneira, além de determinar a melhor rota para atender todos os pontos de demanda, deve-se determinar quais tipos e suas respectivas quantidades. Neste caso, o objetivo do problema pode ser tanto minimizar o custo total do roteamento, somando-se custos fixos de aquisição e manutenção dos veículos, e variáveis, proporcionais à distância percorrida por cada um, como também pode-se buscar o número mínimo de veículos que execute o roteamento.

O primeiro trabalho a abordar o Problema de Roteamento de Veículos com Frota Heterogênea (PRVFH) com foco em vários tipos de veículos foi de GOLDEN *et al.* [44], seguido de suas variantes ao longo dos anos.

Um grande levantamento e uma classificação dos diversos PRVFHs presentes na literatura foi feito em KOÇ *et al.* [12]. Os autores concluíram que as duas variantes mais estudadas do PRVFH são:

1. O Problema de Roteamento de Veículos Misto e Dimensionamento de Frota (PRVMDF), do inglês *Fleet Size and Mix Vehicle Routing Problem*, no qual a frota é ilimitada;
2. O Problema de Roteamento de Veículos Heterogêneos com Frota Fixa (PRVHFF), do inglês *Heterogeneous Fixed Fleet Vehicle Routing Problem*, no qual a frota é fixa.

Desta maneira, as subseções a seguir apresentam abordagens exatas e heurísticas para o PRVMDF (Subseção 2.3.1.1) e para o PRVHFF (Subseção 2.3.1.2). Algumas abordagens consideram os dois problemas simultaneamente e são detalhadas na Subseção 2.3.1.3.

### **2.3.1.1 Problema de Roteamento de Veículos Misto e Dimensionamento de Frota (PRVMDF)**

Do ponto de vista exato, vários estudos ao longo dos anos descreveram limitantes inferiores (*Lower Bounds*) e algoritmos exatos para problema. YAMAN [38] propôs algumas formulações e algumas desigualdades válidas. Das formulações apresentadas, quatro são baseadas na eliminação de restrições de sub-rotas para o TSP de MILLER *et al.* [45], e duas são baseadas em balanceamento de fluxo. O autor comparou os limitantes (*Bounds*) dessas formulações, derivou desigualdades válidas e

eliminou várias restrições para melhorar os limitantes inferiores. Os resultados revelaram que as soluções obtidas pelas formulações mais fortes eram de boa qualidade e tiveram um *gap* máximo de 3,28%.

Mais tarde, BALDACCI *et al.* [39] descreveram uma formulação de programação inteira mista baseada em fluxos de dois tipos e desenvolveram duas novas classes de desigualdades válidas. Estas desigualdades, as quais eram de cobertura e de capacidade dependente da frota, tinham como objetivo melhorar os limitantes inferiores. As desigualdades de capacidade dependente da frota foram capazes de melhorar os limitantes inferiores em 5% em média, e as de cobertura em 2,5%.

No mesmo ano PESSOA *et al.* [46] apresentaram um robusto algoritmo de *Branch Price and Cut*, tal que  $Q$ -rotas eram associadas com colunas, as quais são relaxações de rotas capacitadas elementares que fizeram o *Pricing* resolvível em tempo pseudo-polinomial. Estes autores também propuseram novas famílias de cortes as quais eram exibidas em um grande conjunto de variáveis e não aumentaram a complexidade do subproblema de *Pricing*. Dentre os resultados, instâncias de até 75 clientes tiveram as soluções ótimas encontradas, uma melhoria significativa em relação aos métodos exatos anteriores.

Já do ponto de vista heurístico, foram propostos muitos métodos baseados em Busca Tabu. A primeira foi apresentada por OSMAN e SALHI [47] que modificaram o método de perturbação de rotas de SALHI e RAND [48], melhorando os resultados existentes da época. Mais tarde, GENDREAU *et al.* [49] desenvolveram uma heurística de Busca Tabu junto da heurística de inserção generalizada de GENDREAU *et al.* [50] e do procedimento de memória adaptativa de ROCHAT e TAILLARD [51]. Seus resultados se mostraram os melhores até aquele momento. Alguns anos depois WASSAN e OSMAN [52], apresentaram uma heurística de Busca Tabu relativa, na qual várias vizinhanças e estruturas de dados especiais foram integradas contendo: uma fase de intensificação com movimentos de trocas simples, vários movimentos determinísticos introduzidos para diversificar a busca e os autores também propuseram estruturas de dados especiais para a exploração de várias vizinhanças. Seus resultados superaram os métodos anteriores na época.

Seis anos depois, uma heurística de Busca Tabu foi desenvolvida por LEE *et al.* [53] que aplicaram um procedimento de trocas com o método de partição de uma rota gigante (de PRINS [54], no qual uma distribuição ótima de rotas é gerada para uma dada sequência de clientes) que foi usada para criar soluções iniciais resultando em várias novas melhores soluções nas instâncias de GOLDEN *et al.* [44]. Finalmente, BRANDÃO [55] propôs uma Busca Tabu com três procedimentos usados para gerar soluções iniciais, e três tipos de movimentos para explorar a vizinhança: inserção simples, inserção dupla e troca. O algoritmo usava também estratégias de intensificação e diversificação durante a busca, que permitiram encontrar cinco

novas melhores soluções.

Várias versões de outras heurísticas sem Busca Tabu também foram propostas. Começando por GOLDEN *et al.* [44] que descreveram formalmente e apresentaram a primeira formulação matemática, os autores desenvolveram algumas heurísticas com base no trabalho de CLARKE e WRIGHT [17] (método das economias), e na partição de uma rota gigante nas rotas mais adequadas para vários tipos de veículos, usando o mecanismo de melhorias para o TSP de OR e PIERSKALLA [56]. Também descreveram um procedimento para cálculo de um limitante inferior e aplicaram a heurística de FISHER e JAIKUMAR [57] para resolver o problema de atribuição generalizada para a atribuição de clientes a veículos.

Em seguida, GHEYSENS *et al.* [58] usaram os procedimentos de limitantes inferiores de GOLDEN *et al.* [44] para criar uma nova heurística, gerando inicialmente uma frota mista e então resolveram como um PRV normal. Depois, GHEYSENS *et al.* [59] mostraram que a heurística de GHEYSENS *et al.* [58] apresenta resultados melhores que a de GOLDEN *et al.* [44]. Algum tempo depois DESROCHERS e VERHOOG [60] desenvolveram então o método das economias melhorado, propondo um algoritmo de economias baseado em combinações que usava sucessivos métodos para mesclar as rotas. O método seleciona a melhor solução resolvendo um problema de minimizar a soma da seleção de arestas com pesos a cada iteração, tendo bons resultados em relação aos estudos anteriores.

Um ano depois, SALHI *et al.* [61] apresentaram uma formulação matemática e forneceram uma heurística baseada em perturbações que foi testada em um conjunto de 20 instâncias teste e encontraram vários melhores resultados até aquele momento. No ano seguinte, SALHI e RAND [48] descreveram uma heurística construtiva mais avançada que resolveram um PRV com um único veículo capacitado, a partir dos tipos disponíveis para gerar a solução inicial. Vários procedimentos são então aplicados iterativamente para mudar o tipo de veículo atribuído a cada rota, para mesclar rotas ou move alguns clientes para outras rotas no processo tendo resultados, em média, melhores que as heurísticas anteriores. Vários anos depois, RENAUD e BOCTOR [62] propuseram uma heurística baseada em trocas como uma extensão do trabalho de RENAUD *et al.* [63]. O algoritmo inicialmente cria um grande número de rotas que são servidas por um ou dois veículos, particionado em conjuntos e resolvidos otimamente em tempo polinomial. O método superou os algoritmos existentes.

No mesmo ano, HAN e CHO [64] apresentaram outra heurística construtiva a qual usava métodos de intensificação e diversificação genéricos. O método incorporava vários mecanismos de variantes determinísticas como o critério de aceitação do *Simulated Annealing* e o *Great Deluge Algorithm* de DUECK [65]. O método se comportou bem nas instâncias de pequeno porte de GOLDEN *et al.* [44], mas foi do-

minado pelos métodos de TAILLARD [66] e GENDREAU *et al.* [49] nas instâncias maiores. E recentemente, VIDAL *et al.* [36] também adaptaram a meta-heurística UHGS para o problema de dimensionamento de frota, encontrando todas as melhores soluções conhecidas.

### 2.3.1.2 Problema de Roteamento de Veículos Heterogêneos e Frota Fixa (PRVHFF)

Do ponto de vista heurístico e aproximativo, inicialmente TARANTILIS *et al.* [67] apresentaram uma meta-heurística com critério de aceitação baseado em lista, que explora o espaço de soluções a fim de identificar as regiões promissoras. O método foi competitivo nas instâncias de teste e encontrou várias novas melhores soluções. Em estudos posteriores, os mesmos autores desenvolveram um algoritmo de critério de aceitação adaptativo com *backtracking* em TARANTILIS *et al.* [67], o qual torna o trabalho de TARANTILIS e KIRANOUDIS [68] um caso particular da mesma. A principal diferença deste método para o critério de aceitação padrão é que normalmente o critério de aceitação só fica mais rigoroso, enquanto nesse trabalho ele ocasionalmente relaxa, resultando em novas melhores soluções.

Dois anos depois, foi apresentada a heurística de GENCER *et al.* [69] que é baseada no princípio de primeiro agrupar para então rotear e considera a possibilidade de aumentar o número de veículos caso a atual frota seja insuficiente. O algoritmo encontrou boas soluções em termos da capacidade de utilização dos veículos, mas não forneceu soluções melhores que as conhecidas. Menos de um ano depois LI *et al.* [70] adaptaram o seu antigo algoritmo *record-to-record travel* (LI *et al.* [71]), resultando em um algoritmo que é uma variante determinística da heurística *Simulated Annealing* e produziu algumas melhores soluções conhecidas.

EUCHI e CHABCHOUB [72] apresentaram uma Busca Tabu híbrida acoplada de uma memória adaptativa heurística. Este algoritmo gera três soluções iniciais, e a cada iteração a solução atual é melhorada por vários métodos aplicados sequencialmente. Os resultados nas instâncias teste foram competitivos em termos de qualidade de solução e tempo computacional. Outro algoritmo de Busca Tabu foi proposto por BRANDÃO [73]. O método é iniciado com uma rota gigante que passa por todos os clientes, que é então particionada em rotas menores que são então melhoradas utilizando quatro tipos de movimentos. Este método encontrou quatro novas melhores soluções conhecidas nas instâncias teste.

Mais tarde, LI *et al.* [74] propuseram uma heurística denominada *multistart adaptive memory programming and path relinking*. O método constrói múltiplas soluções provisórias as quais são melhoradas com Buscas Tabu modificadas a cada iteração. Dois novos melhores resultados foram encontrados, e nas outras instâncias teste os resultados foram equivalentes com os da época. Em seguida, LIU [75] desenvolveram

uma heurística de população híbrida acoplada com mecanismos de busca local para diversificação da população que gerou resultados competitivos com baixos tempos computacionais. Finalmente, NAJI-AZIMI e SALARI [76] desenvolveram uma heurística baseada em formulação matemática. O método aplica um mecanismo no qual a solução inicial é destruída e reparada resolvendo o modelo matemático de maneira ótima. Três novas melhores soluções conhecidas foram encontradas nas instâncias de teste.

### 2.3.1.3 Métodos que resolvem o PRVMDF e o PRVHFF simultaneamente

Sob o ponto de vista exato, atualmente existem três algoritmos unificados para o PRVMDF e o PRVHFF. O primeiro foi desenvolvido por CHOI e TCHA [77] e é um algoritmo de geração de colunas que resolve a relaxação de programação linear por geração de colunas. Eles modificaram vários algoritmos de programação dinâmica para o PRV clássico afim de eficientemente gerar colunas viáveis e então aplicar o procedimento de *branch-and-bound* para obter a solução inteira. Os resultados foram os melhores do ponto de vista exato na época. Mais tarde, BALDACCI e MINGOZZI [78] introduziram um algoritmo exato unificado baseado na formulação de particionamento de conjuntos. Três tipos de procedimentos limitantes foram usados, tanto baseados em relaxação de programação linear, quanto em relaxação Lagrangeana. Os novos limitantes inferiores se mostraram superiores a todos os anteriores.

O método unificado exato atual mais recente foi apresentado por BALDACCI *et al.* [40]. Ele combina vários procedimentos de ascendentes duais para gerar uma solução dual quase-ótima do modelo de partição de conjuntos. Adiciona algumas desigualdades válidas a formulação de partições de conjuntos também um algoritmo de *column-and-cut* para fechar o *gap* deixado pelos procedimentos de ascendentes duais. A solução final dual é então usada para gerar o problema reduzido que contém todas as soluções inteiras ótimas. Este então é atualmente o melhor método exato para o PRVFH.

Já do ponto de vista heurístico, o PRVMDF e o PRVHFF foram tratados simultaneamente em alguns trabalhos. O primeiro trabalho foi de TAILLARD [66] que desenvolveu uma heurística baseada em geração de colunas. Seu método resolveu um PRV clássico (homogêneo) com o método de memória adaptativa de ROCHAT e TAILLARD [51] para cada tipo de veículo, com um número de veículos ilimitado, que foi superado pela de OSMAN e SALHI [47] nas instâncias de teste maiores de dimensionamento de frota. Vários anos depois, IMRAN *et al.* [79] adaptaram um *Variable Neighborhood Search Algorithm (VNS)* para ambos os problemas com algumas adições como: uma adaptação dos algoritmos de busca local incluindo o

algoritmo de Dijkstra [80], um método de diversidade, e o uso de uma rota artificial durante a execução. Resultados competitivos também foram encontrados nas instâncias de teste e várias novas melhores soluções conhecidas foram encontradas.

Um algoritmo híbrido que considera ambos os problemas foi mais tarde proposto por SUBRAMANIAN *et al.* [81]. O algoritmo inclui uma heurística baseada em *Iterated Local Search (ILS)* para gerar colunas na formulação de partição de conjuntos. Resultados competitivos e novas melhores soluções conhecidas foram encontradas incluindo grandes instâncias envolvendo até 360 clientes. Os mesmos autores apresentaram melhores resultados integrando o *ILS* com um método baseado em *Variable Neighborhood Descent (VND)* com um esquema de ordenação de vizinhanças aleatórias na fase de busca local que foi proposto em PENNA *et al.* [82]. O desempenho desse método foi testado em todas as 52 instâncias de teste mais conhecidas, que possuem até 100 clientes e quatro novas melhores soluções conhecidas foram encontradas. Até o momento desta revisão bibliográfica, o método de PENNA *et al.* [82] detém os melhores resultados encontrados além dos melhores tempos de processamento.

### 2.3.2 PRV Periódico

O Problema de Roteamento de Veículos Periódico (PRVP) é uma extensão do PRV no qual as rotas dos veículos são distribuídas ao longo de vários dias (FRANCIS *et al.* [1]). O PRVP é considerado como um PRV de múltiplos níveis e consiste em selecionar simultaneamente uma combinação de visitas para cada cliente e estabelecer as rotas dos veículos para cada dia do horizonte de planejamento de acordo com as regras do PRV em questão (CORDEAU *et al.* [5]).

O PRVP pressupõe um horizonte de planejamento tal que cada cliente especifica uma certa frequência de atendimentos e um conjunto de combinações possíveis de visitas. Por exemplo, para o horizonte de planejamento de 5 dias, se um cliente deseja dois atendimentos sem restrição de dias, então ele deve ser visitado duas vezes e as visitas devem ocorrer nos dias 1 e 2, ou nos dias 1 e 3 ou nos dias 1 e 4, e assim sucessivamente (ARENALES *et al.* [83]). Durante cada dia, uma frota de veículos com capacidade finita viaja ao longo de rotas que começam e finalizam no centro de distribuição. Normalmente o objetivo de um PRVP é minimizar a distância ou o custo total associado a todas as rotas executadas (FRANCIS *et al.* [1]).

O PVRP foi mencionado pela primeira vez em BELTRAMI e BODIN [84], seguido de FOSTER e RYAN [85] e mais tarde formalizado por RUSSELL e IGO [86] e CHRISTOFIDES e BEASLEY [87]. Os primeiros métodos de solução para o PRVP foram heurísticas simples de construção e buscas locais propostas por RUS-

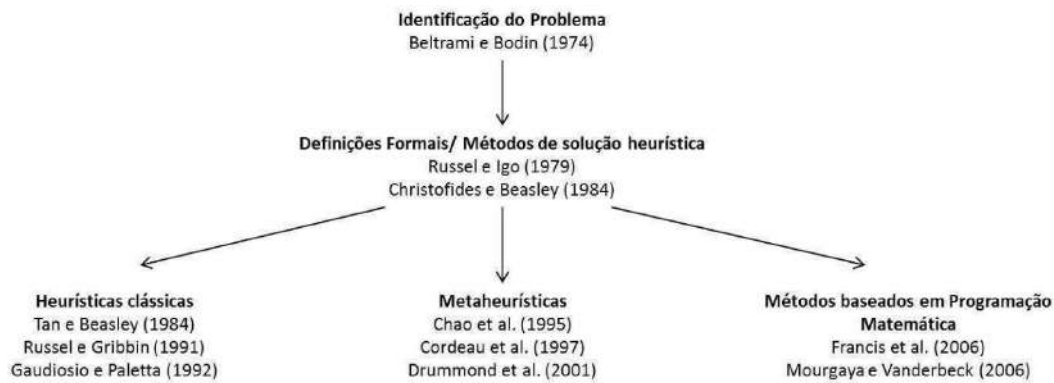


Figura 2.2: Evolução dos modelos e métodos para solução do PRVP (FRANCIS *et al.* [1]).

SELL e GRIBBIN [88], TAN e BEASLEY [89], e GAUDIOSO e PALETTA [90], as quais, mais tarde foram aprimoradas por técnicas meta-heurísticas como em CHAO *et al.* [91], CORDEAU *et al.* [5], e DRUMMOND *et al.* [92]. Em seguida, métodos de solução baseados em programação matemática foram propostos por FRANCIS e SMILOWITZ [93], FRANCIS *et al.* [94], e MOURGAYA e VANDERBECK [95].

Quanto a técnicas de modelagem do PRVP, FRANCIS e SMILOWITZ [93] discutiram detalhadamente as possíveis técnicas de modelagem para o PRVP. E dentre as possíveis aplicações práticas do PRVP abordadas na literatura, destaca-se a agenda da coleta de lixo e recicláveis, temas abordados por ANGELELLI e SPERANZA [96], NUORTIO *et al.* [97], e COENE *et al.* [98], a distribuição e coleta de produtos que foi abordado por ALEGRE *et al.* [99], CLAASSEN e HENDRIKS [100] e CHRISTIANSEN *et al.* [101], a programação de manutenções para fábricas em produção que foi abordado por HADJICONSTANTINOU e BALDACCI [102] e BLAKELEY *et al.* [103], também foi citado na área de saúde como em HEMMELMAYR *et al.* [104], PACHECO *et al.* [105], SHAO *et al.* [106], e MAYA *et al.* [107].

De acordo com o levantamento sobre PRVP feito por FRANCIS *et al.* [1], a Figura 2.2 apresenta uma breve revisão do progresso do PRVP ao longo dos anos até o ano de 2008. Tudo começou com BELTRAMI e BODIN [84], pioneiros nessa área, com uma aplicação na coleta de lixo municipal. Em seguida, outros autores deram continuidade aos estudos e definiram formalmente o PRVP. A partir de então, os estudos foram expandidos abrangendo métodos de solução desde heurísticas clássicas e meta-heurísticas até estudos mais recentes em programação matemática.

A partir do ano de 2008, o levantamento bibliográfico realizado, indicou que houve um aumento na incidência de abordagens heurísticas e exatas ao PRVP. Desta maneira, as subseções a seguir (2.3.2.1 e 2.3.2.2) detalham os métodos exatos, híbridos, heurísticos e meta-heurísticos mais recentes, pois os mesmos apresentaram um

grande progresso.

### 2.3.2.1 Abordagens Exatas e Híbridas

De acordo com a literatura, BALDACCI *et al.* [108] foram os primeiros e únicos até o momento a proporem técnicas de corte e um método exato para o PRVP. Foram usadas diferentes relaxações e uma formulação de partição de conjuntos estendida para se criar cinco métodos para cálculo de limitantes, que são usados para a geração de um problema reduzido que pode ser resolvido por um *solver* de programação linear inteira (e.g. CPLEX ou Gurobi). Testes em instâncias da literatura de até 153 clientes e em conjunto de 20 instâncias propostas pelos autores por serem “inspiradas em casos práticos” de até 199 clientes, mostraram a efetividade do método. Os limitantes inferiores se mostraram eficientemente estreitos, pois geraram um gap ótimo com 1% de média em todas as instâncias. Várias instâncias teste convergiram para o ótimo global e novas melhores soluções conhecidas foram encontradas.

Em GULCZYNSKI *et al.* [109] foi apresentado um método híbrido e duas variantes do problema. A primeira variante atribui clientes para novas rotas enquanto restringe mudanças nas rotas mais bem estabelecidas. A segunda variante tenta distribuir igualmente a carga de trabalho por todas as rotas. O método híbrido combina: (i) uma busca em grande vizinhança que é baseada na solução de um problema de programação linear inteira para a atribuição de múltiplos clientes em novas rotas e (ii) um algoritmo com *record-to-record travel* para melhorar as rotas a cada dia proposto por GROËR *et al.* [110]. Os resultados nas instâncias teste mostraram desempenho competitivo com os melhores métodos até então propostos.

