

**UNIVERSIDADE DO ESTADO DE SANTA CATARINA - UDESC  
CENTRO DE CIÊNCIAS TECNOLÓGICAS - CCT  
MESTRADO EM COMPUTAÇÃO APLICADA**

**VÍTOR GAUER PEREIRA**

**PROPOSTA DE ALGORITMO DE BUSCA LARGA ADAPTATIVA EM  
VIZINHANÇA PARA O PROBLEMA DE ROTEAMENTO DE  
VEÍCULOS COM FROTA HETEROGRÊNEA LIMITADA, JANELA DE  
TEMPO E CUSTO DE CARGA**

**JOINVILLE**

**2023**

**VÍTOR GAUER PEREIRA**

**PROPOSTA DE ALGORITMO DE BUSCA LARGA ADAPTATIVA EM  
VIZINHANÇA PARA O PROBLEMA DE ROTEAMENTO DE  
VEÍCULOS COM FROTA HETEROGÊNEA LIMITADA, JANELA DE  
TEMPO E CUSTO DE CARGA**

Dissertação apresentada ao Programa de  
Pós-Graduação em Computação Aplicada  
do Centro de Ciências Tecnológicas da  
Universidade do Estado de Santa Catarina,  
como requisito parcial para a obtenção do  
grau de Mestre em Computação Aplicada.

Orientador: Dr. Fabiano Baldo  
Coorientador: Dr. Omir C. Alves Junior

**JOINVILLE**

**2023**

**Ficha catalográfica elaborada pelo programa de geração automática da  
Biblioteca Setorial do CCT/UDESC,  
com os dados fornecidos pelo(a) autor(a)**

Pereira, Vítor Gauer

Proposta de algoritmo de busca larga adaptativa em vizinhança para o problema de roteamento de veículos com frota heterogênea limitada, janela de tempo e custo de carga / Vítor Gauer Pereira. -- 2023.

130 p.

Orientador: Fabiano Baldo

Coorientador: Omir Correia Alves Junior

Dissertação (mestrado) -- Universidade do Estado de Santa Catarina, Centro de Ciências Tecnológicas, Programa de Pós-Graduação em Computação Aplicada, Joinville, 2023.

1. Problema de roteamento de veículos. 2. Frota heterogênea limitada. 3. Janela de tempo. 4. Custo de carga. 5. Busca larga adaptativa em vizinhança. I. Baldo, Fabiano. II. Junior, Omir Correia Alves. III. Universidade do Estado de Santa Catarina, Centro de Ciências Tecnológicas, Programa de Pós-Graduação em Computação Aplicada. IV. Titulo.

**VÍTOR GAUER PEREIRA**

**PROPOSTA DE ALGORITMO DE BUSCA LARGA ADAPTATIVA EM VIZINHANÇA  
PARA O PROBLEMA DE ROTEAMENTO DE VEÍCULOS COM FROTA  
HETEROGÊNEA LIMITADA, JANELA DE TEMPO E CUSTO DE CARGA**

Dissertação apresentada ao Programa de Pós-Graduação em Computação Aplicada do Centro de Ciências Tecnológicas da Universidade do Estado de Santa Catarina, como requisito parcial para a obtenção do grau de Mestre em Computação Aplicada.

Orientador: Dr. Fabiano Baldo

Coorientador: Dr. Omir C. Alves Junior

**BANCA EXAMINADORA:**

Dr. Fabiano Baldo

CCT/UDESC

Membros:

Dr. Omir C. Alves Junior

CCT/UDESC

Dr. Geraldo Regis Mauri

Departamento de Computação/UFES

Dr. Rafael Stubs Parpinelli

CCT/UDESC

Joinville, 18 de abril de 2023

Dedico este trabalho de pesquisa aos meus pais, irmãos, namorada e amigos, pelo inteiro apoio e paciência durante todo o curso para a produção desta monografia.

## **AGRADECIMENTOS**

Agradeço aos meus pais, Vera e Luiz Antônio, meus irmãos Lucas e Luana, minha namorada Nathália, e toda a família pelo apoio incondicional ao longo desta jornada.

Ao meu orientador Fabiano Baldo e coorientador Omir Alves, pelo suporte, incentivo e discussões enriquecedoras para a produção deste trabalho em sua melhor versão possível.

Aos meus colegas e amigos de universidade, pela parceria, por estarem presentes em momentos de alegria e desafios acadêmicos, por estarem sempre disponíveis para ajudar.

Agradeço aos professores da UDESC, pelos aprendizados, desafios impostos ao longo do curso e ensinamentos valorosos para o restante da minha vida acadêmica e pessoal.

Por fim, agradeço a todos que aplicaram tempo e paciência para a leitura e apreciação deste trabalho.