CACCHIANI *et al.* [111] apresentaram outro método baseado na fixação e liberação de variáveis da relaxação linear de uma formulação de cobertura de conjuntos do PRVP. Colunas são geradas de maneira heurística usando um algoritmo de busca local iterativa, *Iterated Local Search (ILS)*. Seus resultados no conjunto de instâncias mais antigo, tem um desempenho próximo do HGSADC de VIDAL *et al.* [6], com um desvio médio menor, no entanto tempo computacional maior. Nas instâncias propostas por PACHECO *et al.* [105], a meta-heurística obteve soluções de melhor qualidade que os métodos anteriores, mas também com maior custo computacional que todos os outros.

### 2.3.2.2 Métodos Heurísticos e Meta-heurísticos

Sob o ponto de vista de abordagem heurística, toda a literatura específica do PRVP é bem recente, começando por HEMMELMAYR *et al.* [112] que propuseram um *Variable Neighborhood Search (VNS)* com um critério de aceitação baseado no SA (*Simulated Annealing*). A fase de perturbação é baseada em alterações aleatórias



de clientes dentro da própria rota, entre rotas e até entre dias. A fase de busca local aplica os operadores  $\beta$ -opt de LIN [113]. Foram conduzidos testes em todas as instâncias que mostraram resultados competitivos na qualidade das soluções e bons tempos de processamento. E nos testes extensivos foram encontradas novas melhores soluções para quase todas as instâncias. O método mostrou um bom desempenho em termos de tempo de processamento nas instâncias maiores em relação aos métodos anteriores.

Em PIRKWIESER e RAIDL [114] foi apresentado outro método baseado em VNS que usa uma estratégia de refino em múltiplos níveis a fim de melhorar a escalabilidade para as instâncias de grande porte. Os resultados mostraram resultados equivalentes com os de HEMMELMAYR *et al.* [112] nas instâncias comuns com o mesmo número de iterações. No entanto, os autores também testaram o método em instâncias maiores, de até 576 clientes, geradas utilizando o procedimento de CORDEAU *et al.* [5] para mostrar a escalabilidade de tempo proposta inicialmente.

Em VIDAL *et al.* [6] foi proposta a chamada Busca Genética Híbrida com Controle Adaptativo de Diversidade, do inglês *Hybrid Genetic Search with Adaptive Diversity Control (HGSADC)*, para resolver o PRVP, o PRV de múltiplos depósitos, e sua variante combinada. Nas instâncias de teste, HGSADC obteve melhores resultados médios em 41 das 42 instâncias. E em testes extensivos, encontrou 20 novas melhores soluções.

Finalmente, em CORDEAU e MAISCHBERGER [33] foi proposta uma versão paralela e iterativa da Busca Tabu de CORDEAU *et al.* [31] para resolver o PRV e outras variantes, na qual a Busca Tabu com *ILS* também explora as possibilidades da computação paralela dos computadores atuais com diversos núcleos. Seus resultados médios são os melhores até o presente momento no PRVP, mas não encontrou alguns dos melhores resultados de VIDAL *et al.* [6].

### 2.3.3 PRV *Multi-Trips*

O Problema de Roteamento de Veículos *Multi-Trips* (múltiplas viagens) (PRVMT), caracteriza-se por levar em consideração que os veículos são guiados por motoristas que possuem jornadas de trabalho estabelecidas por lei e, sendo assim, podem realizar mais de uma viagem por dia. Na prática, múltiplas programações de viagens são importantes, pois reduzem os custos uma vez que reduz o número de veículos utilizados e melhor emprega o tempo dos motoristas.

O PRVMT pode ser usado tanto para o planejamento estratégico quanto para o tático. Estratégico no sentido de minimizar o tamanho da frota, reduzindo os custos de utilização de cada veículo. Dentro de um sistema logístico de distribuição, se a frota é fixada à configuração existente, e assim retirando os custos fixos de uso do

veículo, o objetivo se torna minimizar somente os custos de deslocamento (PETCH e SALHI [115]).

Embora na prática, o uso de múltiplas rotas seja comum, há uma escassez de publicações cobrindo este tema, além da falta de um conjunto de instâncias unificado tal que todos os métodos propostos o use. O trabalho mais antigo encontrado que aborda explicitamente *multi-trips* é de SALHI [116] no contexto de composição de frota de veículos. Limitado a duas viagens por veículo, um algoritmo é usado para alocar rotas para veículos com um processo de refinamento. Esse problema também foi abordado por FLEISCHMANN [117] que gerou uma solução utilizando um algoritmo de uma fase ao integrar uma heurística gulosa com a atribuição de rotas para veículos. Usando uma medida de economia em relação a pares de clientes, a viabilidade de rotas parcialmente construídas é avaliada e a atribuição das rotas usa o método *bin-packing heuristic best fit decreasing* de MARTELLO e TOTH [118].

Mais tarde, uma abordagem de duas fases foi proposta por TAILLARD *et al.* [119] na qual um conjunto de soluções de PRV é construído a partir de uma população de rotas geradas usando a Busca Tabu de ROCHAT e TAILLARD [51] sendo que, antes uma heurística *bin-packing* era usada para alocar rotas a veículos.

Em seguida, uma heurística construtiva e de melhoria contínua foi proposta por BRANDAO e MERCER [120]. Os autores abordaram várias viagens como parte de um problema mais amplo envolvendo janelas de tempo e composição da frota de veículos. Usando dados reais, os resultados mostraram que a heurística produziu uma economia de 20% quando comparado com a solução manual. Para comparar com os resultados de TAILLARD *et al.* [119], BRANDAO e MERCER [13] alteraram a sua heurística para resolver o PRVMT clássico. A sua abordagem da fase de construção é baseada na regra do vizinho mais próximo, e também como critério de inserção para atribuir clientes às rotas dos veículos. Este processo é repetido até que todos os clientes são inseridos em alguma rota. A fase de melhoria contínua tenta primeiro balancear os tempos removendo algumas entregas para então realocá-las usando uma heurística de custos mínimos aliada a uma Busca Tabu.

Mais tarde, PETCH e SALHI [115] abordaram o problema com uma heurística construtiva de múltiplas fases, que primeiro constrói várias soluções viáveis e, para cada uma, atribui rotas a veículos usando a heurística *bin-packing*. Soluções viáveis para o PRVMT são construídas com repetidas execuções de outra heurística de economias independentemente, gerando ao final uma população de soluções paralelas. Em seguida OLIVERA e VIERA [121], propuseram uma heurística de memória adaptativa e a testaram considerando os dados dos trabalhos de PETCH e SALHI [115]. Os resultados obtidos se mostraram superiores aos até então conhecidos.

Já do ponto de vista exato, um método exato foi proposto por MINGOZZI *et al.* [122] baseado em *Branch Prince and Cut* e, recentemente, TANG *et al.* [123] também

resolveram, com um método exato, uma variante do PRVMT com coleta e entregas em um aeroporto, por ser o caso real em estudo.

Também, com o recente tema que envolve veículos com energia alternativa, como carros elétricos, que possuem curtos períodos de autonomia quando comparados aos carros à gasolina ou álcool, a comunidade científica começou a estudar variantes do PRV para tratar destas questões. A principal se deve a carga máxima suportada pelas baterias atuais, que força a visita a um ponto de recarga local durante o percurso. Este problema é uma variante do PRVMT e foi estudado por ERDOĞAN e MILLER-HOOKS [124] e por SCHNEIDER *et al.* [125], neste caso com janelas de tempo.

## 2.4 Considerações Finais

De maneira geral, este capítulo apresentou conceitos sobre o PRV e seus principais métodos de solução, para então apresentar os métodos propostos para as variantes do PRV tratadas nesta dissertação. Com isso, o Capítulo 3 detalha o método de solução proposto neste trabalho, ou seja, um ALNS com paralelismo, busca com região de viabilidade oscilante e estratégias de detecção de estagnação e perturbações.

## Capítulo 3

# *Adaptive Large Neighborhood Search (ALNS)*

Neste capítulo, é apresentada a meta-heurística ALNS para o PRV-FH-P-MT com todas as suas características básicas, estratégias de busca e paralelismo propostos nesta dissertação. Apresentado primeiramente por ROPKE e PISINGER [34], o ALNS estende a meta-heurística *Large Neighborhood Search* (LNS) proposta por SHAW [126] que é baseada no princípio de destruição e reconstrução de rotas.

A cada iteração, o ALNS *destrói* uma dada solução  $s$  e a *repara* de uma maneira diferente, gerando uma nova solução  $s'$ . Esta nova solução é aceita de acordo com o critério de aceitação do *Simulated Annealing* (SA) no qual se  $s'$  é melhor do que  $s$ , a busca continua a partir de  $s'$ , caso contrário a busca continua a partir de  $s$  dada uma probabilidade.

O que diferencia o ALNS do LNS é que no LNS os métodos de destruição (remoção) e reparo (inserção) são selecionados com probabilidades iguais, enquanto no ALNS a seleção ocorre de acordo com um mecanismo adaptativo probabilístico. A cada iteração, a probabilidade de seleção de um método depende do seu desempenho no passado.

O ALNS teve a sua eficiência testada em vários problemas de roteamento desde que foi proposto. Inicialmente foi utilizado para resolver o problema de roteamento de veículos capacitado com *backhauls* em que rotas são definidas para os veículos de entrega que no retorno ao depósito, passam a realizar coleta de mercadorias (ROPKE e PISINGER [34]). Os autores testaram o algoritmo em 338 instâncias diferentes do problema, e o ALNS conseguiu achar soluções melhores que as até então conhecidas em 227 delas.

Ainda no mesmo ano, em ROPKE e PISINGER [11] apresentaram outra versão do ALNS com mais heurísticas de inserção e remoção que foi testada em PRVs de coleta e entrega com janelas de tempo. Em mais de 350 instâncias com até 500 clientes, o ALNS foi capaz de encontrar melhores soluções em mais da metade.

Em seguida, PISINGER e ROPKE [127] aplicaram o ALNS a outros PRVs: com janelas de tempo, capacitado, com múltiplos depósitos, no problema de roteamento de veículos dependente do local, e no problema aberto de roteamento de veículos. Os autores transformaram todos estes problemas em um rico PRV e usaram o ALNS. De 486 instâncias testadas, o ALNS foi capaz de encontrar novas melhores soluções em 183 delas.

O ALNS foi aplicado também em outras variantes do PRV por outros autores. DEMIR *et al.* [128] utilizaram um ALNS para resolver o Problema de Roteamento com Poluição que é uma extensão do PRV com janelas de tempo e o objetivo também minimiza a emissão de poluentes, conseqüentemente o consumo de combustível ao mesmo tempo que tenta reduzir a distância total percorrida. Desta maneira foram implementadas heurísticas de inserção e remoção que beneficiam um objetivo de cada vez. Os resultados mostraram que o algoritmo proposto foi capaz de encontrar bons resultados em instâncias de até 200 clientes.

Já ADULYASAK *et al.* [129] implementaram um ALNS para o Problema de Roteamento de Produção, que é um problema que busca soluções otimizando produção, estoque, distribuição e rotas de maneira a satisfazer uma demanda dinâmica de clientes, reduzindo o custo de todo o sistema. Os resultados se mostraram superiores aos métodos anteriores na grande maioria das instâncias de teste, e os autores ainda ressaltaram que o método poderia ser facilmente adaptado para a variante com vários produtos.

MASSON *et al.* [130] implementaram um ALNS com base no de ROPKE e PISINGER [11] para o PRV com Coletas e Entregas com Transferências que é uma variante do PRV com coletas e entregas no qual uma requisição pode mudar de veículo durante a rota e as transferências são realizados em pontos na rede denominados “Pontos de transferência”. Dentre os resultados, foram encontradas melhorias de até 9% em relação aos melhores métodos até então propostos para esse problema.

RIBEIRO *et al.* [15] utilizaram o ALNS para resolver o Problema da Alocação de Sondas de Produção a Poços de Petróleo (PASPPP), no qual o mesmo foi comparado a um algoritmo exato de RIBEIRO *et al.* [131]. Verificou-se que em 80 instâncias testadas do PASPPP, o algoritmo exato de RIBEIRO *et al.* [131] foi capaz de encontrar 67 soluções ótimas. O ALNS encontrou 49 delas com 25000 iterações, e 53 com 50000 iterações. Em todos os testes, o erro máximo encontrado pelo ALNS foi de 0,791% em relação às soluções ótimas conhecidas. Além disso, o ALNS achou novas melhores soluções para as 13 instâncias sem valor ótimo conhecido.

Estes bons resultados motivaram a desenvolver um ALNS para o PRV-FH-P-MT. Sendo assim, seus mecanismos e características são descritos na Seção 3.1.

### 3.1 Mecanismos e Funcionamento Geral do ALNS para o PRV-FH-P-MT

Nesta seção são descritos de maneira geral os principais elementos do ALNS para o PRV-FH-P-MT. Particularidades da implementação de cada componente serão apresentados nas seções posteriores. A meta-heurística contém vários parâmetros denotados por letras Gregas, os quais serão calibrados na Seção 4.1.

**Grande Vizinhaça (*Large Neighborhood*)** Dado uma Solução  $s$ , a cada iteração,  $\gamma$  elementos são removidos da solução e então re-inseridos. Isto é realizado com um par de heurísticas de inserção e remoção.

**Função Objetivo Penalizada** Permite explorar o espaço de busca com rotas que violam tanto as restrições de capacidade de carga, quanto as restrições de tempo máximo de uma rota por meio de pesos. Dado uma solução  $s$ ,  $f(s)$  consiste então na função objetivo que representa o custo das rotas acrescido das violações ponderadas pelos pesos.

Entretanto, como nas heurísticas de remoção e de inserção, esses pesos são atualizados a cada segmento de  $\psi$  iterações consecutivas. Caso a solução  $s$  seja inviável, os pesos são multiplicados por  $\alpha^+ > 1$ , caso contrário, eles são multiplicados por  $0 < \alpha^- < 1$ . Esse ajuste dinâmico ocorre para que a busca em grande vizinhaça oscile entre regiões viáveis e inviáveis.

**Critério de Aceitação (Temperatura)** O critério de aceitação utilizado é o do SA, tal que dada uma solução  $s$ , uma solução modificada  $s'$  é aceita se  $f(s') < f(s)$  ou com probabilidade  $e^{\frac{f(s)-f(s')}{T}}$ , sendo que  $T$  é chamado de temperatura. A temperatura  $T$  começa com o valor inicial ( $T_0 > 0$ ) e decresce a cada iteração sendo multiplicada pela taxa de resfriamento ( $0 < \alpha_T < 1$ ), terminando com um valor  $0 < T_F \ll 1$ .

**Mecanismo de Busca Adaptativa** A escolha das heurísticas de inserção e remoção é regida por uma roleta ponderada na qual, a cada heurística é atribuída um peso que depende de seu comportamento nas iterações anteriores. Especificamente, seja  $w_i$  o peso de seleção da heurística  $i$ . Então, dado  $h$  heurísticas com pesos  $w_i$ , a heurística  $j$  tem a seguinte probabilidade de ser escolhida  $w_j / \sum_{i=1}^h w_i$ . Este processo é aplicado para as heurísticas de remoção e inserção de maneira independente.

**Ajuste de Pesos Adaptativos** A busca é dividida em *segmentos* de  $\psi$  iterações consecutivas. No primeiro segmento, todas as heurísticas possuem o mesmo peso  $w_0$ , ou seja  $w_i = w_0$  para  $i = 1, \dots, h$ . Após  $\psi$  iterações, os pesos usados para selecionar

as heurísticas de inserção e de remoção são atualizados de acordo com a *pontuação* obtida durante o segmento, nas vezes em que foram selecionadas. Cada pontuação mostra o quão bem cada heurística se comportou durante o último segmento.

**Paralelismo** Permite melhor aproveitar os recursos computacionais disponíveis. Sendo assim,  $\eta$  execuções simultâneas do ALNS são realizadas e a cada final de segmento todas as melhores soluções viáveis são analisadas.

**Detecção de Estagnação e Perturbações** É muito comum as meta-heurísticas estagnarem em ótimos locais. Isto acontece quando as buscas ocorrem em regiões do espaço de busca não promissoras ou já totalmente exploradas. Quando verifica-se a estagnação, uma perturbação é aplicada em  $\theta$  das  $\eta$  execuções do ALNS. Este processo ajuda a diversificar e intensificar o processo de busca.

## 3.2 Descrição do ALNS para o PRV-FP-P-MT

Nesta seção, são especificados os detalhes da implementação, tal como o cálculo da função objetivo penalizada levando em conta as particularidades de cada variante (PRVFH, PRVP e PRVMT), bem como suas combinações em pares (Subseção 3.2.1). Na Subseção 3.2.2 é detalhado o algoritmo de geração de uma solução inicial, seguido da Subseção 3.2.3 na qual são detalhadas as heurísticas de remoção e inserção. Os detalhes de execução da Camada Adaptativa e da Temperatura estão nas Subseções 3.2.4 e 3.2.5. Então são especificadas a utilização de paralelismo, bem como a estratégia de detecção de estagnação, seguido de perturbações na Subseção 3.2.6. Finalmente, o código principal do ALNS é detalhado na Subseção 3.2.7.

### 3.2.1 Função Objetivo e Representação das Variações do PRV estudadas

Um dos temas que é pouco abordado em implementações do ALNS consiste em como explorar regiões de busca inviáveis. A grande maioria das implementações como em RIBEIRO e LAPORTE [35], ROPKE e PISINGER [34], RIBEIRO *et al.* [15] entre outras, somente exploram soluções inviáveis com uma constante de penalidade grande o suficiente para que as soluções iniciais possam ser inviáveis e uma vez que o algoritmo passa a encontrar soluções viáveis (se possível), não se volta a explorar as inviáveis pelo fato do peso das penalidades serem grandes. A literatura indica que permitindo a exploração controlada de soluções inviáveis pode-se melhorar a performance da busca, na qual se traduz em uma transição mais natural entre soluções viáveis e inviáveis como dito por GLOVER e HAO [132]. Com base nesta

linha de pensamento, foi definido um espaço de busca no qual as regiões inviáveis implicam em penalidades dinâmicas para a Função Objetivo Penalizada  $f(s)$  aplicada ao PRV-FH-P-MT. Regiões inviáveis são acessadas por meio da relaxação do limite de carga de cada veículo, bem como do tempo total para o atendimento como sugerido em GENDREAU *et al.* [133] e CORDEAU *et al.* [5].

Verificou-se também a possibilidade de aplicar penalidades quando clientes são atendidos fora dos períodos estipulados, no caso do PRV periódico, entretanto os resultados não foram expressivos. Sendo assim, optou-se por trabalhar com soluções que respeitem as restrições dos períodos de atendimento definidos pelos clientes.

Descrevendo formalmente o mecanismo de penalidades implementado para o PRV-FH-P-MT, uma rota qualquer  $r$  é uma tripla sendo  $r = (i, j, E)$  tal que  $i$  é o dia de  $r$ ,  $j$  é o veículo de  $r$  e  $E$  é a sequência de clientes *multi-trip* que o veículo  $j$  atende no dia  $i$ . Por exemplo, caso  $E = \{0, 2, 4, 8, 0, 3, 5, 0\}$  significa que o o veículo deve deixar o depósito (nó 0), atender os clientes 2, 4 e 8, retornar para o depósito novamente, sair novamente para atender os clientes 3 e 5, e retornar para o depósito.

Desta maneira, uma solução  $s$  possui  $n$  rotas, ou seja  $s = \{r_1, r_2, \dots, r_n\}$ . E como a avaliação da função objetivo é independente para cada rota, pode-se escrever que o custo das rotas ( $f'(s)$ ) é dado por  $f'(s) = c(r_1) + c(r_2) + \dots + c(r_n)$ , sendo  $c(r)$  o custo da rota  $r$ . Por sua vez  $c(r) = c_0^j + c_d^j D(r)$ , tal que  $c_0^j$  é o custo fixo do veículo  $j$ ,  $c_d^j$  é o custo por unidade de distância do veículo  $j$  e  $D(r)$  a distância total da rota  $r$ . Também pode-se representar por  $q(r)$  a carga total coletada na rota  $r$  e por  $t(r)$  o tempo total de percurso da rota  $r$ . Considere ainda  $Q^j$  a carga máxima do veículo  $j$  e  $T$  o tempo máximo de percurso.

Agora considere  $\omega_q$  o peso da penalidade aplicada quando a capacidade é violada e  $\omega_t$  o peso da penalidade quando o tempo máximo de rota é violado. Pode-se escrever o custo da rota com penalidades como  $\psi(r)$  conforme a Equação (3.1).

$$\psi(r) = c(r) + \omega_t \max\{t(r) - T, 0\} + \omega_q \max\{q(r) - Q^j, 0\} \quad (3.1)$$

$$\sum_{r \in s} \omega_t \max\{t(r) - T, 0\} + \sum_{r \in s} \omega_q \max\{q(r) - Q^j, 0\} = 0 \quad (3.2)$$

Logo o custo total com penalidades da solução  $s$  com  $n$  rotas agora é escrito como  $f(s) = \psi(r_1) + \psi(r_2) + \dots + \psi(r_n)$ . Assim uma solução  $s$  é chamada de viável quando a Equação (3.2) é satisfeita, caso contrário, a solução  $s$  é chamada de inviável.