"A inovação distingue um líder de um seguidor." (Steve Jobs)

## RESUMO

Na economia atual, as operações logísticas de transporte se tornaram cada vez mais relevantes devido à necessidade de otimizar processos que refletem em benefícios e diminuição de gastos para a organização, gerando efeitos no restante da cadeia produtiva. Operações de transporte variam conforme a frota de veículos, características dos clientes e demais restrições como o impacto do peso das cargas transportadas no custo da operação. Os Problemas de Roteamento de Veículos (VRP) representam esta área de logística de transporte e suas diversas características. Em geral, companhias de transporte possuem uma frota com modelos de veículos variados em termos de custos e capacidades, e um limite fixo de quantidade de veículos. Estas características estão relacionadas ao Problema de Roteamento de Veículos com Frota Heterogênea Limitada (HFVRP). Os problemas de VRP clássicos negligenciam o impacto da carga transportada no custo total da operação, pois consideram a carga constante ao longo da rota. Para incluir a noção do custo de carga, o Problema de Roteamento de Veículos com Custo de Carga (VRPLC) foi introduzido, de modo a considerar a carga como variável e não constante. O Problema de Roteamento de Veículo com Janela de Tempo (VRPTW) adiciona um período de tempo disponível para atendimento dos clientes, o que representa as situações reais de atendimentos com hora marcada. Este trabalho propõe a junção das características de janela de tempo, frota heterogênea limitada e custo de carga no chamado Problema de Roteamento de Veículos com Frota Heterogênea Limitada com Janela de Tempo e Custo de Carga (HFVRPTW-LC). Para a solução deste problema, este trabalho propõe um método de otimização baseado na meta-heurística ALNS (busca larga adaptativa em vizinhança). Os experimentos foram realizados em duas etapas. Na primeira, foi testada a aplicação do método de solução proposto ao HFVRPTW, sem custo de carga. Neste experimento foi possível observar, com significância estatística, uma redução média de 1,48% no custo das soluções em comparação com um trabalho presente na literatura, em que 51 de 56 instâncias do *benchmark* foram melhoradas. Já a segunda, o método de solução foi aplicado ao HFVRPTW-LC. Neste caso, foi possível constatar que a utilização do custo de carga representou uma redução média de 14,04% no custo das soluções, com significância estatística, sem comprometer as ocupações dos veículos, com 48 de 56 instâncias melhoradas. Portanto, o uso de custo de carga gerou, em média, impacto na redução de custo total das rotas.

**Palavras-chaves:** problema de roteamento de veículos, frota heterogênea limitada, janela de tempo, custo de carga, busca larga adaptativa em vizinhança, busca local

## ABSTRACT

In the current economy, transport logistics operations have become increasingly relevant due to the need of optimize process that reflects on benefits and cost reduces for the organization, generating effects on the rest of the productive chain. Transport operations vary on fleet of vehicles, customers particularities and other constraints such as the transported weights impact on the total costs. The Vehicle Routing Problems (VRP) represents this transport logistic area e its features diversity. In general, transport companies owns a predefined fleet of heterogeneous vehicles related to capacity and costs, and a limited number of vehicles. These features are related to the Heterogeneous Fixed Fleet Vehicle Routing Problem (HFVRP). Classics VRPs, neglects the impact of transported loads on the operation overall cost, since takes load weights as constant. In order to include the loading cost, the Vehicle Routing Problem with Loading Cost (VRPLC) was defined, which takes the loads as variable and not as a constant. The Vehicle Routing Problem with Time Window (VRPTW) considers a period of time to start to service the customers, which represents an appointment situations. This work proposes to join the following features, time window, heterogeneous fixed fleet and loading cost, on denominated Heterogeneous Fixed Fleet Vehicle Routing Problem with Time Window and Loading Cost (HFVRPTW-LC). To solve the problem, this work proposes a metaheuristic-based method ALNS (adaptive large neighborhood search). Experiments are made in two steps: the first one for the HFVRPTW problem without loading cost, which the outcomes shows a cost reduction of 1.48% on average, with statistical significance, and 51 from 56 instances of the benchmark improved. For the second step, for HFVRPTW-LC problem, the outcomes shows that using loading costs leads to a average reduce of 14.04% on overall costs with statistical significance, without compromising vehicles occupation, and 48 from 56 improved instances. Therefore, the use of loading costs represents an impact on reducing costs on average.

**Key-words:** vehicle routing problem, heterogeneous fixed fleet, time window, loading cost, adaptive large neighborhood search, local search.

## LISTA DE ILUSTRAÇÕES

Figura 1 – Representação de Roteamento de Veículos (VRP) . . . . .	23
Figura 2 – Configuração do Roteamento de Veículos Capacitado (CVRP) . . . . .	30
Figura 3 – Configuração do Roteamento de Veículos com Janela de Tempo (VRPTW) . . . . .	31
Figura 4 – Configuração do Roteamento de Veículos com Frota Heterogênea (HVRP) . . . . .	33
Figura 5 – Exemplo do Roteamento de Veículos com Custo de Carga (VRPLC) . . . . .	37
Figura 6 – Operador de inserção . . . . .	41
Figura 7 – Operador de remoção . . . . .	43
Figura 8 – Comportamento de uma heurística de refinamento . . . . .	45
Figura 9 – Comportamento de operações de ruído . . . . .	47
Figura 10 – Representação de uma solução do método proposto . . . . .	65
Figura 11 – Método proposto para solução do <i>Heterogeneous Fixed Fleet Vehicle Routing Problem with Time Window and Loading Cost</i> (HFVRPTW-LC) com ALNS . . . . .	68
Figura 12 – Operador <i>Relocate</i> . . . . .	86
Figura 13 – Operador <i>Exchange</i> . . . . .	86
Figura 14 – Operador <i>Cross</i> . . . . .	86
Figura 15 – Distribuição dos clientes . . . . .	93
Figura 16 – Diagrama de Caixa RPD (%) . . . . .	99
Figura 17 – Gráficos convergência e de pesos dos operadores de remoção e inserção (HC101) . . . . .	101
Figura 18 – Gráficos convergência e de pesos dos operadores de remoção e inserção (HC207) . . . . .	102
Figura 19 – Gráficos convergência e de pesos dos operadores de remoção e inserção (HR107) . . . . .	103
Figura 20 – Gráficos convergência e de pesos dos operadores de remoção e inserção (HR208) . . . . .	104
Figura 21 – Gráficos convergência e de pesos dos operadores de remoção e inserção (HRC101) . . . . .	105
Figura 22 – Gráficos convergência e de pesos dos operadores de remoção e inserção (HRC205) . . . . .	106
Figura 23 – Exemplos de análise e seleção dos valores de $w$ . . . . .	110
Figura 24 – Diagrama de Caixa $RPD^{LC}$ (%) para HFVRPTW-LC . . . . .	115

## LISTA DE TABELAS

Tabela 1 – Taxonomia para o Problema de Roteamento de Veículo ( <i>Vehicle Routing Problem</i> (VRP)) . . . . .	25
Tabela 2 – Trabalhos Relacionados . . . . .	60
Tabela 3 – Categorias das instâncias de Solomon . . . . .	93
Tabela 4 – Formatação do arquivo de entrada . . . . .	93
Tabela 5 – Tabela de Tipos de Veículos (Frota Heterogênea) . . . . .	94
Tabela 6 – Definição dos parâmetros . . . . .	96
Tabela 7 – Tabela de Resultados <i>Heterogeneous Fixed Fleet Vehicle Routing Problem with Time Window</i> (HFVRPTW) . . . . .	97
Tabela 8 – Exemplo de $w$ proporcional a capacidade (HR101) . . . . .	108
Tabela 9 – Valores de $w$ para cada tipo de veículo para cada instância . . . . .	112
Tabela 10 – Tabela de Resultados para HFVRPTW-LC . . . . .	114
Tabela 11 – Tabela de média dos indicadores de cada categoria de instância . .	115