Os pesos das penalidades possuem valores iniciais  $\omega_0^q$  para penalidade de sobrecarga e  $\omega_0^t$  para penalidade de sobretempo. Para manter as soluções no limiar da viabilidade, a cada segmento de  $\psi$  iterações, quando uma solução é avaliada como inviável em carga ou tempo, o peso correspondente  $\omega_x$  ( $x$  sendo  $q$  ou  $t$ ) é atualizado com  $\omega_x \leftarrow \omega_x \alpha^+$ , caso contrário, o peso correspondente é atualizado com  $\omega_x \leftarrow \omega_x \alpha^-$ .



Os valores desses parâmetros  $\omega_0^q$ ,  $\omega_0^t$ ,  $\alpha^-$  e  $\alpha^+$ , são discutidos e apresentados na Seção 4.1.

### 3.2.2 Solução Inicial

Como o PRV-FH-P-MT também é um problema periódico, cada cliente  $i$  possui uma frequência  $n$ , e conseqüentemente  $n$  requisições a serem atendidas. Logo o ALNS considera que requisições serão removidas e inseridas a cada iteração. Sendo assim, optou-se por soluções iniciais aleatórias. Dada uma solução inicial  $s$  tal que nenhuma requisição é atendida, para cada requisição  $i$  é escolhido um dia aleatório  $d$  e um veículo aleatório  $v$ . Então, se possível  $i$  é inserido no ponto que menor incremento  $f(s)$  no dia  $d$  e veículo  $v$ . Caso não seja possível, é sorteado um novo dia e veículo para a requisição  $i$  até que todas as requisições sejam atendidas em  $s$ , formando assim a solução inicial.

Como explicado na Seção 3.1, cada execução paralela do ALNS utiliza inicialmente o mesmo procedimento indicado acima que gera  $\eta$  soluções iniciais diferentes.

### 3.2.3 Heurísticas de Remoção e Inseção

Como apresentado na Seção 3.1, a cada iteração do ALNS,  $\gamma$  requisições da solução atual são removidas por uma heurística de remoção, denominada  $H^-$ , que é selecionada pelo Mecanismo de Busca Adaptativo e re-inseridos por uma heurística de inserção, denominada  $H^+$ , que também é selecionada pelo Mecanismo de Busca Adaptativo. Com isso, quatro heurísticas de remoção foram utilizadas e três heurísticas de inserção. Essas heurísticas são apresentadas nas Subseções na sequência.

#### 3.2.3.1 Heurística de Remoção Aleatória

Esta heurística de remoção remove  $\gamma$  requisições selecionadas aleatoriamente da solução atual  $s$ . Ela tende a gerar um pior conjunto de requisições removidas, porém ajuda a diversificar a busca.

#### 3.2.3.2 Heurística de Remoção Shaw baseada na Função Objetivo

Proposta por SHAW [134], a ideia geral desta heurística é a remoção de requisições semelhantes, pois requisições semelhantes podem ser facilmente remanejadas, possibilitando soluções melhores. O grau de similaridade entre duas requisições  $i$  e  $j$  é determinado pelo operador  $R(i, j)$ , sendo que menores valores representam requisições mais semelhantes. Nesta implementação, o grau de similaridade entre duas requisições  $i$  e  $j$  é a diferença absoluta da contribuição de cada requisição para a função objetivo. Como a característica padrão da Heurística de Remoção de Shaw

exige que uma das requisições não esteja na solução  $s$ , a contribuição para a função objetivo da requisição  $i$  que não é atendida em  $s$  é a variação de  $f(s)$  no momento da sua última remoção. Já a contribuição para a função objetivo da requisição  $j$  que é atendida em  $s$  é a variação de  $f(s)$  caso  $j$  seja removida de  $s$ . Tal critério também foi utilizado nas implementações de RIBEIRO e LAPORTE [35] e RIBEIRO *et al.* [135].

O Algoritmo 3.1 detalha a implementação da Heurística de Remoção Shaw iniciando nos Passos 4 e 5 que remove aleatoriamente uma requisição  $r$  de  $s$ . Do Passo 6 até o 11 ocorre o laço principal de remoções da heurística. Iniciando o laço no Passo 7, uma requisição aleatória de  $D$  é selecionada e, no Passo 8 é ordenada a lista de requisições  $L$  com base nos valores de  $R(r, L_i)$  em ordem crescente para cada requisição  $L_i$  em  $L$ . Então é sorteado um número aleatório  $y$  entre 0 e 1 no Passo 9, que seleciona a requisição a ser retirada da solução com o índice  $y^\phi |L|$ , tal que  $\phi > 1$ . A requisição é então removida de  $s$ . Este processo é repetido até que  $\gamma$  requisições sejam removidas.

---

**Algorithm 3.1** Heurística de Remoção Shaw baseada na Função Objetivo

---

- 1: **Entrada:** Solução  $s$ , e os parâmetros  $\gamma, \phi > 1$
  - 2: Considere  $D$  o conjunto de requisições removidas de  $s$
  - 3: Considere  $L$  a lista das requisições atendidas de  $s$
  - 4:  $r \leftarrow$  uma requisição aleatória de  $s$
  - 5:  $s \leftarrow s - \{r\}, D \leftarrow D + \{r\}, L \leftarrow L - \{r\}$
  - 6: **while**  $|D| < \gamma$  **do**
  - 7:  $r \leftarrow$  uma requisição aleatória de  $D$
  - 8: Ordene  $L \mid i < j \Rightarrow |R(r, L_i)| < |R(r, L_j)|$
  - 9: Escolha um número aleatório  $y \in [0, 1]$
  - 10:  $i \leftarrow y^\phi |L|, s \leftarrow s - \{L_i\}, D \leftarrow D + \{L_i\}, L \leftarrow L - \{L_i\}$
  - 11: **end while**
  - 12: **return**  $s$
- 

### 3.2.3.3 Heurística de Remoção Shaw baseada em Distâncias

Se diferencia da Heurística de Remoção Shaw baseada na Função Objetivo, pois o operador  $R(i, j)$  que calcula o grau de similaridade entre as requisições  $i$  e  $j$  é a distância de percurso do cliente da requisição  $i$  até o cliente da requisição  $j$ .

### 3.2.3.4 Heurística de Remoção do Pior

Esta heurística tem como ideia principal remover requisições de alto custo na solução  $s$  atual, na possibilidade de inserir-las em melhores posições posteriormente. Considere  $Custo^-(i, s) = f(s) - f_i^-(s)$  o custo associado à requisição  $i$  na solução  $s$ , tal que  $f_i^-(s)$  representa o custo da solução  $s$  sem atender a requisição  $i$ . A heurística ordena as requisições de maneira crescente conforme o  $Custo^-(i, s)$  e então

escolhe uma a ser removida da mesma maneira que a heurística de Shaw, e repete o processo  $\gamma$  vezes. O Algoritmo 3.2 descreve a implementação desta heurística, que não exige que uma requisição seja removida inicialmente. O Passo 5 ordena a lista de requisições atendidas  $L$  com base na função  $Custo^-(L_i, s)$  para cada  $L_i$  presente em  $L$ , e no Passo 6 é feita a remoção de uma requisição.

---

**Algorithm 3.2** Heurística de Remoção do Pior

---

- 1: **Entrada:** Solução  $s$ , e os parâmetros  $\gamma, \phi > 1$
  - 2: Considere  $D$  o conjunto de requisições removidas de  $s$
  - 3: Considere  $L$  a lista das requisições atendidas de  $s$
  - 4: **while**  $|D| < \gamma$  **do**
  - 5:     Ordene  $L \mid i < j \Rightarrow Custo^-(L_i, s) < Custo^-(L_j, s)$
  - 6:     Escolha um número aleatório  $y \in [0, 1]$
  - 7:      $i \leftarrow y^\phi |L|, s \leftarrow s - \{L_i\}, D \leftarrow D + \{L_i\}, L \leftarrow L - \{L_i\}$
  - 8: **end while**
  - 9: **return**  $s$
- 

### 3.2.3.5 Algoritmo de Inserção Guloso

Este algoritmo não é utilizado diretamente pelo ALNS, mas é empregado pelas heurísticas de inserção descritas seguir. Dada uma requisição  $i$  e uma solução  $s$ , ele encontra o ponto de inserção (dia, veículo, rota e aresta intermediária) que menor incrementa a função objetivo, e insere a requisição  $i$ .

### 3.2.3.6 Heurística de Inserção Gulosa Aleatória

Dada uma solução  $s$ , esta heurística seleciona uma demanda aleatória do conjunto  $D$  de requisições a serem atendidas e aplica o Algoritmo de Inserção Guloso até que não hajam mais demandas não atendidas.

### 3.2.3.7 Heurística de Inserção Gulosa Míope

Dado uma solução  $s$ , busca-se a requisição  $i$ , cujo a inserção pelo Algoritmo de Inserção Guloso gera o menor incremento na função objetivo, caso inserida. A requisição  $i$  é então inserida e uma nova requisição  $j$  é selecionada da mesma maneira até que não haja mais possíveis inserções ( $|D| = 0$ ).

### 3.2.3.8 Heurística de Inserção $K$ -Regret

Esta heurística tenta melhorar o comportamento míope das heurísticas de inserção. Dada uma solução  $s$  e o conjunto  $D$  de requisições não atendidas de  $s$ , esta heurística calcula o valor do “arrependimento” (*regret*) para cada requisição  $i \in D$  representado pela Equação (3.3). O “arrependimento” é igual a diferença da função objetivo entre

duas soluções as quais a requisição  $i$  é inserida na sua melhor rota ou na sua segunda melhor rota. A requisição  $i$  que possui o maior valor de “arrependimento” é escolhida para ser inserida na solução  $s$ .

Formalmente, seja  $w_{il}$  um índice que indica a rota para a qual a requisição  $i$  possui o  $l$ -ésimo menor custo de inserção, i. e.,  $\Delta f_i^{w_{il}} \leq \Delta f_i^{w_{il'}} \forall l < l'$ . Com isso, o valor do “ $k$ -arrependimento” da requisição  $i$  ( $k\text{-Regret}_i$ ) é dado pela soma dos  $k$  primeiros arrependimentos (Equação 3.3).

Desta maneira, o Algoritmo 3.3, a cada iteração, insere a requisição  $D_b$  que possui o maior valor de  $K\text{-Regret}_{D_b}$ .

$$k\text{-Regret}_i := \sum_{j=2}^k (\Delta f_i^{w_{ij}} - \Delta f_i^{w_{i1}}) \quad (3.3)$$

---

**Algorithm 3.3** Heurística de Inserção  $k - \text{Regret}$

---

- 1: **Entrada:** Solução  $s$
  - 2: Considere  $D$  o conjunto de requisições removidas de  $s$
  - 3: Considere  $L$  a lista das requisições atendidas de  $s$
  - 4: **while**  $|D| > 0$  **do**
  - 5:      $i \leftarrow D_b \mid \forall b \in D, b \neq a \Rightarrow k\text{-Regret}_{D_b} \geq k\text{-Regret}_{D_a}$
  - 6:      $s \leftarrow s + \{i\}, D \leftarrow D - \{i\}, L \leftarrow L + \{i\}$
  - 7:      $s \leftarrow s \cup \{i\}$
  - 8: **end while**
  - 9: **return**  $s$
- 

### 3.2.4 Camada Adaptativa

A cada iteração do ALNS, a pontuação do par de heurísticas selecionado aumenta com os parâmetros  $\tau_1$ ,  $\tau_2$  ou  $\tau_3$  quando encontra uma nova solução e a mesma é aceita. Caso seja encontrada uma nova melhor solução, a pontuação das heurísticas é aumentada em  $\tau_1$ . Caso encontre uma solução melhor que a atual, a pontuação é aumentada em  $\tau_2$  e caso a nova solução ainda passe no critério de aceitação do SA, a pontuação é aumentada em  $\tau_3$ . Quando o segmento termina, os pesos de cada heurística são atualizados e a pontuação de cada heurística reiniciada como zero para o próximo segmento. Seja  $\pi_i$  e  $o_i$ , respectivamente, a pontuação e o número de vezes que a heurística  $i$  foi utilizada em um segmento qualquer. Então, no próximo segmento o peso  $w_i$  é atualizado de acordo com a Equação (3.4), sendo que  $0 < \alpha_A < 1$  é um parâmetro chamado de *fator de reação* que indica a velocidade com a qual os pesos adaptativos mudam a cada segmento.

$$w_i \leftarrow (1 - \alpha_A)w_i + \alpha_A \frac{\pi_i}{o_i} \quad (3.4)$$

Como destacado em ROPKE e PISINGER [34], o parâmetro  $\alpha_A$  é um dos componentes determinantes para o correto funcionamento da camada adaptativa. O parâmetro  $\alpha_A$  então foi escolhido de maneira que ao passar 1/3 do número total de iterações, os pesos iniciais possuíssem uma contribuição muito baixa. Observando a Equação (3.4), ao longo das iterações, o peso inicial  $w_0$  terá valor  $w_0(1 - \alpha_A)^R$ , sendo  $R$  o (Número da Iteração Corrente)/ $\psi$ . Fazendo o desenvolvimento matemático, chega-se na Equação (3.5), sendo  $\psi$  o número de iterações de um segmento,  $K$  o número máximo de iterações e  $w_f$  o valor dos pesos quando o número de iterações for igual a  $K/3$ . Basta qualquer valor aceitável para  $w_f$  sendo que  $w_f \ll w_0$ , para a completa adaptação dos pesos em cerca de 1/3 das iterações. Os valores utilizados de  $w_0$  e  $w_f$  nos experimentos computacionais foram  $w_0 = 250$  e  $w_f = 0,01$ .

$$\alpha_A = 1 - (w_f/w_0)^{\frac{3\psi}{K}} \quad (3.5)$$

Os valores escolhidos para os parâmetros de  $\tau_1$ ,  $\tau_2$ , ou  $\tau_3$  foram  $\tau_1 = 50$ ,  $\tau_2 = 20$  e  $\tau_3 = 5$  como sugerido em RIBEIRO *et al.* [135] e RIBEIRO e LAPORTE [35].

### 3.2.5 Temperatura

Como já citado anteriormente na Seção 3.1, a temperatura influencia no do critério de aceitação do SA, e tem a função de limitar a aceitação de soluções inferiores à solução corrente ao longo da busca. Saindo de uma temperatura inicial  $T_0$ , quando a variação  $f(s') - f(s)$  é igual à temperatura da iteração atual, a probabilidade de aceitação, calculada pela Equação (3.6) é de aproximadamente 36,8%. E a cada iteração do ALNS, a Equação (3.7) é aplicada e a temperatura é gradativamente reduzida para que a probabilidade de se aceitar soluções piores seja reduzida, forçando o método a intensificar a busca nas proximidades da melhor solução encontrada. O parâmetro  $\alpha_T$  é conhecido como taxa de resfriamento.

$$e^{\frac{f(s) - f(s')}{T}} \quad (3.6)$$

$$T \leftarrow \alpha_T T \quad (3.7)$$

Desta maneira, sabe-se que a probabilidade de aceitação sempre dependerá do quociente da variação  $f(s') - f(s)$  pela temperatura  $T$ , então espera-se que  $T$  tenha grandeza proporcional à valores de  $f(s)$ . Logo a escolha da temperatura inicial  $T_0$  e da temperatura final  $T_F$  se mostram importantes para um critério de aceitação mais preciso ao longo do processo de busca. Desta maneira, valores de  $T_0$ ,  $\alpha_T$  e o número máximo de iterações  $K$  arbitrários para qualquer grupo de instâncias que possuem valores  $f(s)$  muito distintos, levará limiares de aceitação variados ao longo

das mesmas, o que pode não ser desejado.

A fim de que o número máximo de iterações e a temperatura de congelamento ( $T_F$ ) sejam ambos critérios de parada, o valor da taxa de resfriamento ( $\alpha_T$ ) foi escolhido considerando o valor das temperaturas ao longo das iterações, já que a cada iteração a temperatura é modificada. Sabendo que o algoritmo possui um valor  $K$  de iterações, pode-se usar relações iniciais ( $Iteracao = 0 \Rightarrow T = T_0$ ) e finais ( $Iteracao = K \Rightarrow T = T_F$ ), com um pouco de desenvolvimento algébrico para obter a Equação (3.8).

$$\alpha_T = \sqrt[K-1]{\frac{T_F}{T_0}}. \quad (3.8)$$

Desta maneira,  $\alpha_T$  deixa de ser um parâmetro e passa a ser dependente de  $T_0$ ,  $T_F$  e  $K$ . Finalmente, a temperatura inicial  $T_0$  foi escolhida como sendo 1/3 do valor da função objetivo da solução inicial, e a temperatura de congelamento ( $T_F$ ) foi escolhida como  $\frac{1}{500}$  do valor da solução inicial, para a cristalização das soluções.

Variados valores de  $T_0$  e  $T_F$  foram testados no processo de calibração do ALNS, mas nenhuma alteração significativa foi observada nos resultados.

### 3.2.6 Estratégias de Paralelismo para Avaliar Estagnação e Perturbações

A implementação proposta considera  $\eta$  execuções simultâneas do ALNS com o intuito de aproveitar a capacidade ociosa das novas CPUs que possuem vários núcleos de processamento. Mas a execução simultânea de vários ALNS sem aproveitar as informações obtidas durante as buscas, nada mais é que executar o método  $\eta$  vezes e selecionar a melhor solução encontrada em todas. Desta maneira foi criada a mecânica de avaliação de estagnação seguida de perturbações afim de se aproveitar as informações geradas pelas  $\eta$  execuções simultâneas.

Mais especificamente, as melhores soluções viáveis únicas encontradas em todas as execuções simultâneas do ALNS são armazenadas em uma lista  $A$  (Conjunto Elite) que é ordenada de maneira crescente. A cada segmento de  $\psi$  iterações, as execuções paralelas são pausadas e analisadas. Verifica-se se a melhor solução viável encontrada (a primeira posição da lista  $A - A_1$ ) foi melhorada durante o último segmento de  $\psi$  iterações. Caso negativo um contador de estagnação de valor inicial zero é incrementado com valor igual a 1, caso contrário, o contador é reiniciado com valor zero. Caso o contador de estagnação atinja um limiar  $\rho$ , isto indica que as execuções estão a  $\rho$  segmentos sem melhorar a melhor solução viável encontrada. Então é iniciada a mecânica de perturbação, tal que as  $\theta$  execuções com piores soluções correntes das  $\eta$  soluções correntes de cada execução simultânea são descartadas

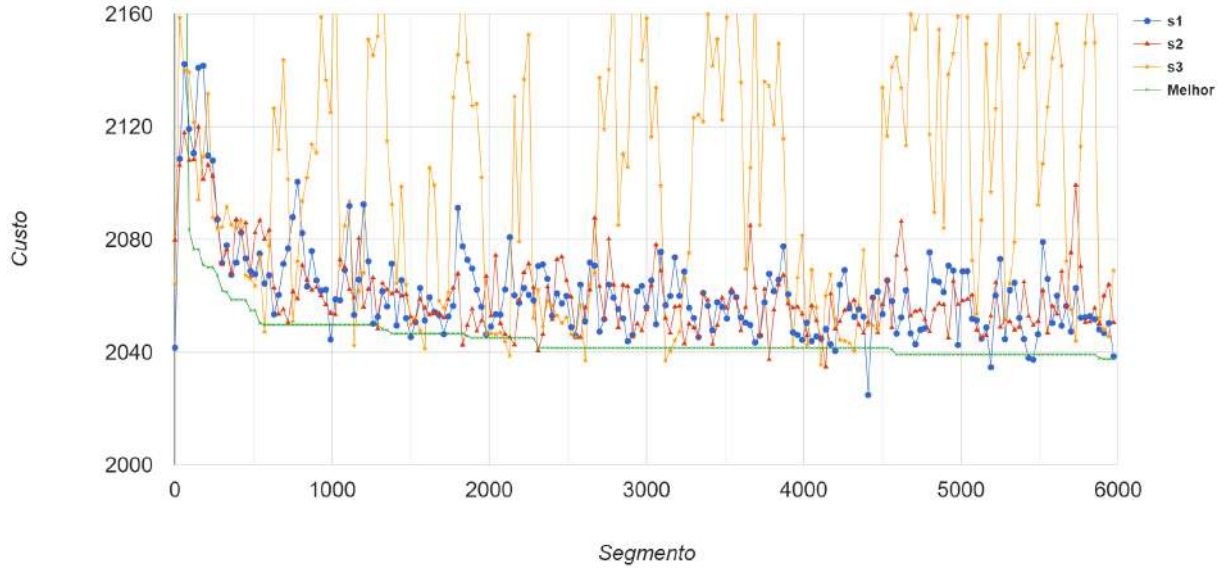


Figura 3.1: Comportamento oscilante das soluções correntes ao longo dos segmentos na Instância  $p5$  do PRVP.

e recebem uma nova solução inicial como estratégia de diversificação, ou uma das  $\epsilon$  soluções do conjunto elite  $A$ . Cada vez que a execução é perturbada, o contador de estagnação é reiniciado com valor igual a zero.