## LISTA DE ABREVIATURAS E SIGLAS

**ACO** *Ant Colony Optimization*

**ALNS** *Adaptive Large Neighborhood Search*

**CVRP** *Capacited Vehicle Routing Problem*

**FSM** *Fleet Size and Mix Vehicle Routing Problem*

**GA** *Genetic Algorithm*

**HFVRP** *Heterogeneous Fixed Fleet Vehicle Routing Problem*

**HFVRPTW** *Heterogeneous Fixed Fleet Vehicle Routing Problem with Time Window*

**HFVRPTW-LC** *Heterogeneous Fixed Fleet Vehicle Routing Problem with Time Window and Loading Cost*

**HVRP** *Heterogeneous Vehicle Routing Problem*

**MaaS** *Mobility-as-a-service*

**PSO** *Particle Swarm Optimization*

**SA** *Simulated Annealing*

**TaaS** *Transportation-as-a-Service*

**TMS** *Transport Management Systems*

**TSP** *Traveling Salesman Problem*

**TS** *Tabu Search*

**VRPLC** *Vehicle Routing Problem with Loading Cost*

**VRPTW** *Vehicle Routing Problem with Time Windows*

**VRP** *Vehicle Routing Problem*

**WVRP** *Weighted Vehicle Routing Problem*

## SUMÁRIO

<b>1</b>	<b>INTRODUÇÃO</b>	<b>14</b>
1.1	QUESTÕES DE PESQUISA	18
1.2	OBJETIVO	18
1.2.1	<b>Objetivos Específicos</b>	<b>18</b>
1.3	METODOLOGIA DE PESQUISA	19
1.3.1	<b>Caracterização Metodológica</b>	<b>19</b>
1.3.2	<b>Procedimento Metodológico</b>	<b>19</b>
1.4	ESTRUTURA DO TRABALHO	20
<b>2</b>	<b>REFERENCIAL TEÓRICO</b>	<b>22</b>
2.1	PROBLEMA DE ROTEAMENTO DE VEÍCULOS (VRP)	22
2.2	ROTEAMENTO DE VEÍCULOS CAPACITADO	28
2.3	JANELA DE TEMPO	30
2.4	FROTA HETEROGRÉA	32
2.5	CUSTO DE CARGA	36
2.5.1	<b>VRP with Loading Cost</b>	<b>36</b>
2.5.2	<b>Weighted VRP</b>	<b>39</b>
2.6	MÉTODOS DE SOLUÇÃO	40
2.6.1	<b>Heurísticas de Construção/Inserção</b>	<b>40</b>
2.6.2	<b>Heurísticas de Destrução/Remoção</b>	<b>43</b>
2.6.3	<b>Heurísticas de Melhoria/Refinamento</b>	<b>44</b>
2.6.4	<b>Operações com Ruído</b>	<b>46</b>
2.6.5	<b>Meta-heurísticas</b>	<b>47</b>
2.7	REVISÃO DOS TRABALHOS RELACIONADOS	52
2.7.1	<b>Considerações sobre os trabalhos relacionados</b>	<b>59</b>
<b>3</b>	<b>SOLUÇÃO PROPOSTA</b>	<b>62</b>
3.1	FORMALIZAÇÃO MATEMÁTICA DO PROBLEMA	62
3.2	MÉTODO PROPOSTO	64
3.2.1	<b>Representação da Solução</b>	<b>64</b>
3.2.2	<b>Função Objetivo Hierárquica</b>	<b>65</b>
3.2.3	<b>Modelagem da Solução</b>	<b>66</b>
3.3	MODELAGEM DO ALGORITMO	71
3.3.1	<b>Solução Inicial Semi-Paralela</b>	<b>71</b>
3.3.2	<b>ALNS</b>	<b>73</b>

3.3.2.1	<b>Random Removal</b>	77
3.3.2.2	<b>Worst Removal</b>	78
3.3.2.3	<b>Shaw Removal</b>	79
3.3.2.4	<b>Route Removal</b>	80
3.3.2.5	<b>Greedy Insertion</b>	81
3.3.2.6	<b>Regret-k Insertion</b>	83
3.3.3	<b>Busca Local</b>	85
3.4	IMPLEMENTAÇÃO DA TECNOLOGIA	87
4	<b>EXPERIMENTOS E ANÁLISE DOS RESULTADOS</b>	89
4.1	PROTOCOLO DE EXPERIMENTAÇÃO	89
4.2	BASE DE DADOS	92
4.3	DEFINIÇÃO DE PARÂMETROS DO ALGORITMO	94
4.4	RESULTADOS DOS EXPERIMENTOS PARA <i>HFVRPTW</i>	95
4.4.1	<b>Resultados</b>	95
4.4.2	<b>Tempo de Execução</b>	99
4.4.3	<b>Análise do Comportamento do Método Proposto</b>	100
4.5	RESULTADOS DOS EXPERIMENTOS PARA <i>HFVRPTW-LC</i>	107
4.5.1	<b>Definição do coeficiente de custo de carga (<i>w</i>)</b>	107
4.5.2	<b>Resultados</b>	113
4.6	CONSIDERAÇÕES SOBRE OS EXPERIMENTOS	116
5	<b>CONCLUSÃO</b>	120
5.1	Trabalhos Futuros	122
	<b>REFERÊNCIAS</b>	124

## 1 INTRODUÇÃO

O Problema de Roteamento de Veículos - *Vehicle Routing Problem* (VRP) é um problema prático de logística de transporte de bens ou pessoas. Pode ser definido como um problema de seleção de rotas que têm como origem um depósito central de onde partem os veículos para atender demandas de clientes distribuídos geograficamente, sujeitas a diversas características e restrições. A importância deste problema se dá pela necessidade de se otimizar rotas com intuito de reduzir os custos relacionados ao transporte e melhorar o gerenciamento da operação logística (BRAYSY; GENDREAU, 2005; TOTH; VIGO, 2014).

Aplicações no mundo real, tanto na América do Norte quanto na Europa, mostraram que o uso de sistemas computacionais geraram economia substancial entre 5% a 20% nos custos globais de transporte (TOTH; VIGO, 2002). Dado o impacto que os custos logísticos têm no setor industrial e de serviços, grandes esforços estão sendo realizados em direção à redução de custos, eficiência e sustentabilidade, principalmente alavancada pela utilização de novas tecnologias. Em relação ao transporte de carga, a necessidade de envio de produtos e *commodities* entre cidades e países se fortalece ao passo que as relações comerciais globais se estreiam (STATISTA, 2022).

A forte disseminação dos *Transport Management Systems* (TMS) foi um catalisador do uso de tecnologias na área de transporte para promover uma melhor gestão com o uso de sistemas inteligentes. O estudo de VRP para TMS foca em fazer o roteamento dos veículos que irão coletar/entregar produtos visando otimizar objetivos relacionados a custos operacionais e a satisfação dos clientes (HASLE; KLOSTER, 2007). Segundo McCrea (2021), o TMS está a caminho de se tornar um mercado de 2,5 bilhões de dólares em 2023, impulsionado principalmente após a pandemia de COVID-19.