Esta estratégia permite diversificar a busca e compartilhar informações obtidas nas  $\eta$  execuções paralelas.

A Figura 3.1 mostra uma execução do ALNS com 3 soluções simultâneas e a melhor solução viável encontrada “Melhor”. As soluções  $s1$  e  $s2$  não são perturbadas, enquanto a solução  $s3$  é perturbada a cada  $\rho$  segmentos sem melhora na solução “Melhor”. Estas perturbações começam a ser visualizadas em torno de 600 segmentos. O comportamento oscilante é bem observado nas soluções  $s1$  e  $s2$  devido às constantes contrações e relaxamentos das penalidades. As soluções  $s1$  e  $s3$  atingem valores menores que a melhor solução corrente algumas vezes ao longo da execução, porém estas soluções são inviáveis. Como consequência, novas melhores soluções viáveis normalmente são encontradas durante as curvas de contração das penalidades.

### 3.2.7 Código Principal do ALNS

Descrevendo formalmente, o Algoritmo 3.4 apresenta o ALNS proposto. No Passo 2 é iniciada a lista das  $\epsilon$  melhores soluções viáveis  $A$ , que é uma lista global acessada por todas as execuções do ALNS. No Passo 3 é iniciada a variável  $A_1^{old}$  que representa a melhor solução encontrada no segmento anterior. No Passo 5 é construída uma

solução inicial para o problema conforme descrito na Subseção 3.2.2. No Passo 6, a melhor solução  $s_{best}$  recebe a solução inicial criada no passo anterior.

---

**Algorithm 3.4** Adaptive Large Neighborhood Search

---

```

1: Entrada: Uma instância de um PRV-FH-P-MT
2:  $A \leftarrow \{\}$ 
3:  $A_1^{old} \leftarrow SoluçãoInicial()$ 
4: Inicia as  $\eta$  Execuções Paralelas
5:  $s \leftarrow SoluçãoInicial()$ 
6:  $s_{best} \leftarrow s$ 
7: repeat
8:    $N^- \leftarrow SeleccionaH^-$ 
9:    $s' \leftarrow N^-(s)$ 
10:   $N^+ \leftarrow SeleccionaH^+$ 
11:   $s' \leftarrow N^+(s')$ 
12:  Aplica critério do SA
13:  Atualiza  $\pi_{N^+}$ ,  $\pi_{N^-}$ ,  $o_{N^+}$  e  $o_{N^-}$ 
14:  if Viavel( $s'$ ) e  $s' \notin A$  e  $f(s') < Min\{f(A_k)\} \forall A_k \in A$  then
15:     $A \leftarrow A \cup s'$ 
16:  end if
17:   $T \leftarrow \alpha_T T$ 
18:  if Final do segmento é atingido then
19:    Pausa as  $\eta$  Execuções Paralelas
20:    Atualiza os pesos adaptativos de  $H^+$  e  $H^-$ 
21:    Atualiza os pesos de viabilidade  $\omega_q$  e  $\omega_t$ 
22:     $CalculaSolucoesPertubadas()$ 
23:    Continua as  $\eta$  Execuções Paralelas
24:    if  $PerturbaSolucão(s_{best})$  then
25:       $s_{best} \leftarrow SoluçãoInicial()$  ou  $A_x | x \in \{1, \dots, \epsilon\}$ 
26:       $s \leftarrow s_{best}$ 
27:    end if
28:  end if
29: until Critério de parada seja atendido
30: Finaliza as  $\eta$  Execuções Paralelas
31: return  $A_1$ 

```

---

No Passo 7 é iniciado o ciclo principal do Algoritmo. Do Passos 8 ao 11 são selecionadas e aplicadas as heurísticas de destruição e reconstrução. No Passo 12 é aplicado o critério de aceitação do SA que é seguido dos Passos 14 até 16 nos quais caso  $s'$  seja viável e não pertencente à Lista  $A$ , então ele é adicionado a  $A$ . No Passo 17 a temperatura é atualizada.

Nos Passos 18 a 28 ocorrem as instruções que são executadas quando o número de iterações correntes atinge o tamanho de um segmento  $\psi$ . Estes passos diferenciam essa implementação do ALNS em relação às aplicadas até então. No Passo 22 é executado o Algoritmo 3.5, descrito formalmente logo a seguir, que verifica se alguma solução será perturbada. Caso a execução seja selecionada para ser perturbada, no



Passo 25  $s$  e  $s_{best}$  desta execução recebem uma nova solução inicial ou uma das  $\epsilon$  soluções elite. Essa decisão ocorre de maneira aleatória. Finalmente, no Passo 28 as instruções entre segmentos terminam e o algoritmo segue para o laço principal no Passo 29. No Passo 31 o algoritmo retona a melhor solução obtida por todas as execuções.

---

**Algorithm 3.5** *CalculaSolucoesPerturbadas()*

---

```

1: Entrada: Lista  $K$  de Soluções  $s_{best}$  correntes, a Lista de Soluções Viáveis solução  $A_1^{old}$  que é  $A_1$  da execução anterior deste método
2:  $PerturbaSolucao(K_i) \leftarrow \text{Falso} \forall i \in \{1, \dots, \eta\}$ 
3: if  $A_1 \neq A_1^{old}$  then
4:    $Estagnado \leftarrow 0$ 
5:    $A_1^{old} \leftarrow A_1$ 
6:   return
7: end if
8:  $Estagnado \leftarrow Estagnado + 1$ 
9: if  $Estagnado = \rho$  then
10:  Ordene  $K \mid i < j \Rightarrow f(K_i) > f(K_j)$ 
11:   $PerturbaSolucao(K_i) \leftarrow \text{Verdadeiro} \forall i \in \{1, \dots, \theta\}$ 
12:   $Estagnado \leftarrow 0$ 
13: end if
14: return

```

---

O Algoritmo 3.5 verifica quais execuções do ALNS devem ter suas soluções descartadas. Tem como entrada o conjunto de soluções  $K$  com todas as melhores soluções  $s_{best}$  das execuções dos ALNS correntes, a lista de soluções viáveis  $A$  e  $A_1^{old}$  que é o  $A_1$  do segmento anterior. No Passo 2, todas as  $\eta$  soluções tem o  $PerturbaSolucao(s_{best})$  marcado como *Falso*, que é seguido dos Passos 3 até 6 que verificam se a melhor solução viável foi melhorada no último segmento, se sim, o contador  $Estagnado$  é reiniciado e o método interrompido, caso contrário é a execução continua com o incremento de  $Estagnado$  no Passo 8 e é feita a verificação se  $Estagnado$  atingiu o limiar de estagnação  $\rho$  no Passo 9. Se verdadeiro, indica que a  $\rho$  segmentos, não ocorreu nenhuma melhora na melhor solução viável, então nos Passos 10 a 12 as soluções são ordenadas de acordo com os maiores valores de  $f(A_i)$  (viáveis ou não) e as  $\theta$  soluções com os maiores valores de  $f(s_{best})$  têm o atributo  $PerturbaSolucao(s_{best})$  definidos como *Verdadeiro*, indicando as soluções a serem descartadas. Finalmente o contador  $Estagnado$  é reiniciado no Passo 12 e em seguida o Algoritmo 3.5 é encerrado.

### 3.3 Considerações Finais

Este capítulo apresentou o ALNS para o PRV-FH-P-MT que considera diversas estratégias para gerar soluções de boa qualidade. Ele emprega paralelismo e estra-

tégias de perturbação para sair de ótimos locais que não foram até então propostos na literatura. Sendo assim, o próximo capítulo apresenta os resultados computacionais para diversas instâncias propostas na literatura para os problemas abordados neste trabalho.

# Capítulo 4

## Testes Computacionais

Este capítulo detalha os experimentos computacionais realizados nos grupos de instâncias referentes aos problemas relacionados ao PRV-P-FH-MT. A Seção 4.1 apresenta como foi o processo de calibração dos parâmetros do ALNS, em seguida os resultados de cada grupo de instâncias são exibidos.

O ALNS foi implementado em linguagem C e compilado com o compilador GCC V6.2.1 com o parâmetro de otimização das instruções “-Ofast”, que “sacrifica” espaço em disco e memória para ganho de velocidade de processamento. O processamento paralelo foi implementado para sistemas operacionais UNIX, utilizado com a biblioteca “pthread.h”. Para os sistemas operacionais Windows utilizou-se a biblioteca “windows.h” para processamento paralelo. Durante a implementação testes foram realizados em ambos os sistemas operacionais, mas o processo de calibração e experimentos de todas as instâncias foram realizados em uma máquina com processador Intel Core i7 6700k 4,6Ghz, 16GB de memória RAM DDR4@3200Mhz e sistema operacional Windows 10 64 Bits. Este processador possui 4 núcleos e capacidade de processamento de até 8 *threads* simultâneas.

O ALNS foi testado em instâncias conhecidas da literatura do PRVC, PRVP, PRVMD e PRVFNH. Para o PRVMT, como não existem um grupo de instâncias unificado, optou-se por utilizar as instâncias de PRVFNH compatíveis com cenários de múltiplas viagens e comparar os resultados com o cenário de somente uma viagem. E para efeito de comparação dos resultados encontrados com quaisquer outros da literatura, foi mantida a nomenclatura usual da literatura para cada instância.

Cada grupo de instâncias foi tratado como um caso particular do PRV-FH-P-MT. Por exemplo, o PRVP é um PRV-FH-P-MT no qual o limite de viagens de um veículo por período é igual a um com apenas uma classe de veículos sem custos de utilização e custo de deslocamento igual a um. O mesmo raciocínio vale para todas as outras instâncias testadas. Com exceção de uma pequena modificação necessária para as instâncias de PRV com múltiplos depósitos que foi o vínculo de um depósito para cada período, pois em um PRVP tradicional, todos os dias os veículos deixam

o mesmo depósito que é único.

## 4.1 Calibração dos Parâmetros

Para a escolha do melhor valor para cada um dos parâmetros, foram realizados testes computacionais variando o valor de um parâmetro enquanto os demais se mantiveram constantes. Todos os testes para calibração foram realizados com 15 execuções do ALNS com no mínimo 3000 segmentos em cada execução. Para efeito de calibração, considerou-se a Instância *p5* do PRVP que possui 50 clientes, 153 requisições e 6 períodos. Optou-se por utilizá-la devido ao seu grau de dificuldade verificado durante os testes preliminares. A etapa de calibração foi executada de maneira linear, ou seja, uma vez definido o valor de um parâmetro, este não foi alterado enquanto os demais eram ajustados. Este processo de calibração foi indicado por PISINGER e ROPKE [127] para o ALNS.

O processo de calibração foi aplicado ao número de requisições a serem removidas ( $\gamma$ ), ao tamanho do segmento ( $\psi$ ), número máximo de iterações ( $K$ ), número de soluções simultâneas ( $\eta$ ), número de soluções perturbadas ( $\theta$ ), tamanho do conjunto elite ( $\epsilon$ ), limiar de estagnação ( $\rho$ ), taxa de atualização de peso de penalidade para soluções inviáveis ( $\alpha^+$ ), taxa de atualização de peso de penalidade para soluções viáveis ( $\alpha^-$ ), peso inicial da penalidade de sobrecarga ( $\omega_0^g$ ) e ao peso inicial da penalidade de sobretempo ( $\omega_0^t$ ).

Quanto aos parâmetros da camada adaptativa e ao do critério de aceitação baseado em temperatura, eles foram selecionados com base em seus comportamentos matemáticos e estatísticos como já detalhados nas Subseções 3.2.5 e 3.2.4.

O número de soluções a serem descartas por segmento ( $\theta$ ) foi escolhido como  $\eta/3$ , para que as outras  $2\eta/3$  soluções continuem suas buscas quando uma estagnação é detectada.

A variação do tamanho da elite  $\epsilon$  apresentou uma melhora de 1 até 10, depois começa a cair a qualidade dos resultados médios. Então o valor escolhido foi  $\epsilon = 10$ .

A variação de  $\omega_0^g$  e  $\omega_0^t$  não mostraram impacto algum nos resultados, que era de se esperar, pois a atualização dos pesos ocorre de maneira geométrica, logo com poucos segmentos já não há valor residual destas constantes. Logo ambos tiveram valor escolhido de  $\omega_0^g = 0, 2$  e  $\omega_0^t = 0, 2$ .

A variação do parâmetro  $\phi$  não resultou em impacto significativo nos testes, então foi escolhido como  $\phi = 4$  de acordo com as implementações de RIBEIRO e LAPORTE [35] e DEMIR *et al.* [128].

O parâmetro  $\rho$  que é o limiar de estagnação foi fixado como 100 segmentos após experimentos com passos incrementais de 10 segmentos a partir de 50 até 200.

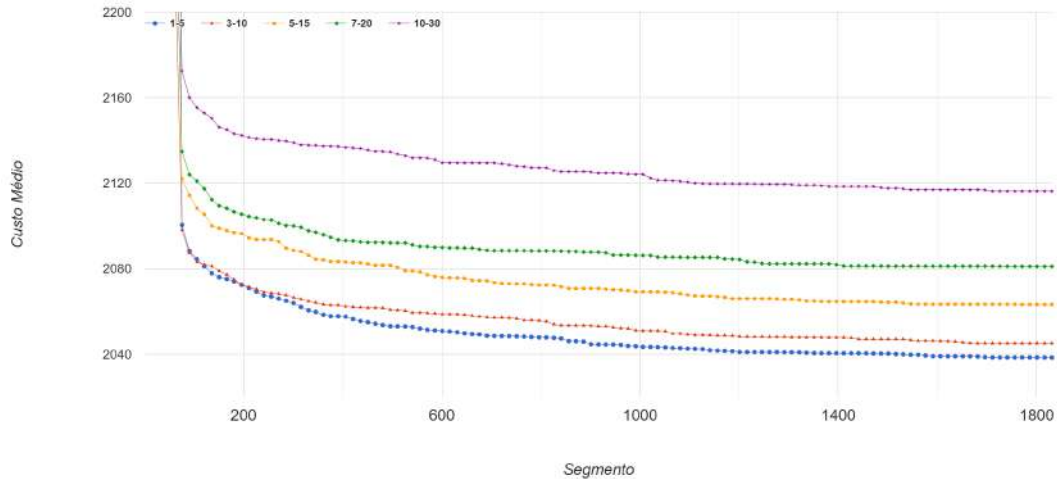


Figura 4.1: Impacto dos intervalos de  $\gamma$  ao longo dos segmentos na Instância  $p5$  do PRVP.

#### 4.1.1 Número de Elementos Removidos e Re-inseridos por Iteração

No processo de calibração do número de requisições a serem removidas ( $\gamma$ ) verificou-se como relatado por ROPKE e PISINGER [34], que esse parâmetro possui um grande impacto nos resultados e no tempo de solução. O parâmetro  $\gamma$  foi obtido com base nas seguintes faixas de percentuais:  $[1-5]\%$ ,  $[3-10]\%$ ,  $[5-10]\%$ ,  $[7-10]\%$ ,  $[5-15]\%$ ,  $[7-20]\%$ ,  $[7-25]\%$ ,  $[10-25]\%$  e  $[10-30]\%$ . Por exemplo, para a faixa  $\gamma = [1-5]\%$ , a cada iteração do ALNS, o número de requisições removidas é um número aleatório entre 1% e 5% do total.

A Figura 4.1 mostra o valor das soluções geradas pelo ALNS para diferentes percentuais de remoção em função do número de segmentos, para a Instância  $p5$  do PRVP. Esses resultados são valores médios de 15 execuções do algoritmo e 3000 segmentos em cada execução, porém a figura apresenta os resultados dos primeiros 2000 segmentos para facilitar sua visualização. E a Figura 4.2 mostra o valor das soluções geradas pelo ALNS e tempos de processamento em função de diferentes intervalos de  $\gamma$ , para a instância  $p5$  do PRVP.

Desta maneira, o intervalo de  $\gamma$  escolhido foi  $\gamma = [1-5]\%$  do número de requisições, pois apresentou os melhores resultados e em um menor tempo de processamento.

#### 4.1.2 Tamanho do Segmento

O tamanho do segmento ( $\psi$ ) foi obtido com base nas seguintes valores testados: 25, 50, 100 e 200.

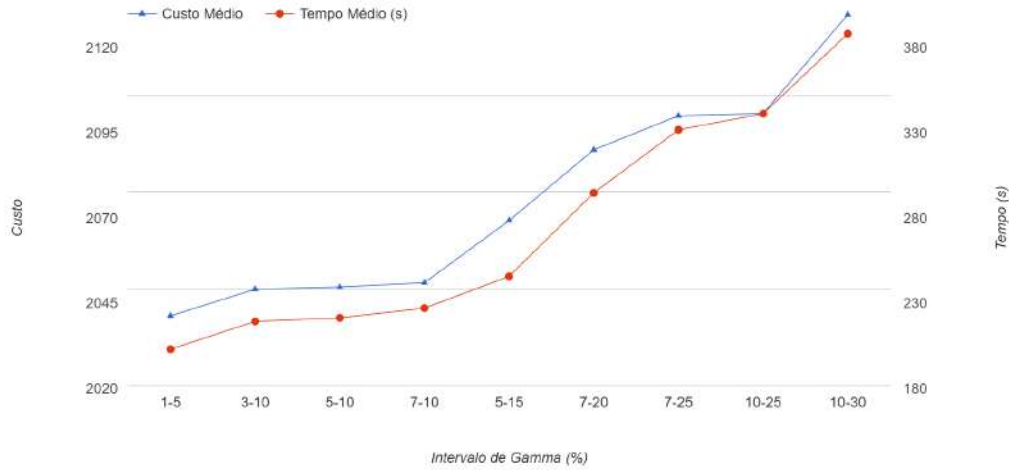


Figura 4.2: Impacto dos intervalos de  $\gamma$  em relação ao tempo gasto e o custo das soluções na Instância  $p5$  do PRVP.

A Figura 4.3 mostra o valor das soluções geradas pelo ALNS para diferentes valores de  $\psi$  em função do número de segmentos, para a Instância  $p5$  do PRVP.

Desta maneira, as curvas  $\psi = 50$  e  $\psi = 100$  se mostraram muito semelhantes e com resultados intermediários entre as curvas de  $\psi = 25$  e  $\psi = 200$ . Assim optou-se por adotar  $\psi = 50$  por apresentar melhores resultados por tempo de processamento.

### 4.1.3 Impacto dos Mecanismos de Perturbação

Afim de avaliar a eficácia dos mecanismos de perturbação propostos nesta dissertação, quatro diferentes cenários foram considerados:

1. **Sem Perturbações.** Uma vez que  $\rho$  segmentos são atingidos sem melhora da melhor solução obtida ( $A_1$ ), nenhuma perturbação é aplicada.
2. **Somente Diversificação.** Uma vez que  $\rho$  segmentos são atingidos sem melhora da melhor solução obtida ( $A_1$ ), as  $\theta$  piores soluções das  $\eta$  execuções paralelas do ALNS são substituídas por novas soluções iniciais.
3. **Somente Intensificação.** Uma vez que  $\rho$  segmentos são atingidos sem melhora da melhor solução obtida ( $A_1$ ), as  $\theta$  piores soluções das  $\eta$  execuções paralelas do ALNS são substituídas por alguma das  $\epsilon$  soluções de  $A$ .
4. **Com Diversificação e Intensificação.** Uma vez que  $\rho$  segmentos são atingidos sem melhora da melhor solução obtida ( $A_1$ ), as  $\theta$  piores soluções das  $\eta$  execuções paralelas do ALNS são substituídas por alguma das  $\epsilon$  soluções de  $A$  ou por novas soluções iniciais. Seleção esta feita de maneira aleatória.

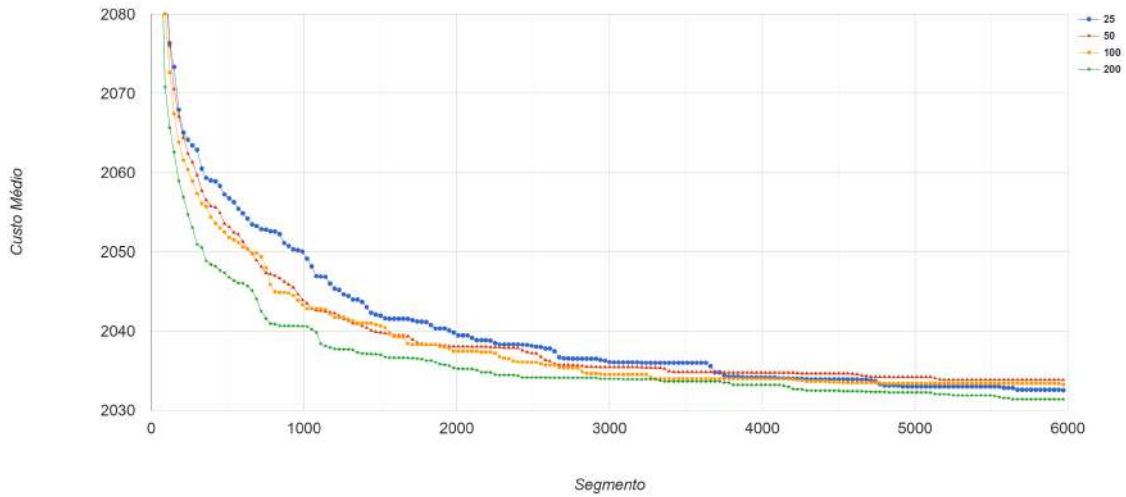


Figura 4.3: Impacto do tamanho do segmento  $\psi$  em relação ao custo médio das soluções na Instância  $p5$  do PRVP.

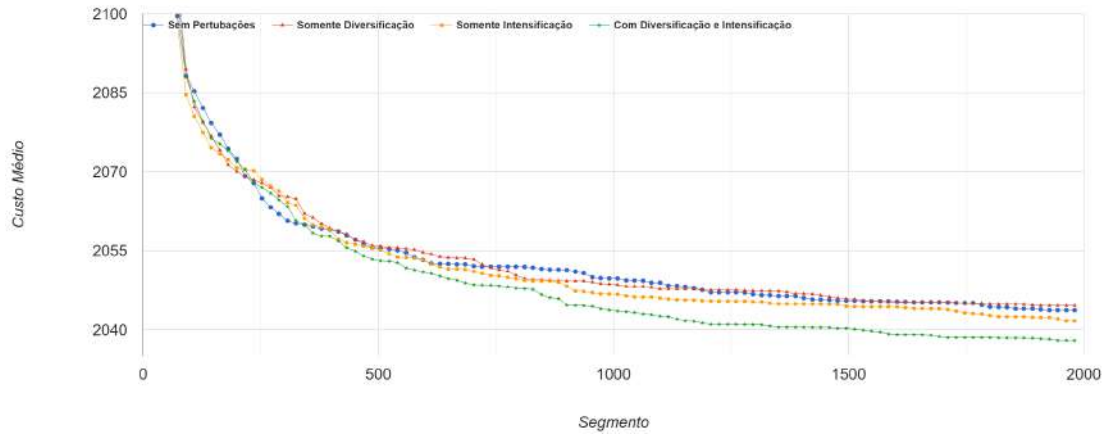


Figura 4.4: Impacto dos mecanismos de perturbação em relação ao custo médio das soluções na Instância  $p5$  do PRVP.

A Figura 4.4 mostra o valor da função objetivo das soluções geradas pelo ALNS com diferentes mecanismos de perturbação em função do número de segmentos, para a Instância  $p5$  do PRVP. Esses resultados são valores médios de 15 execuções do algoritmo com 3000 segmentos em cada execução, porém a figura apresenta os resultados dos primeiros 2000 segmentos para facilitar sua visualização.

Observa-se que os comportamentos médios para as curvas de “Sem Perturbações” e “Somente Diversificação” são muito parecidas, e a curva de “Somente Intensificação” com valores levemente superiores. Mas nota-se que curva “Com Diversificação e Intensificação” se mostra em média, com resultados superiores às anteriores. Logo, conclui-se que os mecanismos de perturbação propostos apresentam resultados positivos.

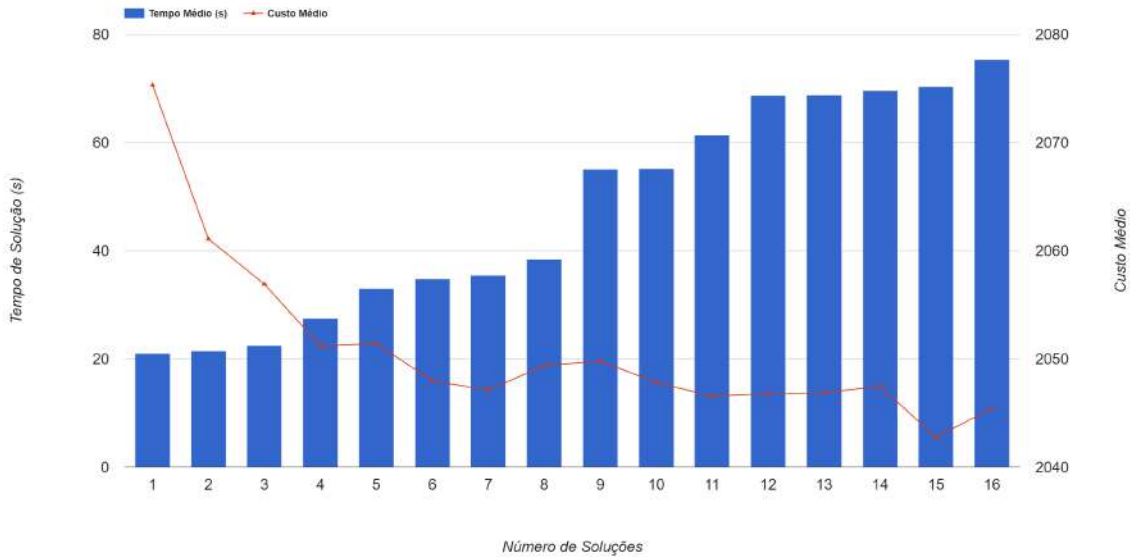


Figura 4.5: Impacto de diferentes valores de  $\eta$  em relação ao custo médio e tempo de solução na Instância  $p5$  do PRVP.

#### 4.1.4 Número de Soluções Simultâneas

O número de soluções simultâneas ( $\eta$ ) foi obtido com base nos valores testados de 1 até 16. Relembrando que para cada solução simultânea, é criada uma *thread* e a CPU utilizada no processamento possui 4 núcleos e capacidade de processar até 8 *threads* simultâneas.

A Figura 4.5 mostra o valor da função objetivo das soluções geradas pelo ALNS e tempos de processamento em função de diferentes valores de  $\eta$ , para a instância  $p5$  do PRVP. Verifica-se um decréscimo médio constante do custo médio com o aumento de  $\eta$  que aparenta estável a partir de  $\eta = 7$ . Também um acréscimo nulo no tempo de processamento até  $\eta = 3$ , o que era esperado pois 3 núcleos físicos estão dedicados à execução, e todas as outras tarefas do sistema operacional ficam com o quarto núcleo. Mas a partir de  $\eta = 4$ , o método começa a disputar recursos com os outros aplicativos em execução no sistema operacional, mas ainda assim apresentam um impacto no tempo de processamento muito baixo até  $\eta = 8$  que é o número máximo de *threads* simultâneas que o processador suporta. Para  $\eta = 9$  observa-se um impacto grande no tempo de processamento. O que leva a crer que o sistema operacional Windows 10 64 bits cria uma fila de *threads* por processo, já que a capacidade da CPU foi excedida. A Figura 4.5 mostra uma estabilidade do custo médio para  $\eta = 11, 12$  e  $13$ . Desta forma o valor escolhido foi de  $\eta = 12$ .



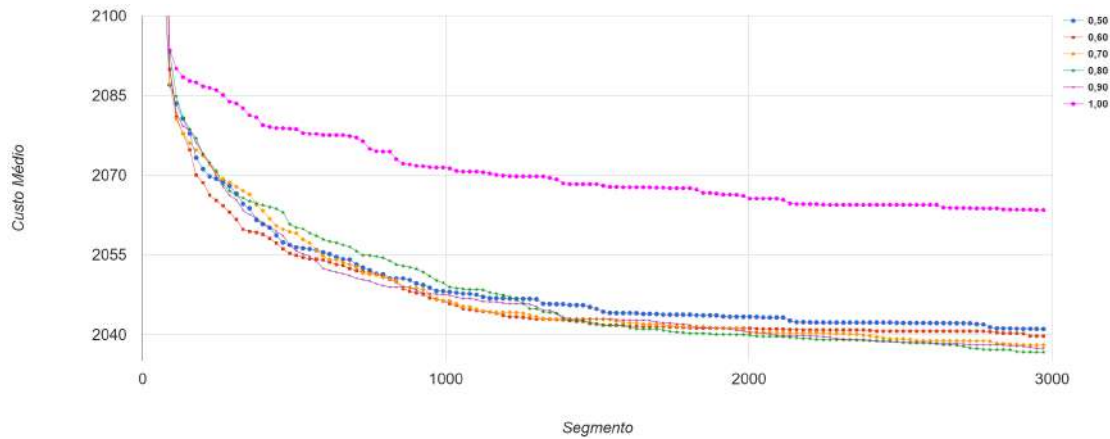


Figura 4.6: Impacto de  $\alpha^-$  relação ao custo médio das soluções na Instância  $p5$  do PRVP.

#### 4.1.5 Taxas de Atualização dos Pesos das Penalidades

A taxa de atualização de peso de penalidade para soluções viáveis ( $\alpha^-$ ) foi obtida com base nos seguintes valores testados: 0,50; 0,60; 0,70; 0,80; 0,90 e 1,00.

A Figura 4.6 mostra o valor das soluções geradas pelo ALNS para diferentes valores de  $\alpha^-$  em função do número de iterações, para a Instância  $p5$  do PRVP. Percebe-se que em média as soluções encontradas por todos os valores de  $\alpha^-$  testados são muito próximas, exceto para o valor de  $\alpha^- = 1,00$  que é quando o peso não recebe modificação quando uma solução viável é encontrado, não incentivando a exploração de soluções inviáveis. Com base nestes resultados, optou-se por  $\alpha^- = 0,8$ .

A taxa de atualização de peso de penalidade para soluções viáveis ( $\alpha^+$ ) foi obtida com base nos seguintes valores testados: 1,04; 1,10; 1,15; 1,20 e 1,25.

A Figura 4.7 mostra o valor das soluções geradas pelo ALNS para diferentes valores de  $\alpha^+$  em função do número de iterações, para a Instância  $p5$  do PRVP. Percebe-se que em média as soluções encontradas por todos os valores de  $\alpha^+$  testados são muito próximas. Com base nestes resultados, optou-se por  $\alpha^+ = 1,15$ .

#### 4.1.6 Número Máximo de Segmentos e Iterações

Finalmente, o número máximo de segmentos foi o último parâmetro a ser calibrado. Para uma maior estabilidade do desempenho médio, foram realizados 20 testes computacionais para a calibração deste parâmetro.

A Figura 4.8 mostra o valor das soluções geradas pelo ALNS em função do número de segmentos, para a Instância  $p5$  do PRVP. Esses resultados são valores médios ao longo de 15 execuções do algoritmo. Verifica-se nesta figura que existe uma tendência de redução do melhor valor encontrado com o aumento do número de iterações. Entretanto, percebe-se também uma estabilização por volta de 4500

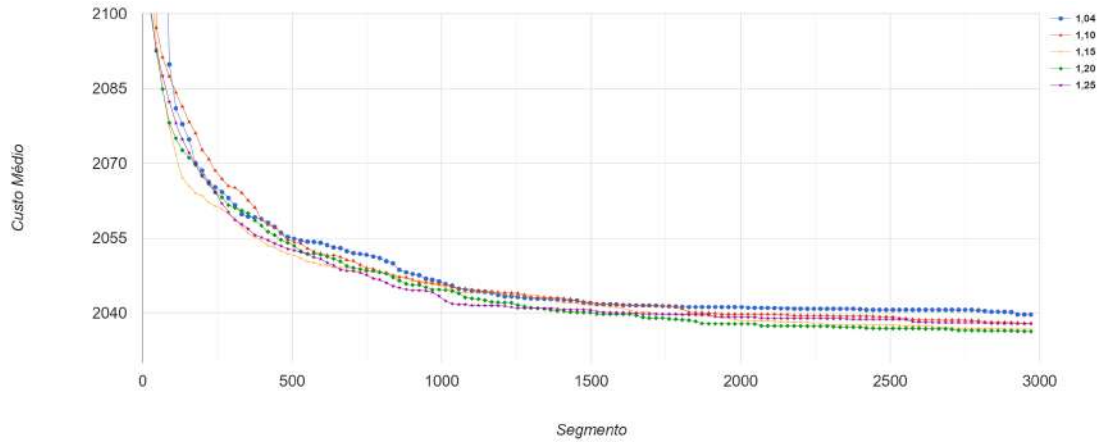


Figura 4.7: Impacto de  $\alpha^+$  em relação ao custo médio das soluções na Instância  $p_5$  do PRVP.

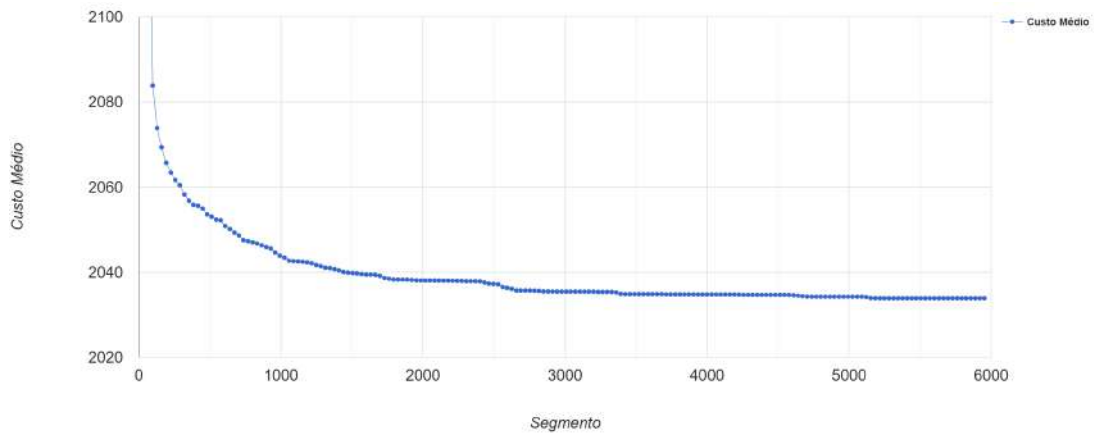


Figura 4.8: Valor médio da melhor solução encontrada ao longo dos segmentos na Instância  $p_5$  do PRVP.

segmentos do ALNS. Desta maneira o número máximo de segmentos foi fixado em 5000. Logo, o número máximo de iterações é de  $K = 5000 \times \psi = 2500000$ .

#### 4.1.7 Resultado da Calibração

A Tabela 4.1 apresenta o resultado final do processo de calibração. Com isso, todos os testes computacionais foram então realizados considerando 10 execuções do ALNS para cada instância.

## 4.2 Instâncias do PRV Clássico

O conjunto de instâncias do PRVC testado foi proposto em CHRISTOFIDES *et al.* [136] que possui de 50 a 199 clientes. Essas instâncias são as mais utilizadas de

Tabela 4.1: Valor utilizado dos parâmetros para todas as instâncias

Parâmetro	Descrição	Valor
$\gamma$	Número de Requisições a serem removidas	1-5%
$\psi$	Tamanho do Segmento	50
$K$	Número Máximo de Iterações	2500000
$w_0$	Valor Inicial dos Pesos Adaptativos	250
$w_f$	Valor Residual em $K/3$ Iterações	0,01
$\alpha_A$	Taxa de Recálculo de Pesos	Equação (3.5)
$\eta$	Número de Soluções Simultâneas	12
$\theta$	Número de Soluções Perturbadas	$\eta/3$
$\epsilon$	Tamanho do Conjunto Elite	10
$\rho$	Limiar de Estagnação	100
$\alpha^+$	Taxa de Atualização de Peso de Penalidade para Soluções Inviáveis	1,15
$\alpha^-$	Taxa de Atualização de Peso de Penalidade para Soluções Viáveis	0,80
$\omega_0^q$	Peso Inicial da Penalidade de Sobrecarga	0,2
$\omega_0^t$	Peso Inicial da Penalidade de Sobretempo	0,2
$\alpha_T$	Taxa de Atualização da Temperatura	Equação (3.8)

PRVC na literatura.

A Tabela 4.2 apresenta os resultados do ALNS para as instâncias do PRVC de CHRISTOFIDES *et al.* [136]. A primeira coluna possui o nome da instância, a segunda o número de clientes e a terceira o tipo de restrição do problema, sendo “C” somente restrições de carga, e “CT” restrições de carga e tempo. A quarta coluna “MSC” apresenta a melhor solução conhecida. A quinta coluna apresenta o resultado médio de 10 processamentos do método de NAGATA e BRÄYSY [137] que é o método com os melhores resultados médios para as instâncias de PRVC atualmente. A coluna “PR” é o resultado médio de 10 processamentos do método de PISINGER e ROPKE [127] com  $50 \times 10^3$  iterações, que é o ALNS original. As três últimas colunas apresentam os resultados do ALNS: o resultado médio de 10 execuções, o melhor valor encontrado e o tempo médio de processamento em segundos. Os valores em negrito representam a melhor solução média encontrada ou a melhor de todas as execuções. Esta notação é empregada também nos resultados das demais variações do PRV estudadas.

Na Tabela 4.2 observa-se o bom desempenho do ALNS proposto em relação aos métodos de NAGATA e BRÄYSY [137] e PISINGER e ROPKE [127]. Em termos de soluções médias, o ALNS foi capaz de encontrar oito valores iguais às melhores soluções conhecidas. Em relação às melhores soluções obtidas pelo ALNS, o método foi capaz de encontrar dez melhores soluções conhecidas. Por outro lado, o tempo computacional variou de pouco mais de um segundo até 892,69 segundos na instância mais complexa.

O ALNS também se mostrou robusto, ou seja, seus valores médios estão próximos das melhores soluções encontradas durante as 10 execuções do método. A instância *p05* apresentou o maior desvio de 0,98%, ou seja, a solução média do ALNS é apenas 0,98% pior do que a melhor solução encontrada durante as 10 execuções do método.

Tabela 4.2: Instâncias do PRVC.

Instância	$n$	Tipo	MSC	NB	PR	ALNS		T(s)
				Média	Média	Média	Melhor	
p01	50	C	<b>524,61</b>	<b>524,61</b>	<b>524,61</b>	<b>524,61</b>	<b>524,61</b>	0,15
p02	75	C	<b>835,26</b>	835,61	839,62	<b>835,26</b>	<b>835,26</b>	59,68
p03	100	C	<b>826,14</b>	<b>826,14</b>	826,99	826,77	<b>826,14</b>	163,25
p04	150	C	<b>1028,42</b>	<b>1028,42</b>	1034,20	1031,49	1029,56	537,69
p05	199	C	<b>1291,45</b>	<b>1291,84</b>	1306,63	1304,58	1298,30	892,69
p06	50	CT	<b>555,43</b>	<b>555,43</b>	<b>555,43</b>	<b>555,43</b>	<b>555,43</b>	1,17
p07	75	CT	<b>909,68</b>	910,41	911,78	<b>909,68</b>	<b>909,68</b>	4,38
p08	100	CT	<b>865,94</b>	<b>865,94</b>	866,97	<b>865,94</b>	<b>865,94</b>	10,99
p09	150	CT	<b>1162,55</b>	<b>1162,56</b>	1167,68	1164,29	<b>1162,55</b>	527,73
p10	199	CT	<b>1395,85</b>	<b>1398,30</b>	1410,27	1412,09	1408,38	876,67
p11	120	C	<b>1042,11</b>	<b>1042,11</b>	1042,46	<b>1042,11</b>	<b>1042,11</b>	193,18
p12	100	C	<b>819,56</b>	<b>819,56</b>	<b>819,56</b>	<b>819,56</b>	<b>819,56</b>	1,07
p13	120	CT	<b>1541,14</b>	<b>1542,99</b>	1543,54	1544,79	1544,24	353,96
p14	150	CT	<b>866,37</b>	<b>866,37</b>	<b>866,37</b>	<b>866,37</b>	<b>866,37</b>	99,06
<b>Médias</b>			<b>976,04</b>	<b>976,45</b>	979,72	978,78	977,72	

Com relação ao tempo computacional do ALNS, observa-se que ele é similar aos obtidos pelos métodos de NAGATA e BRÄYSY [137] e PISINGER e ROPKE [127].

### 4.3 Instâncias do PRV Periódico

Dois grupos de instâncias do PRVP foram utilizados nos testes computacionais. O primeiro conjunto ( $p01 - p32$ ) tem as instâncias  $p1 - p10$  propostas por EILON *et al.* [138] para o PRV clássico e adaptadas para o PVRP por CHRISTOFIDES e BEASLEY [87]. A instância  $p11$  proposta por RUSSELL e IGO [86], as instâncias  $p12$  e  $p13$  propostas por RUSSELL e GRIBBIN [88], e as instâncias  $p14 - p32$  foram propostas por CHAO *et al.* [91].

A Tabela 4.3 apresenta os resultados do ALNS para o primeiro grupo de instâncias do PRVP. A primeira coluna possui o nome da instância, da segunda até a quinta coluna é o número de clientes, número de períodos, número de requisições e número de veículos por período, respectivamente. A sexta coluna “MSC” é a melhor solução conhecida. A sétima coluna é o resultado médio de 10 processamentos do método de CACCHIANI *et al.* [111]. A oitava coluna é o resultado médio de 10 processamentos do método de HEMMELMAYR *et al.* [112]. A nona coluna é o resultado médio de 10 processamentos do método de VIDAL *et al.* [6] que é o método com os melhores resultados médios para as instâncias analisadas atualmente. As três últimas colunas apresentam os resultados do ALNS: o resultado médio de 10 execuções, o melhor valor encontrado e o tempo médio de processamento em minutos. Os valores em negrito representam a melhor solução média encontrada ou a melhor de todas as execuções.