Além do transporte de cargas, o transporte de pessoas tem sido motivo de grande preocupação e foco de pesquisas nos últimos anos devido à necessidade de modernização dos modais de transporte, dada a rápida urbanização e mudança no estilo de vida social e econômica das pessoas. A infraestrutura urbana congestionada torna os meios de transporte de uso individual cada vez mais cansativos, o que faz com que a relevância dos serviços de mobilidade compartilhada aumente. O *Mobility-as-a-service* (MaaS), também conhecido como *Transportation-as-a-Service* (TaaS) surgiu como resposta a essa demanda e possui mercado de 230 bilhões de dólares até 2025 (CARLIER, 2022). Os modelos de VRP são de extrema importância para a engrenagem destes mecanismos, pois têm objetivo de levar mais eficiência e conveniência aos

serviços de mobilidade, considerando disponibilidade limitada de recursos (veículos) e demanda por transportes compartilhados (*ridesharing*).

O VRP, proposto por Clarke e Wright (1964), é um problema de Otimização Combinatória que, dado um conjunto de clientes, suas demandas de transporte e uma frota de veículos (homogêneos), tem por objetivo determinar um conjunto de rotas que saem do depósito e atendam todos os clientes a um custo mínimo de transporte. O problema clássico de VRP considera apenas a restrição de limite de capacidade de carga da frota de veículos. A essa variação de VRP é dado o nome de *Capacited Vehicle Routing Problem* (CVRP).

O VRP é uma área de pesquisa que contempla diversas características. Devido aos seus diversos potenciais objetivos e ao número de restrições que podem ser consideradas no problema, este problema pode representar diversos cenários reais dentro da logística de transporte e representar redução de custos significativa. Alguns exemplo de objetivos encontrados em problemas de VRP são: redução da distância percorrida, redução do tempo de viagem, redução dos tempos de espera, redução de atrasos, redução da emissão de gás carbônico, aumento do número de atendimentos, aumento do grau de satisfação dos clientes e até mesmo redução do custo geral da operação, quando esse custo pode ser representado de maneira objetiva na modelagem do problema. As restrições para este problema, em geral, estão relacionadas a disponibilidade e características dos veículos (heterogêneos ou homogêneos), jornada máxima de trabalho, distância/duração das rotas, características dos clientes como janela de tempo de atendimento (CORDEAU et al., 2007).

Em cenários reais, dificilmente as demandas dos clientes são atendidas por uma frota de veículos iguais. A composição e dimensionamento da frota de veículos são problemas típicos da logística de transporte e é um *trade-off* entre adquirir e manter uma frota ou subcontratar (alugar) de terceiros. Dimensionar corretamente o tamanho da frota é uma decisão que envolve a escolha da quantidade e dos tipos de veículos a serem utilizados. Essas decisões se baseiam em muitas variáveis como os custos de manutenção, os custos de transporte e, portanto, acabam limitando a quantidade de recursos (veículos) disponíveis para atender um conjunto de demandas (entregas) dos clientes (KOC et al., 2015).

O problema que representa este cenário é denominado de Problema de Rotteamento de Veículos com Frota Heterogênea - *Heterogeneous Vehicle Routing Problem* (HVRP). Esse problema considera veículos com diferentes capacidades e com seus respectivos custos de utilização (curso fixo e variável). Assim, a capacidade do veículo selecionado para uma determinada rota deve ser respeitado como no CVRP e os custos de sua utilização devem ser adicionados ao custo total da operação. (GENDREAU et al., 1999)

Segundo Koç et al. (2016), o HVRP em geral considera frota limitada ou ilimitada de veículos, com seus respectivos custo fixos e/ou variáveis. Os dois principais modelos de HVRP são *Fleet Size and Mix Vehicle Routing Problem* (FSM), introduzido por Golden et al. (1984), que trabalha com frota heterogênea ilimitada de veículos, e *Heterogeneous Fixed Fleet Vehicle Routing Problem* (HFVRP), introduzido por Taillard (1999), no qual o tamanho da frota é pré-determinado (limitada).

As restrições de janela de tempo, segundo o *Vehicle Routing Problem with Time Windows* (VRPTW), consideram não só as distâncias entre os clientes, mas também o tempo de viagem entre eles. Cada cliente possui uma demanda de entrega, uma janela de tempo, a qual contém o intervalo de tempo em que o cliente deve começar a ser atendido, e ainda um tempo de serviço que especifica o tempo necessário para o cliente ser atendido. Essas restrições têm a finalidade de evitar que as rotas excedam as janelas de tempo de atendimento do cliente, o que tornaria a solução inválida (VIEIRA, 2013; HOMBERGER; GEHRING, 2005). Como exemplos de aplicações de VRPTW, tem-se o *delivery* de mercados, a coleta de lixo industrial, os serviços de patrulha de segurança e as rotas de ônibus escolares, onde é necessário ter um período máximo para o início e/ou fim do atendimento (TOTH; VIGO, 2014).