Novamente, na Tabela 4.3 percebe-se que o ALNS apresentou bons resultados em relação à literatura. O método supera os resultados de HEMMELMAYR *et al.* [112] e mostrou-se competitivo em relação aos demais. O ALNS encontrou quatorze melhores valores médios e, em especial, obteve melhor desempenho médio nas instâncias *p04*, *p05*, *p06*, *p07* e *p13* que todos os outros métodos considerados. Em relação às melhores soluções obtidas pelo ALNS, o método foi capaz de encontrar dezoito melhores soluções conhecidas. O tempo computacional variou de pouco mais de vinte segundos até 64,92 minutos na instância *p13*.

O segundo conjunto de instâncias do PRVP (*pr01 – pr10*) foi proposto por CORDEAU *et al.* [5] e se diferencia do primeiro pelo fato de cada cliente possuir um tempo de atendimento distinto e as rotas apresentam limite de tempo.

A Tabela 4.4 apresenta os resultados do ALNS para o segundo grupo de instâncias do PRVP testadas. Esta tabela apresenta a mesma estrutura da Tabela 4.3, mas sem os resultados de CACCHIANI *et al.* [111] pois os autores não testaram tais instâncias.

Nos resultados do ALNS da Tabela 4.4, observa-se que o método produz resultados médios melhores que os de HEMMELMAYR *et al.* [112] e está próximo dos de VIDAL *et al.* [6]. Na instância *pr06* o ALNS apresentou resultados médios superiores aos reportados por HEMMELMAYR *et al.* [112] e VIDAL *et al.* [6]. O tempo computacional variou de um minuto até 115,65 minutos na instância *pr10*.

Com relação ao tempo computacional do ALNS nas instâncias de PRVP, observa-se que ele é similar aos obtidos pelos métodos de VIDAL *et al.* [6] e melhores que os de CACCHIANI *et al.* [111].

### 4.3.1 Instâncias do PRV com Múltiplos Depósitos

Segundo o trabalho de VIDAL *et al.* [6], o PRVMD é um caso particular do PVRP. Esta relação ocorre quando se associa um depósito a um período e cada período possui um nó inicial diferente. Isso torna ambos os modelos matemáticos do PRVP e do PRV com múltiplos depósitos idênticos. Por exemplo, um PRV com múltiplos depósitos com 4 depósitos, pode ser resolvido como se fosse um PRVP com 4 períodos, tal que cada cliente gera uma única requisição, sem restrição do período de atendimento. O conjunto de instâncias utilizado foi o proposto em CORDEAU *et al.* [5] que tem um total de 33 instâncias com 50 até 288 clientes e de 2 a 6 depósitos.

A Tabela 4.5 apresenta os resultados do processamento das instâncias do PRV com múltiplos depósitos testadas. A primeira coluna possui o nome da instância, a segunda, terceira e quarta colunas apresentam o número de clientes, número de depósitos e número de veículos por depósito, respectivamente. A quinta coluna indica o tipo de restrição do problema, sendo “C” somente restrições de carga, e “CT”

Tabela 4.3: Primeiro Grupo de Instâncias do PRVP.

Instância	$n$	$t$	$r$	$d$	MSC	CHT	HDH	VCGLR	ALNS		T (min)
						Média	Média	Média	Média	Melhor	
p01	50	2	50	3	<b>524,61</b>	<b>524,61</b>	<b>524,61</b>	<b>524,61</b>	<b>524,61</b>	<b>524,61</b>	0,59
p02	50	5	104	3	<b>1322,87</b>	1326,16	1332,01	<b>1322,87</b>	1323,39	<b>1322,87</b>	1,46
p03	50	5	50	1	<b>524,61</b>	<b>524,61</b>	528,97	<b>524,61</b>	<b>524,61</b>	<b>524,61</b>	0,58
p04	75	2	75	5	<b>835,26</b>	837,04	847,48	836,59	<b>835,26</b>	<b>835,26</b>	1,35
p05	75	5	153	6	<b>2024,96</b>	2048,29	2059,74	2033,72	<b>2033,04</b>	2027,98	3,42
p06	75	10	75	1	<b>835,26</b>	841,96	884,69	842,48	<b>835,26</b>	<b>835,26</b>	1,35
p07	100	2	100	4	<b>826,14</b>	834,14	829,92	827,02	<b>826,98</b>	<b>826,14</b>	2,21
p08	100	5	202	5	<b>2022,47</b>	2042,02	2058,36	<b>2022,85</b>	2043,02	2028,49	5,48
p09	100	8	100	1	<b>826,14</b>	847,03	834,92	<b>826,94</b>	827,06	<b>826,14</b>	2,20
p10	100	5	174	4	<b>1593,43</b>	1664,90	1629,76	<b>1605,22</b>	1614,24	1598,99	4,52
p11	126	5	191	4	<b>770,89</b>	790,77	791,18	<b>775,84</b>	779,71	777,09	7,73
p12	163	5	185	3	<b>1186,47</b>	1228,14	1258,46	<b>1195,29</b>	1195,58	1191,48	7,79
p13	417	7	457	9	<b>3483,24</b>	3549,30	3835,90	3599,86	<b>3509,86</b>	3490,36	64,92
p14	20	4	40	2	<b>954,81</b>	<b>954,81</b>	<b>954,81</b>	<b>954,81</b>	<b>954,81</b>	<b>954,81</b>	0,33
p15	38	4	72	2	<b>1862,63</b>	<b>1862,63</b>	<b>1862,63</b>	<b>1862,63</b>	<b>1862,63</b>	<b>1862,63</b>	0,77
p16	56	4	104	2	<b>2875,24</b>	<b>2875,24</b>	<b>2875,24</b>	<b>2875,24</b>	<b>2875,24</b>	<b>2875,24</b>	1,44
p17	40	4	80	4	<b>1597,75</b>	<b>1597,75</b>	1601,75	<b>1597,75</b>	<b>1597,75</b>	<b>1597,75</b>	0,95
p18	76	4	144	4	<b>3131,09</b>	3149,09	3147,91	<b>3131,09</b>	3143,99	<b>3131,09</b>	2,27
p19	112	4	208	4	<b>4834,34</b>	4839,87	4851,41	4834,50	4835,85	4834,34	4,50
p20	184	4	336	4	<b>8367,40</b>	<b>8367,40</b>	<b>8367,40</b>	<b>8367,40</b>	<b>8367,40</b>	<b>8367,40</b>	11,22
p21	60	4	120	6	<b>2170,61</b>	<b>2170,61</b>	2180,33	<b>2170,61</b>	<b>2170,61</b>	<b>2170,61</b>	1,59
p22	114	4	216	6	<b>4193,95</b>	4232,50	4218,46	<b>4194,23</b>	4224,41	4195,97	4,61
p23	168	4	312	6	<b>6420,71</b>	6573,33	6644,93	<b>6434,10</b>	6516,42	<b>6420,71</b>	9,61
p24	51	6	90	3	<b>3687,46</b>	<b>3687,46</b>	3704,60	<b>3687,46</b>	3691,92	<b>3687,46</b>	1,12
p25	51	6	90	3	<b>3777,15</b>	<b>3777,15</b>	3781,38	<b>3777,15</b>	<b>3777,15</b>	<b>3777,15</b>	1,16
p26	51	6	90	3	<b>3795,32</b>	<b>3795,32</b>	<b>3795,32</b>	3795,33	3795,33	<b>3795,32</b>	1,10
p27	102	6	180	6	<b>21833,87</b>	21963,23	22153,31	<b>21885,70</b>	21965,40	21935,91	3,77
p28	102	6	180	6	<b>22242,51</b>	<b>22271,91</b>	22418,52	22272,60	22327,59	22296,43	3,53
p29	102	6	180	6	<b>22543,76</b>	22564,80	22864,23	<b>22564,05</b>	22619,76	22591,78	3,47
p30	153	6	270	9	<b>73875,19</b>	75193,09	75579,23	<b>74534,38</b>	75202,28	75039,96	7,67
p31	153	6	270	9	<b>75957,62</b>	<b>76496,67</b>	77459,14	76686,65	77402,80	77247,27	7,64
p32	153	6	270	9	<b>77591,23</b>	<b>78065,24</b>	79487,97	78168,82	78829,38	78587,14	7,57
Médias					<b>11207,03</b>	11296,78	11417,64	<b>11272,89</b>	11344,79	11318,07	

Tabela 4.4: Segundo Grupo de Instâncias do PRVP.

Instância	$n$	$t$	$r$	$d$	MSC	HDH	VCGLR	ALNS		T (min)
						Média	Média	Média	Melhor	
pr01	48	4	96	2	<b>2209,02</b>	2209,11	<b>2209,02</b>	<b>2209,02</b>	<b>2209,02</b>	1,00
pr02	96	4	192	4	<b>3767,50</b>	3787,51	<b>3768,86</b>	3773,89	3770,50	9,13
pr03	144	4	288	6	<b>5153,54</b>	5243,09	<b>5174,80</b>	5192,17	5167,26	20,29
pr04	192	4	384	8	<b>5877,37</b>	6011,39	<b>5936,16</b>	5955,73	5904,58	37,15
pr05	240	4	480	10	<b>6581,86</b>	6778,00	<b>6651,76</b>	6669,07	6634,50	60,71
pr06	288	4	576	12	<b>8207,21</b>	8461,45	8284,94	<b>8246,79</b>	8229,69	88,82
pr07	72	6	216	3	<b>4996,14</b>	5007,01	<b>4996,14</b>	<b>4996,14</b>	<b>4996,14</b>	7,12
pr08	144	6	432	6	<b>6970,68</b>	7119,61	<b>7035,52</b>	7112,53	7095,47	42,31
pr09	216	6	648	9	<b>10038,43</b>	10259,09	<b>10162,22</b>	10205,27	10141,13	72,45
pr10	288	6	864	12	<b>12897,01</b>	13342,41	<b>13091,00</b>	13302,41	13140,29	115,65
Médias					<b>6669,88</b>	6821,87	<b>6731,04</b>	6766,30	6728,86	

restrições de carga e tempo. A coluna “MSC” apresenta a melhor solução conhecida. A sétima coluna é o resultado médio de 10 processamentos do método de VIDAL *et al.* [6] que é o método com os melhores resultados médios para as instâncias de PRV com múltiplos depósitos atualmente. A oitava coluna é o resultado médio de 10 processamentos do método de PISINGER e ROPKE [127] com  $50 \times 10^3$  iterações, que é o ALNS original. As três últimas colunas apresentam os resultados do ALNS: o resultado médio de 10 execuções, o melhor valor encontrado e o tempo médio de processamento em segundos. Os valores em negrito representam a melhor solução média encontrada ou a melhor de todas as execuções.

Os resultados do ALNS da Tabela 4.5 novamente são interessantes. O método foi capaz de encontrar 12 melhores resultados médios, superando os resultados médios de PISINGER e ROPKE [127] e VIDAL *et al.* [6] na instância *p07*. Com relação aos melhores resultados das 10 execuções, o ALNS foi capaz de encontrar 22 melhores soluções conhecidas da literatura.

Com relação ao tempo computacional do ALNS nas instâncias de PRVMD, observa-se que ele é similar aos obtidos pelos métodos de VIDAL *et al.* [6] e maiores que os de PISINGER e ROPKE [127].

## 4.4 Instâncias do PRV com Frota Heterogênea

O primeiro conjunto de instâncias apresentado para o PRV-FH foi introduzido em GOLDEN *et al.* [44] chamado de “VFMP-F”. Esse conjunto consiste em 20 instâncias com 12 até 100 clientes e de 3 até 6 tipos de veículos. Mas como as instâncias 1 – 2 e 7 – 12 possuem somente a matriz de distância entre os clientes e não suas coordenadas, assim muitos autores passaram a utilizar somente as instâncias com coordenadas em seus experimentos. TAILLARD [66] apresentaram as variações “VFMP-V”, “VFMP-FV” e “HVRP” a partir do conjunto de GOLDEN *et al.* [44]. Sendo assim, as instâncias do PRV-FH testadas possuem a seguinte nomenclatura:

- “**HVRP**” - indica que é um problema de frota fixa e sem custo fixo de utilização;
- “**VFMP**” - indica que é um problema de dimensionamento de frota;
- “**F**” - indica que cada tipo de veículo possui somente custo fixo de utilização e custo unitário de deslocamento;
- “**V**” - indica que cada tipo de veículo não possui custo fixo de utilização mas somente custo de deslocamento; e
- “**FV**” - indica que cada tipo de veículo possui ambos os custos de utilização e de deslocamento.

Tabela 4.5: Instâncias do PRV com Múltiplos Depósitos

Instância	$n$	$t$	$d$	Tipo	MSC	VCGLR	PR	ALNS		T(min)
						Média	Média	Média	Melhor	
p01	50	4	4	C	<b>576,87</b>	<b>576,87</b>	<b>576,87</b>	<b>576,87</b>	<b>576,87</b>	0,788
p02	50	4	2	C	<b>473,53</b>	<b>473,53</b>	<b>473,53</b>	<b>473,53</b>	<b>473,53</b>	0,621
p03	75	5	3	C	<b>641,19</b>	<b>641,19</b>	<b>641,19</b>	<b>641,19</b>	<b>641,19</b>	1,485
p04	100	2	8	C	<b>1001,04</b>	<b>1001,04</b>	1006,09	<b>1001,04</b>	<b>1001,04</b>	2,607
p05	100	2	5	C	<b>750,03</b>	<b>750,03</b>	752,34	750,33	<b>750,03</b>	2,291
p06	100	3	6	C	<b>876,5</b>	<b>876,5</b>	883,01	876,50	<b>876,5</b>	2,7
p07	100	4	4	C	<b>881,97</b>	884,43	889,36	<b>881,97</b>	<b>881,97</b>	2,594
p08	249	2	14	CT	<b>4372,78</b>	<b>4397,42</b>	4421,03	4405,71	4383,74	16,817
p09	249	3	12	CT	<b>3858,66</b>	<b>3868,59</b>	3892,5	3893,64	3887,51	17,962
p10	249	4	8	CT	<b>3631,11</b>	<b>3636,08</b>	3666,85	3655,12	3635,71	17,513
p11	249	5	6	CT	<b>3546,06</b>	<b>3548,25</b>	3573,23	3570,05	<b>3546,06</b>	17,31
p12	80	2	5	C	<b>1318,95</b>	<b>1318,95</b>	1319,13	1319,76	<b>1318,95</b>	1,458
p13	80	2	5	CT	<b>1318,95</b>	<b>1318,95</b>	<b>1318,95</b>	<b>1318,95</b>	<b>1318,95</b>	1,492
p14	80	2	5	CT	<b>1360,12</b>	<b>1360,12</b>	<b>1360,12</b>	<b>1360,12</b>	<b>1360,12</b>	1,492
p15	160	4	5	C	<b>2505,42</b>	<b>2505,42</b>	2519,64	2511,54	<b>2505,42</b>	6,32
p16	160	4	5	CT	<b>2572,23</b>	<b>2572,23</b>	2573,95	<b>2572,23</b>	<b>2572,23</b>	6,503
p17	160	4	5	CT	<b>2709,09</b>	<b>2709,09</b>	<b>2709,09</b>	<b>2709,09</b>	<b>2709,09</b>	6,501
p18	240	6	5	C	<b>3702,85</b>	<b>3702,85</b>	3736,53	3731,06	3708,71	15,079
p19	240	6	5	CT	<b>3827,06</b>	<b>3827,06</b>	3838,76	3831,98	<b>3827,06</b>	15,527
p20	240	6	5	CT	<b>4058,07</b>	<b>4058,07</b>	4064,76	<b>4058,07</b>	<b>4058,07</b>	15,497
p21	360	9	5	C	<b>5474,84</b>	<b>5476,41</b>	5501,58	5496,68	5488,33	36,475
p22	360	9	5	CT	<b>5702,16</b>	<b>5702,16</b>	5722,19	5716,91	<b>5702,16</b>	37,651
p23	360	9	5	CT	<b>6078,75</b>	<b>6078,75</b>	6092,66	6090,44	<b>6078,75</b>	37,69
pr01	48	4	1	CT	<b>861,32</b>	<b>861,32</b>	<b>861,32</b>	<b>861,32</b>	<b>861,32</b>	0,526
pr02	96	4	2	CT	<b>1307,34</b>	<b>1307,34</b>	1308,17	1310,32	<b>1307,34</b>	2,07
pr03	144	4	3	CT	<b>1803,8</b>	<b>1803,8</b>	1810,66	1813,06	<b>1803,8</b>	4,932
pr04	192	4	4	CT	<b>2058,31</b>	<b>2059,36</b>	2073,16	2074,44	2069,11	9,205
pr05	240	4	5	CT	<b>2331,2</b>	<b>2340,29</b>	2350,31	2382,64	2344,24	15,027
pr06	288	4	6	CT	<b>2676,3</b>	<b>2681,93</b>	2695,74	2696	2696	22,37
pr07	72	6	1	CT	<b>1089,56</b>	<b>1089,56</b>	<b>1089,56</b>	<b>1089,56</b>	<b>1089,56</b>	1,158
pr08	144	6	2	CT	<b>1664,85</b>	<b>1665,05</b>	1675,74	1675,91	1665,5	4,975
pr09	216	6	3	CT	<b>2133,2</b>	<b>2134,17</b>	2144,84	2154,63	2149,75	12,011
pr10	288	6	4	CT	<b>2868,26</b>	<b>2886,59</b>	2905,43	2903,03	2893,36	22,545
<b>Médias</b>					<b>2425,22</b>	<b>2427,68</b>	2437,83	2436,48	2429,76	



Destaca-se porém, que nenhuma das instâncias de PRVFH testadas possui restrição de tempo de rota.

A Tabela 4.6 apresenta os resultados do processamento das instâncias do PRVFH testadas. A primeira coluna apresenta o nome da instância, a segunda o número de clientes e a terceira o número de tipos de veículos disponíveis. A quarta coluna “MSC” é a melhor solução conhecida. A quinta coluna é o resultado médio de 30 processamentos do método de PENNA *et al.* [82] que é o que detém os melhores resultados médios para as instâncias de PRVFH atualmente. As três últimas colunas apresentam os resultados do ALNS: o resultado médio de 10 execuções, o melhor valor encontrado e o tempo médio de processamento em segundos. Os valores em negrito representam a melhor solução média encontrada ou a melhor de todas as execuções.

Os resultados da Tabela 4.6 mostram que o ALNS apresenta um bom desempenho ao encontrar 16 melhores resultados médios e 26 melhores soluções conhecidas. O ALNS também se mostrou robusto ao encontrar um desvio médio de apenas 0,72%. O tempo computacional permaneceu abaixo de 350 segundos, o que é interessante dada a dificuldade destes problemas.

Com relação ao tempo computacional do ALNS nas instâncias de PRVFH, observa-se que ele é similar aos obtidos pelos métodos de PENNA *et al.* [82].

## 4.5 Instâncias do PRV *Multi-trip*

Como, mencionado na Subseção 2.3.3, não existe um conjunto de instâncias unificado do PRVMT. Desta maneira, utilizou-se as instâncias de PRVFH com múltiplas viagens para cada veículo. Os resultados foram comparados com os melhores resultados do problema original para se verificar a economia (se alguma) pelo uso estendido de cada veículo. As instâncias “VFMP-V” não foram testadas pois são de dimensionamento de frota e não existem custos de utilização, logo não tem como verificar o ganho com a possibilidade de múltiplas viagens.

Foram avaliados dois cenários:

1. **PRVFH-MT2**: Cenário no qual as instâncias do PRVFH são avaliadas com um limite de duas viagens por veículo; e
2. **PRVFH-MT3**: Cenário no qual as instâncias do PRVFH são avaliadas com um limite de três viagens por veículo.

A Tabela 4.7 apresenta os resultados do processamento das instâncias do PRVFH com *Multi-Trips* testadas. A primeira coluna possui o nome da instância, a segunda o número de clientes e a terceira o número de tipos de veículos disponíveis. A

Tabela 4.6: Instâncias do PRVFH.

Instância	$n$	$v$	MSC	PSO	ALNS		
				Média	Média	Melhor	T(s)
HVRP-13	50	6	1517,84	1517,84	1518,81	1517,84	92,03
HVRP-14	50	3	607,53	607,53	607,53	607,53	82,50
HVRP-15	50	3	1015,29	1015,29	1015,29	1015,29	83,76
HVRP-16	50	3	1144,94	1144,94	1144,94	1144,94	83,76
HVRP-17	75	4	1061,96	1061,96	1061,96	1061,96	143,18
HVRP-18	75	6	1823,58	1823,58	1827,24	1823,58	144,50
HVRP-19	100	3	1117,51	1120,30	1128,37	1120,34	221,41
HVRP-20	100	3	1534,17	1534,17	1543,69	1534,17	228,55
VFMP-F-03	20	5	961,03	961,03	961,03	961,03	52,34
VFMP-F-04	20	3	6437,33	6437,33	6437,33	6437,33	52,06
VFMP-F-05	20	5	1007,05	1007,05	1007,05	1007,05	53,10
VFMP-F-06	20	3	6516,47	6516,47	6516,47	6516,47	52,79
VFMP-F-13	50	6	2406,36	2408,53	2415,79	2410,82	146,47
VFMP-F-14	50	3	9119,03	9119,03	9476,11	9119,03	138,06
VFMP-F-15	50	3	2586,37	2586,37	2586,80	2586,37	109,11
VFMP-F-16	50	3	2720,43	2720,43	2754,42	2738,73	140,65
VFMP-F-17	75	4	1734,53	1734,53	1748,95	1734,53	276,68
VFMP-F-18	75	6	2369,65	2371,55	2494,20	2459,84	335,79
VFMP-F-19	100	3	8661,81	8662,68	8665,32	8664,13	325,94
VFMP-F-20	100	3	4029,74	4037,80	4079,70	4068,56	334,81
VFMP-FV-03	20	5	1144,22	1144,22	1145,67	1145,67	53,14
VFMP-FV-04	20	3	6437,33	6437,33	6437,33	6437,33	52,46
VFMP-FV-05	20	5	1322,26	1322,26	1323,02	1323,02	54,13
VFMP-FV-06	20	3	6516,47	6516,47	6516,47	6516,47	52,66
VFMP-FV-13	50	6	2964,65	2964,65	3041,28	3023,51	145,82
VFMP-FV-14	50	3	9126,9	9126,9	9127,00	9126,9	139,17
VFMP-FV-15	50	3	2634,96	2634,96	2672,78	2672,78	139,83
VFMP-FV-16	50	3	3168,92	3168,92	3171,26	3168,95	139,10
VFMP-FV-17	75	4	2004,48	2004,48	2027,92	2014,22	283,09
VFMP-FV-18	75	6	3147,99	3149,56	3224,38	3202,69	338,67
VFMP-FV-19	100	3	8661,81	8661,81	8967,22	8667,16	332,90
VFMP-FV-20	100	3	4153,02	4153,02	4403,90	4382,93	342,59
VFMP-V-03	20	5	623,22	623,22	623,22	623,22	52,27
VFMP-V-04	20	3	387,18	387,18	387,18	387,18	51,91
VFMP-V-05	20	5	742,87	742,87	742,87	742,87	52,89
VFMP-V-06	20	3	415,03	415,03	418,69	418,69	52,18
VFMP-V-13	50	6	1491,86	1491,86	1525,07	1504,08	142,95
VFMP-V-14	50	3	603,21	603,21	604,42	603,21	137,64
VFMP-V-15	50	3	999,82	999,82	1002,59	999,82	136,69
VFMP-V-16	50	3	1131	1131	1132,06	1131,00	137,21
VFMP-V-17	75	4	1038,6	1038,6	1046,24	1041,99	278,63
VFMP-V-18	75	6	1800,8	1800,8	1815,08	1808,79	330,54
VFMP-V-19	100	3	1105,44	1105,44	1111,47	1105,44	334,59
VFMP-V-20	100	3	1530,43	1530,58	1542,98	1532,66	333,39
<b>Médias</b>			<b>2761,93</b>	<b>2762,33</b>	2795,48	2775,23	

quarta coluna “MSC” é a melhor solução conhecida para o PRVFH com somente uma viagem. A quinta coluna é o tempo de solução do ALNS para o cenário de somente uma viagem por veículo. Da sexta coluna em diante são apresentados os resultados do ALNS para os dois cenários. Em cada cenário exibe-se o custo médio, a melhor solução encontrada e o tempo de processamento em segundos.

Observando os resultados das instâncias “HVRP”, eram esperados ganhos modestos, pois são instâncias com frotas limitadas. Assim, foram observados sete idênticas soluções das oito melhores soluções encontradas em ambos os cenários *Multi-trips*, indicando uma limitação do espaço de busca e robustez do ALNS. A maior economia em relação ao cenário sem múltiplas viagens foi de 9,3% na instância HVRP-15. Alguns resultados médios no cenário PRVFH-MT3 foram inferiores ao PRVFH-MT2 indicando uma dificuldade de exploração de um maior espaço de busca. Quanto aos tempos computacionais em relação ao cenário sem múltiplas viagens, em média o cenário PRVFH-MT2 foi 1,17 vezes maior e no cenário PRVFH-MT3 o tempo computacional foi somente 1,33 vezes maior.

Observando os resultados das instâncias “VFMP-F” e “VFMP-FV”, foram observadas economias em relação ao cenário sem múltiplas viagens bem mais expressivas. No cenário PRVFH-MT2 em relação ao cenário sem múltiplas viagens, foi observada uma economia de 33,79% na média das soluções e nas melhores soluções de 35,54%. Foram encontrados os mesmos resultados nas instâncias VFMP-F-03, VFMP-F-15, VFMP-F-16, VFMP-FV-03 e VFMP-FV-04 em todas as execuções do ALNS mostrando a estabilidade do método. Já no cenário PRVFH-MT3 em relação ao cenário sem múltiplas viagens, foi observada uma economia de 43,48% na média das soluções e nas melhores soluções de 47,65%. Foram encontrados os mesmos resultados nas instâncias VFMP-F-03, VFMP-F-05, VFMP-F-16, VFMP-FV-04 e VFMP-FV-06 em todas as execuções do ALNS mostrando a estabilidade do método. Quanto aos tempos computacionais em relação ao cenário sem múltiplas viagens, em média o cenário PRVFH-MT2 foi 1,87 vezes maior e no cenário PRVFH-MT3 o tempo computacional foi 3,10 vezes maior.

Tabela 4.7: Instâncias do PRVFH com Múltiplas Viagens

Instância	$c$	$v$	PRVFH-ST		PRVFH-MT2			PRVFH-MT3		
			MSC	T (s)	Média	Melhor	T (s)	Média	Melhor	T (s)
HVRP-13	50	6	1517,84	92,03	1397,48	<b>1233,25</b>	118,16	1387,48	<b>1233,25</b>	143,77
HVRP-14	50	3	607,53	82,50	562,61	<b>540,14</b>	86,68	570,33	<b>540,14</b>	94,24
HVRP-15	50	3	1015,29	83,76	921,87	<b>878,05</b>	96,28	909,96	<b>878,05</b>	105,63
HVRP-16	50	3	1144,94	83,76	1052,18	1018,28	96,63	1031,05	<b>1003,24</b>	105,09
HVRP-17	75	4	1061,96	143,18	1059,93	<b>1051,46</b>	158,67	1057,19	<b>1051,46</b>	174,23
HVRP-18	75	6	1823,58	144,50	1757,54	<b>1708,23</b>	181,54	1762,92	<b>1708,23</b>	213,36
HVRP-19	100	3	1117,51	221,41	1103,41	<b>1079,12</b>	254,37	1099,75	<b>1079,12</b>	280,19
HVRP-20	100	3	1534,17	228,55	1495,63	<b>1437,26</b>	276,95	1505,99	<b>1437,26</b>	319,02
VFMP-F-03	20	5	961,03	52,34	666,52	666,52	61,69	<b>528,99</b>	<b>528,99</b>	83,87
VFMP-F-04	20	3	6437,33	52,06	3699,72	3437,33	57,69	2731,61	2437,33	78,43
VFMP-F-05	20	5	1007,05	53,10	699,72	688,22	62,84	<b>558,59</b>	<b>558,59</b>	83,56
VFMP-F-06	20	3	6516,47	52,79	3847,63	3516,47	59,30	2821,03	2516,47	78,69
VFMP-F-13	50	6	2406,36	146,47	1498,84	1490,24	286,64	1198,25	1197,55	498,79
VFMP-F-14	50	3	9119,03	138,06	5462,00	5101,37	255,34	4476,21	4062,04	426,11
VFMP-F-15	50	3	2586,37	109,11	1655,79	1655,79	157,12	1389,84	1376,86	244,11
VFMP-F-16	50	3	2720,43	140,65	1710,53	1710,53	250,94	<b>1357,25</b>	<b>1357,25</b>	436,31
VFMP-F-17	75	4	1734,53	276,68	1245,96	1238,20	583,79	1109,01	1089,78	1068,80
VFMP-F-18	75	6	2369,65	335,79	1699,84	1664,33	801,76	1450,48	1367,64	1529,45
VFMP-F-19	100	3	8661,81	325,94	5160,60	5149,46	575,139	5056,65	3979,83	854,87
VFMP-F-20	100	3	4029,74	334,81	2586,12	2578,84	562,07	2219,24	2100,14	851,93
VFMP-FV-03	20	5	1144,22	53,14	890,60	890,60	62,41	801,33	788,57	81,46
VFMP-FV-04	20	3	6437,33	52,46	3437,33	3437,33	59,95	<b>2437,33</b>	<b>2437,33</b>	77,54
VFMP-FV-05	20	5	1322,26	54,13	1068,74	1064,19	65,48	939,21	890,09	83,42
VFMP-FV-06	20	3	6516,47	52,66	3516,48	3516,47	60,78	<b>2516,47</b>	<b>2516,47</b>	78,90
VFMP-FV-13	50	6	2964,65	145,82	2319,90	2314,94	295,61	2075,87	2062,24	502,08
VFMP-FV-14	50	3	9126,90	139,17	5202,59	5083,96	265,27	3944,67	3616,60	426,73
VFMP-FV-15	50	3	2634,96	139,83	1849,03	1834,96	264,02	1615,88	1592,06	425,78
VFMP-FV-16	50	3	3168,92	139,10	2173,49	2169,77	260,34	1840,37	1833,82	421,49
VFMP-FV-17	75	4	2004,48	283,09	1567,31	1546,73	613,70	1431,52	1408,16	1024,85
VFMP-FV-18	75	6	3147,99	338,67	2605,60	2584,63	814,79	2396,00	2361,40	1492,62
VFMP-FV-19	100	3	8661,81	332,90	5569,79	5327,32	587,86	4931,39	3823,62	904,85
VFMP-FV-20	100	3	4153,02	342,59	3115,50	3061,87	580,97	2663,89	2560,61	908,52
<b>Médias</b>			3426,74	131,92	2268,76	2208,62	278,59	1936,75	1793,57	440,58

# Capítulo 5

## Considerações Finais

Neste trabalho foi apresentada uma meta-heurística *Adaptive Large Neighborhood Search* (ALNS) com Mecanismos de Paralelismo, Detecção de Estagnação e Perturbações para o Problema de Roteamento de Veículos com Frota Heterogênea, Periódico e *Multi-trips*.

Novos mecanismos de busca até então não empregados no ALNS foram implementados, sendo eles:

- Uma busca oscilante entre regiões viáveis e inviáveis do espaço de busca;
- Uma memória contendo as melhores soluções viáveis únicas encontradas para possível re-exploração ao longo da busca;
- Paralelismo para melhor proveito dos recursos ociosos das CPUs modernas com múltiplos núcleos de processamento; e
- Detecção de estagnação e perturbações para melhor aplicação da memória e paralelismo empregados.

O ALNS foi testado em instâncias conhecidas da literatura para as variantes do PRV que deram origem ao PRV-FH-P-MT, ou seja, Problema de Roteamento de Veículos Capacitado (PRVC), Problema de Roteamento de Veículos Periódico (PRVP), Problema de Roteamento de Veículos com Frota Heterogênea (PRVFH) e o Problema de Roteamento de Veículos com Múltiplos Depósitos (PRVMD) devido à sua similaridade com o PRVP. Os resultados nestas instâncias da literatura mostraram que o ALNS proposto é competitivo. Ele obteve bons resultados, foi robusto e apresentou tempos computacionais próximos dos métodos utilizados nas comparações realizadas.

Devido à falta de instâncias para o Problema de Roteamento de Veículos com *Multi-trips* (PRVMT), utilizou-se as instâncias do PRVFH modificados para permitir múltiplas viagens. A inserção da característica de múltiplas viagens altera

a dinâmica das soluções do PRVFH já que veículos maiores podem realizar mais viagens sem que o custo fixo, normalmente alto para essa classe, seja considerado mais de uma vez. Os resultados obtidos foram comparados com a situação em que múltiplas viagens não eram permitidas. Com o incremento do número de viagens, destacou-se as reduções no custo e o incremento do tempo computacional.

Com o intuito de permitir aprimoramentos para melhorar os resultados obtidos, sugere-se os seguintes trabalhos futuros:

- Estudar a utilização de métodos de inserção baseados em Programação Linear para refinar o processo de reconstrução da solução. Entretanto, dependendo do número de requisições a serem inseridas, esses métodos podem apresentar elevados tempos computacionais. Diante disso, estudos devem ser realizados para avaliar a performance do ALNS como um todo, bem como estratégias de seleção das heurísticas de inserção que levem em consideração o tempo computacional utilizado. Destaca-se que ROPKE e PISINGER [11] já indicaram que tais métodos poderiam ser utilizados, mas com cautela para não penalizar a execução do ALNS.
- Implementar o mecanismo de paralelismo em outros métodos de otimização. Como exibido na Seção 4.1, quando o número de soluções simultâneas é inferior ao número de núcleos disponíveis na CPU, praticamente não existe incremento de tempo computacional. E mesmo quando o número de soluções simultâneas processadas paralelamente é exatamente igual ao número máximo de *threads* da CPU, o tempo total é de aproximadamente duas vezes o tempo de execução sequencial. Assim, considera-se interessante usar este mecanismo em outros métodos de otimização;
- Implementar a estratégia de detecção de estagnação e perturbações em outros métodos de solução como Busca Tubu, ILS, CS, entre outros métodos de otimização com estratégias de paralelismo. As informações geradas por tais execuções simultâneas podem melhorar o desempenho da busca como um todo, como mostrado pelos experimentos da Seção 4.1; e
- Propor e resolver instâncias para problemas que envolvem o recente tema de veículos com energia alternativa, conforme mencionado na Seção 2.3. Carros elétricos possuem curtos períodos de autonomia, conseqüentemente, se empregados para realizar tarefas de coleta e entrega de produtos, devem retornar ao ponto inicial para realizar recargas, caracterizando um PRVMT. Ao considerar que podem existir mais do que um ponto de recarga, este passa a ser ainda mais complicado e necessita de métodos de solução específicos.

# Referências Bibliográficas

- [1] FRANCIS, P. M., SMILOWITZ, K. R., TZUR, M. “The period vehicle routing problem and its extensions”. In: *The vehicle routing problem: latest advances and new challenges*, Springer, pp. 73–102, 2008.
- [2] GOLDEN, B. L., RAGHAVAN, S., WASIL, E. A. *The vehicle routing problem: latest advances and new challenges*, v. 43. Springer Science & Business Media, 2008.
- [3] TOTH, P., VIGO, D. *Vehicle routing: problems, methods, and applications*, v. 18. Siam, 2014.
- [4] LIU, F.-H., SHEN, S.-Y. “The fleet size and mix vehicle routing problem with time windows”, *Journal of the Operational Research Society*, v. 50, n. 7, pp. 721–732, 1999.
- [5] CORDEAU, J.-F., GENDREAU, M., LAPORTE, G. “A tabu search heuristic for periodic and multi-depot vehicle routing problems”, *Networks*, v. 30, n. 2, pp. 105–119, 1997.
- [6] VIDAL, T., CRAINIC, T. G., GENDREAU, M., et al. “A hybrid genetic algorithm for multidepot and periodic vehicle routing problems”, *Operations Research*, v. 60, n. 3, pp. 611–624, 2012.
- [7] HOFF, A., ANDERSSON, H., CHRISTIANSEN, M., et al. “Industrial aspects and literature survey: Fleet composition and routing”, *Computers & Operations Research*, v. 37, n. 12, pp. 2041–2061, 2010.
- [8] COELHO, V., GRASAS, A., RAMALHINHO, H., et al. “An ILS-based algorithm to solve a large-scale real heterogeneous fleet VRP with multi-trips and docking constraints”, *European Journal of Operational Research*, v. 250, n. 2, pp. 367–376, 2016.
- [9] CORDENONSI, A. Z. “Ambientes, objetos e dialogicidade: uma estratégia de ensino superior em heurísticas e metaheurísticas”, 2008.

- [10] ZANAKIS, S. H., EVANS, J. R., VAZACOPOULOS, A. A. “Heuristic methods and applications: a categorized survey”, *European Journal of Operational Research*, v. 43, n. 1, pp. 88–110, 1989.
- [11] ROPKE, S., PISINGER, D. “A unified heuristic for a large class of vehicle routing problems with backhauls”, *European Journal of Operational Research*, v. 171, n. 3, pp. 750–775, 2006.
- [12] KOÇ, Ç., BEKTAŞ, T., JABALI, O., et al. “Thirty years of heterogeneous vehicle routing”, *European Journal of Operational Research*, v. 249, n. 1, pp. 1–21, 2016.
- [13] BRANDAO, J., MERCER, A. “The multi-trip vehicle routing problem”, *Journal of the Operational research society*, v. 49, n. 8, pp. 799–805, 1998.
- [14] FLORIAN, M., LENSTRA, J. K., RINNOOY KAN, A. “Deterministic production planning: Algorithms and complexity”, *Management science*, v. 26, n. 7, pp. 669–679, 1980.
- [15] RIBEIRO, G. M., DESAULNIERS, G., DESROSIERS, J., et al. “Efficient heuristics for the workover rig routing problem with a heterogeneous fleet and a finite horizon”, *Journal of Heuristics*, v. 20, n. 6, pp. 677–708, 2014.
- [16] DANTZIG, G. B., RAMSER, J. H. “The truck dispatching problem”, *Management science*, v. 6, n. 1, pp. 80–91, 1959.
- [17] CLARKE, G., WRIGHT, J. W. “Scheduling of vehicles from a central depot to a number of delivery points”, *Operations research*, v. 12, n. 4, pp. 568–581, 1964.
- [18] LAPORTE, G., SEMET, F. “Classical heuristics for the capacitated VRP”. In: *The vehicle routing problem*, pp. 109–128. Society for Industrial and Applied Mathematics, 2001.
- [19] PRINS, C., LACOMME, P., PRODHON, C. “Order-first split-second methods for vehicle routing problems: A review”, *Transportation Research Part C: Emerging Technologies*, v. 40, pp. 179–200, 2014.
- [20] CAMPELLO, R. E., MACULAN, N. F. “Lagrangian relaxation for a lower bound to a set partitioning problem with side constraints: properties and algorithms”, *Discrete applied mathematics*, v. 18, n. 2, pp. 119–136, 1987.
- [21] LAPORTE, G. “Modeling and solving several classes of arc routing problems as traveling salesman problems”, *Computers & Operations Research*, v. 24, n. 11, pp. 1057–1061, 1997.



- [22] HWANG, C.-R. “Simulated annealing: theory and applications”, *Acta Applicandae Mathematicae*, v. 12, n. 1, pp. 108–111, 1988.
- [23] ROSE, K. “Deterministic annealing for clustering, compression, classification, regression, and related optimization problems”, *Proceedings of the IEEE*, v. 86, n. 11, pp. 2210–2239, 1998.
- [24] GLOVER, F., LAGUNA, M. *Tabu Search*. Springer, 2013.
- [25] GOLDBERG, D. E., HOLLAND, J. H. “Genetic algorithms and machine learning”, *Machine learning*, v. 3, n. 2, pp. 95–99, 1988.
- [26] EYCKELHOF, C. J., SNOEK, M. “Ant systems for a dynamic TSP”. In: *International Workshop on Ant Algorithms*, pp. 88–99. Springer, 2002.
- [27] FEO, T. A., RESENDE, M. G. “Greedy randomized adaptive search procedures”, *Journal of global optimization*, v. 6, n. 2, pp. 109–133, 1995.
- [28] LOURENÇO, H. R., MARTIN, O. C., STÜTZLE, T. “Iterated local search”. In: *Handbook of metaheuristics*, Springer, pp. 320–353, 2003.
- [29] HERTZ, A., MITTAZ, M. “A variable neighborhood descent algorithm for the undirected capacitated arc routing problem”, *Transportation science*, v. 35, n. 4, pp. 425–434, 2001.
- [30] PISINGER, D., ROPKE, S. “Large neighborhood search”. In: *Handbook of metaheuristics*, Springer, pp. 399–419, 2010.
- [31] CORDEAU, J.-F., LAPORTE, G., MERCIER, A. “A unified tabu search heuristic for vehicle routing problems with time windows”, *Journal of the Operational research society*, v. 52, n. 8, pp. 928–936, 2001.
- [32] MAISCHBERGER, M., CORDEAU, J.-F. “Solving variants of the vehicle routing problem with a simple parallel iterated tabu search”. In: *Network Optimization*, Springer, pp. 395–400, 2011.
- [33] CORDEAU, J.-F., MAISCHBERGER, M. “A parallel iterated tabu search heuristic for vehicle routing problems”, *Computers & Operations Research*, v. 39, n. 9, pp. 2033–2050, 2012.
- [34] ROPKE, S., PISINGER, D. “An adaptive large neighborhood search heuristic for the pickup and delivery problem with time windows”, *Transportation science*, v. 40, n. 4, pp. 455–472, 2006.