Os problemas clássicos de VRP negligenciam o impacto do peso transportado no custo total da rota, ou seja, considera esse impacto como uma constante. Porém, em cenários reais, este impacto é considerado como uma variável importante e que afeta diretamente o custo final, já que transportar cargas mais pesadas por longos trajetos pode representar um consumo de combustível muito maior. A partir disso, surgiram as definições de Problema de Roteamento de Veículos com Custo de Carga - *Vehicle Routing Problem with Loading Cost* (VRPLC) (TANG; ZHANG; PAN, 2010) e Problema de Roteamento de Veículos Ponderados - *Weighted Vehicle Routing Problem* (WVRP) (ZHANG; TANG; FUNG, 2011), sendo que o segundo é uma generalização do primeiro. Estes problemas consideram um coeficiente de custo para a carga existente nos veículos, o que expressa o impacto que elas exercem no custo total da rota ao longo dos atendimentos dos serviços. Portanto, uma abordagem para sua otimização seria descarregar/entregar primeiro as encomendas com maior peso quando os outros fatores de otimização, tais como a distância, são semelhantes.

Dessa forma, os problemas VRPLC e WVRP apresentam a noção de custo não constante de carga, uma vez que a carga presente nos veículos não se mantém constante ao longo da execução das rotas (aumenta em casos de coleta e diminui em casos de entrega). Assim, este modelo aplica certa penalidade em forma de custo adicional caso o veículo mantenha cargas pesadas por longas distâncias, para representar cenários onde a distância pela qual a carga é carregada afeta nos custos do transporte.

No que diz respeito aos estudos sobre os diferentes tipos de VRP, as variações do tipo VRPTW apresentam um volume de pesquisa significativo. Já a classe de problemas que contemplam frota heterogênea possui menor volume de pesquisa, sendo a relacionada com frota limitada (HFVRP) menos explorada do que a com frota ilimitada (FSM), dado que lidar com recursos limitados de frota torna a solução mais desafiadora. Por fim, as categorias de WVRP e VRPLC possuem pouca abrangência de pesquisa por ser uma vertente de VRP relativamente nova, sendo seus primeiros estudos datados do ano de 2010. Portanto, possuem volume menor de trabalhos, especialmente em conjunto com outras restrições e características.

Embora existam trabalhos relevantes para o VRPLC e WVRP, como os de Tang, Zhang e Pan (2010), Kuo e Wang (2012) e Tang et al. (2013), os quais apresentam outras características relevantes para cenários logísticos, tais como múltiplos depósitos e demandas fracionadas, não identificou-se um trabalho que abordasse o custo de carga em uma frota heterogênea limitada e com janela de tempo para os clientes. Além disso, as definições dos coeficientes de custo de carga são definidos com base em cenários locais transporte, tais como a China em Tang, Zhang e Pan (2010), o que dificulta a replicabilidade para outros cenários de experimentações.

O VRP, independente de sua variação, é um problema combinatorial complexo que pertence à classe NP-Difícil. Portanto, algoritmos exatos são eficientes apenas para cenários com poucos nós (clientes), ou seja, com espaço de busca reduzido. A utilização de heurísticas e meta-heurísticas, como *Simulated Annealing* (SA), *Adaptive Large Neighborhood Search* (ALNS), *Tabu Search* (TS), *Ant Colony Optimization* (ACO), *Genetic Algorithm* (GA) e *Particle Swarm Optimization* (PSO) são mais apropriadas para problemas de escalas maiores (problemas do mundo real), haja vista que esses métodos conseguem se aproximar da solução ótima em tempo computacional aceitável (BRAEKERS; RAMAEKERS; Van Nieuwenhuyse, 2016). Dentre as opções disponíveis, a meta-heurística ALNS tem se mostrado uma abordagem eficiente para a solução de problemas de roteamento, como apresentado em Ropke e Pisinger (2006), Sacramento, Pisinger e Ropke (2019), Schmitt (2020), Gunawan et al. (2020), Mehdi, Abdelmoutalib e Imad (2021), Bayer (2021) e Wen et al. (2022).

Diante das possíveis variações de VRP, este trabalho tem o intuito de abordar o cenário de entregas de mercadorias em que as empresas possuem uma frota pré-definida de diferentes tipos de veículos, o que leva a um problema adicional que é definir os melhores tipos de veículos a serem utilizados na realização das rotas e que contemple a maior quantidade de entregas possível. Esse tipo de problema, além de envolver os custos fixos de utilização dos veículos, e os custos variáveis relacionados a distância percorrida, também contempla o consumo de combustível associado ao peso da carga transportada. Ou seja, transportar cargas pesadas por trajetos mais

longos têm impacto relevante nos custos operacionais. Além disso, em geral, os clientes precisam ser atendidos em horários pré-determinados. Portanto, este cenário se enquadra em um Problema de Roteamento de Veículos com Frota Heterogênea Fixa, Janela de Tempo e Custo de Carga - HFVRPTW-LC.

## 1.1 QUESTÕES DE PESQUISA

Considerando a relevância que a otimização do custo de carga tem em VRPs com frota heterogênea limitada e com janela de atendimento ao cliente, este trabalho apresenta a seguinte pergunta de pesquisa: Como solucionar o HFVRPTW-LC de forma a conseguir um resultado competitivo perante a literatura, em tempo de processamento aceitável e aplicável a grandes operações logísticas? Uma grande operação significa uma operação com centenas de entregas a serem realizadas com uma frota de dezenas de veículos heterogêneos.

Até o momento, a literatura não apresenta trabalhos que combinem todas as características do cenário de transporte definido pelo problema HFVRPTW-LC. Além disso, os trabalhos que abordam o custo de carga não apresentam protocolos de experimentação replicáveis, pois foram realizados utilizando *datasets* proprietários e que não estão disponíveis publicamente.

## 1.2 OBJETIVO

Este trabalho tem como objetivo construir uma solução baseada na metaheurística ALNS para otimização do HFVRPTW-LC, que seja capaz de minimizar os custos relacionados ao uso da frota, selecionando de forma eficiente o conjunto de veículos mais adequado a solução da otimização.

Diante da ausência de trabalhos com experimentos replicáveis e comparáveis a cerca do custo de carga (VRPLC), este trabalho propõe uma experimentação em duas etapas. A primeira realiza a redução do problema, desconsiderando o custo de carga (HFVRPTW), afim de validar o algoritmo ALNS frente a um resultado presente na literatura. Em seguida, o custo de carga é incluído e um estudo amplo é realizado para a definição dos coeficientes de custo de carga e demonstração dos impactos do uso deste custo.

### 1.2.1 Objetivos Específicos

Com base no objetivo geral apresentado, os seguintes objetivos específicos são definidos:

- Definir e modelar matematicamente o Problema de Roteamento de Veículos com Frota Heterogênea Fixa, Janela de Tempo e Custo de Carga (HFVRPTW-LC);
- Adaptar a meta-heurística de busca em vizinhança (ALNS), e seu conjunto de operadores, para solucionar o problema proposto.
- Utilizar um mecanismo robusto para a seleção dos tipos de veículos para cada rota, de modo a evitar seleções enviesadas, tais como ordenações ou priorizações de veículos.
- Realizar experimentos e análises de desempenho do ALNS para o problema reduzido HFVRPTW (sem custo de carga), a fim de avaliar o desempenho da solução frente a resultados relevantes existentes na literatura atual.
- Realizar experimentos e análises de desempenho do ALNS para o problema HFVRPTW-LC (com custo de carga), a fim de definir os coeficientes de custo de carga mais adequados para o *dataset* de experimentação utilizado.
- Avaliar os impactos do uso do custo de carga nos principais indicadores como custo total, distância percorrida e ocupação dos veículos.

### 1.3 METODOLOGIA DE PESQUISA

Esta seção apresenta a caracterização metodológica da pesquisa, assim como o detalhamento do procedimento metodológico utilizado neste trabalho.

#### 1.3.1 Caracterização Metodológica

De acordo com as classificações definida por Wazlawick (2017), em relação à sua natureza, esta pesquisa é classificada como primária, pois propõe a produção de um novo conhecimento por meio da definição e experimentação de um método de solução para uma variação de VRP ainda não abordado na literatura. Em relação ao seu objetivo, esta pesquisa é considerada como explicativa, pois apresenta os dados acerca dos experimentos e analisa o comportamento do algoritmo em diferentes cenários. Quanto ao procedimento técnico, esta pesquisa é classificada como experimental, pois são feitas manipulações de variáveis de entrada do problema e, por fim, a medição e observação dos resultados a fim de concluir algo sobre a hipótese de pesquisa definida, feita por meio de validação estatística.

#### 1.3.2 Procedimento Metodológico

O procedimento metodológico deste trabalho inicia com um levantamento bibliográfico sobre os problemas de roteamento de veículos envolvidos na proposta:

VRP com Frota Heterogênea Fixa, VRP com Janela de Tempo e VRP com Custo de Carga, bem como sobre as abordagens meta-heurísticas e de busca local aplicadas a essas categorias de problemas. Em seguida, é realizada a conceituação e modelagem matemática do Problema de Roteamento de Veículos com Frota Heterogênea Fixa, Janela de Tempo e Custo de Carga (HFVRPTW-LC), o qual não possui definição formal na literatura.

A partir da conceituação e formalização matemática do problema proposto, é definido o método de solução baseado na meta-heurística ALNS, assim como do conjunto com operadores utilizados. Nesta fase são detalhados os passos para a construção do método de solução proposto, bem como as adaptações para lidar com as peculiaridades do problema, sobretudo no que se refere a seleção de veículos de modo não enviesado por ordenações ou priorizações.