- [35] RIBEIRO, G. M., LAPORTE, G. “An adaptive large neighborhood search heuristic for the cumulative capacitated vehicle routing problem”, *Computers & Operations Research*, v. 39, n. 3, pp. 728–735, 2012.
- [36] VIDAL, T., CRAINIC, T. G., GENDREAU, M., et al. “A unified solution framework for multi-attribute vehicle routing problems”, *European Journal of Operational Research*, v. 234, n. 3, pp. 658–673, 2014.
- [37] NEMHAUSER, G. L., WOLSEY, L. A. “Integer programming and combinatorial optimization”, Wiley, Chichester. *GL Nemhauser, MWP Savelsbergh, GS Sigismondi (1992). Constraint Classification for Mixed Integer Programming Formulations. COAL Bulletin*, v. 20, pp. 8–12, 1988.
- [38] YAMAN, H. “Formulations and valid inequalities for the heterogeneous vehicle routing problem”, *Mathematical Programming*, v. 106, n. 2, pp. 365–390, 2006.
- [39] BALDACCI, R., BATTARRA, M., VIGO, D. “Valid inequalities for the fleet size and mix vehicle routing problem with fixed costs”, *Networks*, v. 54, n. 4, pp. 178–189, 2009.
- [40] BALDACCI, R., BARTOLINI, E., MINGOZZI, A., et al. “An exact solution framework for a broad class of vehicle routing problems”, *Computational Management Science*, v. 7, n. 3, pp. 229–268, 2010.
- [41] DESAULNIERS, G., DESROSIERS, J., SOLOMON, M. M. *Column generation*, v. 5. Springer Science & Business Media, 2006.
- [42] DESROSIERS, J., LÜBBECKE, M. E. “Branch-Price-and-Cut Algorithms”, *Wiley encyclopedia of operations research and management science*, 2011.
- [43] FEILLET, D. “A tutorial on column generation and branch-and-price for vehicle routing problems”, *4or*, v. 8, n. 4, pp. 407–424, 2010.
- [44] GOLDEN, B., ASSAD, A., LEVY, L., et al. “The fleet size and mix vehicle routing problem”, *Computers & Operations Research*, v. 11, n. 1, pp. 49–66, 1984.
- [45] MILLER, C. E., TUCKER, A. W., ZEMLIN, R. A. “Integer programming formulation of traveling salesman problems”, *Journal of the ACM (JACM)*, v. 7, n. 4, pp. 326–329, 1960.
- [46] PESSOA, A., UCHOA, E., POGGI DE ARAGÃO, M. “A robust branch-cut-and-price algorithm for the heterogeneous fleet vehicle routing problem”, *Networks*, v. 54, n. 4, pp. 167–177, 2009.

- [47] OSMAN, I. H., SALHI, S. “Local search strategies for the vehicle fleet mix problem”, *Modern heuristic search methods*, pp. 131–153, 1996.
- [48] SALHI, S., RAND, G. K. “Incorporating vehicle routing into the vehicle fleet composition problem”, *European Journal of Operational Research*, v. 66, n. 3, pp. 313–330, 1993.
- [49] GENDREAU, M., LAPORTE, G., MUSARAGANYI, C., et al. “A tabu search heuristic for the heterogeneous fleet vehicle routing problem”, *Computers & Operations Research*, v. 26, n. 12, pp. 1153–1173, 1999.
- [50] GENDREAU, M., HERTZ, A., LAPORTE, G. “New insertion and postoptimization procedures for the traveling salesman problem”, *Operations Research*, v. 40, n. 6, pp. 1086–1094, 1992.
- [51] ROCHAT, Y., TAILLARD, É. D. “Probabilistic diversification and intensification in local search for vehicle routing”, *Journal of heuristics*, v. 1, n. 1, pp. 147–167, 1995.
- [52] WASSAN, N. A., OSMAN, I. H. “Tabu search variants for the mix fleet vehicle routing problem”, *Journal of the Operational Research Society*, v. 53, n. 7, pp. 768–782, 2002.
- [53] LEE, Y., KIM, J., KANG, K., et al. “A heuristic for vehicle fleet mix problem using tabu search and set partitioning”, *Journal of the Operational Research Society*, v. 59, n. 6, pp. 833–841, 2008.
- [54] PRINS, C. “A simple and effective evolutionary algorithm for the vehicle routing problem”, *Computers & Operations Research*, v. 31, n. 12, pp. 1985–2002, 2004.
- [55] BRANDÃO, J. “A deterministic tabu search algorithm for the fleet size and mix vehicle routing problem”, *European journal of operational research*, v. 195, n. 3, pp. 716–728, 2009.
- [56] OR, I., PIERSKALLA, W. P. *BTAP: A computer program to obtain solutions to the blood transportation-allocation problem and other traveling salesman type problems*. Relatório técnico, Technical Report, Department of Industrial Engineering and Management Sciences, Northwestern University, Evanston, Illinois (August 1976), 1976.
- [57] FISHER, M. L., JAIKUMAR, R. “A generalized assignment heuristic for vehicle routing”, *Networks*, v. 11, n. 2, pp. 109–124, 1981.

- [58] GHEYSENS, F., GOLDEN, B., ASSAD, A. “A comparison of techniques for solving the fleet size and mix vehicle routing problem”, *Operations-Research-Spektrum*, v. 6, n. 4, pp. 207–216, 1984.
- [59] GHEYSENS, F., GOLDEN, B., ASSAD, A. “A new heuristic for determining fleet size and composition”. In: *Netflow at Pisa*, Springer, pp. 233–236, 1986.
- [60] DESROCHERS, M., VERHOOG, T. “A new heuristic for the fleet size and mix vehicle routing problem”, *Computers & Operations Research*, v. 18, n. 3, pp. 263–274, 1991.
- [61] SALHI, S., SARI, M., SAIDI, D., et al. “Adaptation of some vehicle fleet mix heuristics”, *Omega*, v. 20, n. 5, pp. 653–660, 1992.
- [62] RENAUD, J., BOCTOR, F. F. “A sweep-based algorithm for the fleet size and mix vehicle routing problem”, *European Journal of Operational Research*, v. 140, n. 3, pp. 618–628, 2002.
- [63] RENAUD, J., BOCTOR, F. F., LAPORTE, G. “An improved petal heuristic for the vehicle routing problem”, *Journal of the Operational Research Society*, v. 47, n. 2, pp. 329–336, 1996.
- [64] HAN, A. F.-W., CHO, Y.-J. “A GIDS metaheuristic approach to the fleet size and mix vehicle routing problem”. In: *Essays and surveys in metaheuristics*, Springer, pp. 399–413, 2002.
- [65] DUECK, G. “New optimization heuristics: The great deluge algorithm and the record-to-record travel”, *Journal of Computational physics*, v. 104, n. 1, pp. 86–92, 1993.
- [66] TAILLARD, É. D. “A heuristic column generation method for the heterogeneous fleet VRP”, *RAIRO-Operations Research*, v. 33, n. 1, pp. 1–14, 1999.
- [67] TARANTILIS, C. D., KIRANOUDIS, C. T., VASSILIADIS, V. S. “A threshold accepting metaheuristic for the heterogeneous fixed fleet vehicle routing problem”, *European Journal of Operational Research*, v. 152, n. 1, pp. 148–158, 2004.
- [68] TARANTILIS, C., KIRANOUDIS, C. “A meta-heuristic algorithm for the efficient distribution of perishable foods”, *Journal of food Engineering*, v. 50, n. 1, pp. 1–9, 2001.

- [69] GENCER, C., TOP, İ., AYDOGAN, E. K. “A new intuitional algorithm for solving heterogeneous fixed fleet routing problems: Passenger pickup algorithm”, *Applied Mathematics and Computation*, v. 181, n. 2, pp. 1552–1567, 2006.
- [70] LI, F., GOLDEN, B., WASIL, E. “A record-to-record travel algorithm for solving the heterogeneous fleet vehicle routing problem”, *Computers & Operations Research*, v. 34, n. 9, pp. 2734–2742, 2007.
- [71] LI, F., GOLDEN, B., WASIL, E. “Very large-scale vehicle routing: new test problems, algorithms, and results”, *Computers & Operations Research*, v. 32, n. 5, pp. 1165–1179, 2005.
- [72] EUCHI, J., CHABCHOUB, H. “A hybrid tabu search to solve the heterogeneous fixed fleet vehicle routing problem”, *Logistics Research*, v. 2, n. 1, pp. 3–11, 2010.
- [73] BRANDÃO, J. “A tabu search algorithm for the heterogeneous fixed fleet vehicle routing problem”, *Computers & Operations Research*, v. 38, n. 1, pp. 140–151, 2011.
- [74] LI, X., TIAN, P., ANEJA, Y. “An adaptive memory programming metaheuristic for the heterogeneous fixed fleet vehicle routing problem”, *Transportation Research Part E: Logistics and Transportation Review*, v. 46, n. 6, pp. 1111–1127, 2010.
- [75] LIU, S. “A hybrid population heuristic for the heterogeneous vehicle routing problems”, *Transportation Research Part E: Logistics and Transportation Review*, v. 54, pp. 67–78, 2013.
- [76] NAJI-AZIMI, Z., SALARI, M. “A complementary tool to enhance the effectiveness of existing methods for heterogeneous fixed fleet vehicle routing problem”, *Applied Mathematical Modelling*, v. 37, n. 6, pp. 4316–4324, 2013.
- [77] CHOI, E., TCHA, D.-W. “A column generation approach to the heterogeneous fleet vehicle routing problem”, *Computers & Operations Research*, v. 34, n. 7, pp. 2080–2095, 2007.
- [78] BALDACCI, R., MINGOZZI, A. “A unified exact method for solving different classes of vehicle routing problems”, *Mathematical Programming*, v. 120, n. 2, pp. 347–380, 2009.

- [79] IMRAN, A., SALHI, S., WASSAN, N. A. “A variable neighborhood-based heuristic for the heterogeneous fleet vehicle routing problem”, *European Journal of Operational Research*, v. 197, n. 2, pp. 509–518, 2009.
- [80] KNUTH, D. E. *The state of the Art of Computer Programming*. Relatório técnico, DTIC Document, 1976.
- [81] SUBRAMANIAN, A., PENNA, P. H. V., UCHOA, E., et al. “A hybrid algorithm for the heterogeneous fleet vehicle routing problem”, *European Journal of Operational Research*, v. 221, n. 2, pp. 285–295, 2012.
- [82] PENNA, P. H. V., SUBRAMANIAN, A., OCHI, L. S. “An iterated local search heuristic for the heterogeneous fleet vehicle routing problem”, *Journal of Heuristics*, v. 19, n. 2, pp. 201–232, 2013.
- [83] ARENALES, M., ARMENTANO, V., OTHERS. *Pesquisa Operacional, 2ª Edição: Para Cursos de Engenharia*. Elsevier Brasil, 2015.
- [84] BELTRAMI, E. J., BODIN, L. D. “Networks and vehicle routing for municipal waste collection”, *Networks*, v. 4, n. 1, pp. 65–94, 1974.
- [85] FOSTER, B. A., RYAN, D. M. “An integer programming approach to the vehicle scheduling problem”, *Journal of the Operational Research Society*, v. 27, n. 2, pp. 367–384, 1976.
- [86] RUSSELL, R., IGO, W. “An assignment routing problem”, *Networks*, v. 9, n. 1, pp. 1–17, 1979.
- [87] CHRISTOFIDES, N., BEASLEY, J. E. “The period routing problem”, *Networks*, v. 14, n. 2, pp. 237–256, 1984.
- [88] RUSSELL, R. A., GRIBBIN, D. “A multiphase approach to the period routing problem”, *Networks*, v. 21, n. 7, pp. 747–765, 1991.
- [89] TAN, C., BEASLEY, J. “A heuristic algorithm for the period vehicle routing problem”, *Omega*, v. 12, n. 5, pp. 497–504, 1984.
- [90] GAUDIOSO, M., PALETTA, G. “A heuristic for the periodic vehicle routing problem”, *Transportation Science*, v. 26, n. 2, pp. 86–92, 1992.
- [91] CHAO, I., GOLDEN, B. L., WASIL, E., et al. “An improved heuristic for the period vehicle routing problem”, *Networks*, v. 26, n. 1, pp. 25–44, 1995.
- [92] DRUMMOND, L. M., OCHI, L. S., VIANNA, D. S. “An asynchronous parallel metaheuristic for the period vehicle routing problem”, *Future generation computer systems*, v. 17, n. 4, pp. 379–386, 2001.

- [93] FRANCIS, P., SMILOWITZ, K. “Modeling techniques for periodic vehicle routing problems”, *Transportation Research Part B: Methodological*, v. 40, n. 10, pp. 872–884, 2006.
- [94] FRANCIS, P., SMILOWITZ, K., TZUR, M. “The period vehicle routing problem with service choice”, *Transportation Science*, v. 40, n. 4, pp. 439–454, 2006.
- [95] MOURGAYA, M., VANDERBECK, F. “Column generation based heuristic for tactical planning in multi-period vehicle routing”, *European Journal of Operational Research*, v. 183, n. 3, pp. 1028–1041, 2007.
- [96] ANGELELLI, E., SPERANZA, M. G. “The application of a vehicle routing model to a waste collection problem: two case studies”. In: *Quantitative Approaches to Distribution Logistics and Supply Chain Management*, Springer, pp. 269–286, 2002.
- [97] NUORTIO, T., KYTÖJOKI, J., NISKA, H., et al. “Improved route planning and scheduling of waste collection and transport”, *Expert systems with applications*, v. 30, n. 2, pp. 223–232, 2006.
- [98] COENE, S., ARNOUT, A., SPIEKSMAN, F. C. “On a periodic vehicle routing problem”, *Journal of the Operational Research Society*, v. 61, n. 12, pp. 1719–1728, 2010.
- [99] ALEGRE, J., LAGUNA, M., PACHECO, J. “Optimizing the periodic pick-up of raw materials for a manufacturer of auto parts”, *European Journal of Operational Research*, v. 179, n. 3, pp. 736–746, 2007.
- [100] CLAASSEN, G., HENDRIKS, T. H. “An application of special ordered sets to a periodic milk collection problem”, *European Journal of Operational Research*, v. 180, n. 2, pp. 754–769, 2007.
- [101] CHRISTIANSEN, M., FAGERHOLT, K., NYGREEN, B., et al. “Maritime transportation”, *Handbooks in operations research and management science*, v. 14, pp. 189–284, 2007.
- [102] HADJICONSTANTINO, E., BALDACCI, R. “A multi-depot period vehicle routing problem arising in the utilities sector”, *Journal of the Operational Research Society*, v. 49, n. 12, pp. 1239–1248, 1998.
- [103] BLAKELEY, F., ARGÜELLO, B., CAO, B., et al. “Optimizing periodic maintenance operations for Schindler Elevator Corporation”, *Interfaces*, v. 33, n. 1, pp. 67–79, 2003.

- [104] HEMMELMAYR, V., DOERNER, K. F., HARTL, R. F., et al. “Delivery strategies for blood products supplies”, *OR spectrum*, v. 31, n. 4, pp. 707–725, 2009.
- [105] PACHECO, J., ALVAREZ, A., GARCÍA, I., et al. “Optimizing vehicle routes in a bakery company allowing flexibility in delivery dates”, *Journal of the Operational Research Society*, v. 63, n. 5, pp. 569–581, 2012.
- [106] SHAO, Y., BARD, J. F., JARRAH, A. I. “The therapist routing and scheduling problem”, *Iie Transactions*, v. 44, n. 10, pp. 868–893, 2012.
- [107] MAYA, P., SÖRENSEN, K., GOOS, P. “A metaheuristic for a teaching assistant assignment-routing problem”, *Computers & Operations Research*, v. 39, n. 2, pp. 249–258, 2012.
- [108] BALDACCI, R., BARTOLINI, E., MINGOZZI, A., et al. “An exact algorithm for the period routing problem”, *Operations research*, v. 59, n. 1, pp. 228–241, 2011.
- [109] GULCZYNSKI, D., GOLDEN, B., WASIL, E. “The period vehicle routing problem: New heuristics and real-world variants”, *Transportation Research Part E: Logistics and Transportation Review*, v. 47, n. 5, pp. 648–668, 2011.
- [110] GROËR, C., GOLDEN, B., WASIL, E. “A library of local search heuristics for the vehicle routing problem”, *Mathematical Programming Computation*, v. 2, n. 2, pp. 79–101, 2010.
- [111] CACCHIANI, V., HEMMELMAYR, V. C., TRICOIRE, F. “A set-covering based heuristic algorithm for the periodic vehicle routing problem”, *Discrete Applied Mathematics*, v. 163, pp. 53–64, 2014.
- [112] HEMMELMAYR, V. C., DOERNER, K. F., HARTL, R. F. “A variable neighborhood search heuristic for periodic routing problems”, *European Journal of Operational Research*, v. 195, n. 3, pp. 791–802, 2009.
- [113] LIN, S. “Computer solutions of the traveling salesman problem”, *The Bell System Technical Journal*, v. 44, n. 10, pp. 2245–2269, 1965.
- [114] PIRKWIESER, S., RAIDL, G. R. “Multilevel variable neighborhood search for periodic routing problems”. In: *European Conference on Evolutionary Computation in Combinatorial Optimization*, pp. 226–238. Springer, 2010.



- [115] PETCH, R. J., SALHI, S. “A multi-phase constructive heuristic for the vehicle routing problem with multiple trips”, *Discrete Applied Mathematics*, v. 133, n. 1, pp. 69–92, 2003.
- [116] SALHI, S. *The integration of routing into the location-allocation and vehicle fleet composition problems*. Tese de Doutorado, Université Abderrahmane Mira de Béjaia, 1987.
- [117] FLEISCHMANN, B. “The discrete lot-sizing and scheduling problem”, *European Journal of Operational Research*, v. 44, n. 3, pp. 337–348, 1990.
- [118] MARTELLO, S., TOTH, P. “Algorithms for knapsack problems”, *North-Holland Mathematics Studies*, v. 132, pp. 213–257, 1987.
- [119] TAILLARD, É. D., LAPORTE, G., GENDREAU, M. “Vehicle routing with multiple use of vehicles”, *Journal of the Operational research society*, v. 47, n. 8, pp. 1065–1070, 1996.
- [120] BRANDAO, J., MERCER, A. “A tabu search algorithm for the multi-trip vehicle routing and scheduling problem”, *European journal of operational research*, v. 100, n. 1, pp. 180–191, 1997.
- [121] OLIVERA, A., VIERA, O. “Adaptive memory programming for the vehicle routing problem with multiple trips”, *Computers & Operations Research*, v. 34, n. 1, pp. 28–47, 2007.
- [122] MINGOZZI, A., ROBERTI, R., TOTH, P. “An exact algorithm for the multi-trip vehicle routing problem”, *INFORMS Journal on Computing*, v. 25, n. 2, pp. 193–207, 2013.
- [123] TANG, J., YU, Y., LI, J. “An exact algorithm for the multi-trip vehicle routing and scheduling problem of pickup and delivery of customers to the airport”, *Transportation Research Part E: Logistics and Transportation Review*, v. 73, pp. 114–132, 2015.
- [124] ERDOĞAN, S., MILLER-HOOKS, E. “A green vehicle routing problem”, *Transportation Research Part E: Logistics and Transportation Review*, v. 48, n. 1, pp. 100–114, 2012.
- [125] SCHNEIDER, M., STENGER, A., GOEKE, D. “The electric vehicle-routing problem with time windows and recharging stations”, *Transportation Science*, v. 48, n. 4, pp. 500–520, 2014.

- [126] SHAW, P. “Using constraint programming and local search methods to solve vehicle routing problems”. In: *International Conference on Principles and Practice of Constraint Programming*, pp. 417–431. Springer, 1998.
- [127] PISINGER, D., ROPKE, S. “A general heuristic for vehicle routing problems”, *Computers & operations research*, v. 34, n. 8, pp. 2403–2435, 2007.
- [128] DEMIR, E., BEKTAŞ, T., LAPORTE, G. “An adaptive large neighborhood search heuristic for the pollution-routing problem”, *European Journal of Operational Research*, v. 223, n. 2, pp. 346–359, 2012.
- [129] ADULYASAK, Y., CORDEAU, J.-F., JANS, R. “Optimization-based adaptive large neighborhood search for the production routing problem”, *Transportation Science*, v. 48, n. 1, pp. 20–45, 2012.
- [130] MASSON, R., LEHUÉDÉ, F., PÉTON, O. “An adaptive large neighborhood search for the pickup and delivery problem with transfers”, *Transportation Science*, v. 47, n. 3, pp. 344–355, 2013.
- [131] RIBEIRO, G. M., DESAULNIERS, G., DESROSIERS, J. “A branch-price-and-cut algorithm for the workover rig routing problem”, *Computers & Operations Research*, v. 39, n. 12, pp. 3305–3315, 2012.
- [132] GLOVER, F., HAO, J.-K. “The case for strategic oscillation”, *Annals of Operations Research*, v. 183, n. 1, pp. 163–173, 2011.
- [133] GENDREAU, M., HERTZ, A., LAPORTE, G. “A tabu search heuristic for the vehicle routing problem”, *Management science*, v. 40, n. 10, pp. 1276–1290, 1994.
- [134] SHAW, P. “A new local search algorithm providing high quality solutions to vehicle routing problems”, *APES Group, Dept of Computer Science, University of Strathclyde, Glasgow, Scotland, UK*, 1997.
- [135] RIBEIRO, G. M., LAPORTE, G., MAURI, G. R. “A comparison of three metaheuristics for the workover rig routing problem”, *European Journal of Operational Research*, v. 220, n. 1, pp. 28–36, 2012.
- [136] CHRISTOFIDES, N., MINGOZZI, A., TOTH, P. “Exact algorithms for the vehicle routing problem, based on spanning tree and shortest path relaxations”, *Mathematical programming*, v. 20, n. 1, pp. 255–282, 1981.
- [137] NAGATA, Y., BRÄYSY, O. “A powerful route minimization heuristic for the vehicle routing problem with time windows”, *Operations Research Letters*, v. 37, n. 5, pp. 333–338, 2009.

- [138] EILON, S., WATSON-GANDY, C., CHRISTOFIDES, N. *Distribution management*. Griffin London, 1971.