Na sequência, são conduzidos experimentos, os quais são divididos em duas etapas. A primeira busca avaliar o método proposto frente a um resultado existente na literatura. Para isso, reduz-se o problema para uma versão presente na literatura, a qual considera frota heterogênea limitada e janela de tempo (HFVRPTW) e desconsidera o custo de carga. Nesta etapa, são realizadas comparações com um trabalho relevante presente na literatura e análises comportamentais acerca do algoritmo. Na segunda etapa, o custo de carga é considerado (HFVRPTW-LC) e uma análise acerca dos impactos do uso deste custo é realizada por meio da comparação dos resultados da primeira etapa (sem custo de carga) e da segunda etapa (com custo de carga). Para conseguir fazer essa comparação, foi necessário definir os valores dos coeficientes de custo de carga mais adequados para o cenário de frota heterogênea presente no *dataset* de experimentação utilizado. Isso traz como contribuição adicional a possibilidade de permitir a replicabilidade e comparação dos experimentos com trabalhos futuros.

#### 1.4 ESTRUTURA DO TRABALHO

O restante deste trabalho está estruturado da seguinte forma.

O Capítulo 2 apresenta as definições e revisão dos conceitos que envolvem este trabalho, tais como as variações de problemas de roteamento de veículos e métodos heurísticos e meta-heurísticos de solução. Portanto, as características e conceitos referentes ao problema abordado e ao método de solução escolhido serão detalhados com mais profundidade. Também, nesta seção, é apresentada a revisão dos trabalhos relacionados.

O Capítulo 3 apresenta a modelagem matemática do problema proposto e a adaptação da meta-heurística ALNS e de seus operadores de destruição e reconstru-

ção de soluções. É apresentado o diagrama de fluxo que ilustra as etapas do método de solução proposto e, ainda, são detalhadas cada uma das etapas do método, incluindo os seus algoritmos.

O Capítulo 4 detalha os resultados dos experimentos, incluindo a metodologia de experimentação, bem como a análise quantitativa e qualitativa das duas etapas de experimentação do método de solução proposto.

Por fim, o Capítulo 5 resume as conclusões do trabalho, retomando os objetivos propostos e os resultados obtidos, assim como detalha os trabalhos futuros.

## 2 REFERENCIAL TEÓRICO

Este trabalho visa solucionar de maneira eficiente o problema de roteamento de veículos com frota heterogênea fixa, janela de tempo e custo de carga. Neste capítulo são apresentados os conceitos relevantes sobre esse tema. As definições sobre Problemas de Roteamento de Veículos (VRP), bem como as variações pertinentes, respectivas formulações matemáticas e aplicações são apresentadas. Na seção 2.1 é introduzido o conceito de VRP e sua caracterização. Na seção 2.2 é apresentada a versão clássica de VRP e sua modelagem matemática. Na seção 2.3 é apresentado o VRP com janela de tempo. Na seção 2.4 é definido o problema de VRP com frota heterogênea, bem como as características específicas utilizadas neste trabalho. Na seção 2.5 é definida a versão de VRP com custo de carga. Na seção 2.6 são apresentados os métodos heurísticos e meta-heurísticos de solução para problemas combinatórios, especialmente para VRP, e na seção 2.7 é apresentada a revisão dos trabalhos relacionados.

### 2.1 PROBLEMA DE ROTEAMENTO DE VEÍCULOS (VRP)

Um sistema de roteamento é um conjunto organizado de meios que tem como objetivo o atendimento de demandas. Este sistema visa selecionar rotas que incluem pontos de consumo (clientes), a partir de pontos de suprimentos (depósitos). Dantzig e Ramser (1959) foram os primeiros a introduzir o chamado Problema de Despacho de Caminhões (*Truck Despatching Problem*), o qual tinha objetivo de descobrir como uma frota de caminhões poderia servir as demandas de óleo de um determinado número de estações de gás, a partir de uma ponto central, com a menor distância percorrida. Anos depois, Clarke e Wright (1964) generalizaram o problema como um problema de otimização combinatória, o qual foi categorizado no domínio de problemas de logística e transporte. Este problema passou a ser chamado de *Vehicle Routing Problem* (VRP), que tem por objetivo servir um conjunto de clientes geograficamente dispersos no espaço em torno de um depósito central, usando uma frota de veículos, de maneira a minimizar a distância total percorrida por esses veículos.

O aumento no interesse da comunidade internacional de pesquisadores em diferentes variações de VRP não se dá apenas pela notória dificuldade trazida pela complexidade de problemas de otimização combinatória, mas também por sua relevância prática no mundo real, dado que empresas de logística enfrentam diversos problemas relacionados à roteirização de suas demandas sujeita a restrições e diversas características peculiares (TOOTH; VIGO, 2014). As aplicações de VRP podem ser das

mais diversas formas: coleta e entrega de produtos, transporte de pessoas (executivo ou compartilhado), serviços de atendimento programado ou até mesmo roteamento de robôs (WEI; ZHUO, 2010).

O VRP pode ser apresentado como um grafo, como ilustrado na Figura 1, o qual considera um conjunto de clientes, representados pelos vértices, e um conjunto de trajetos, que correspondem às arestas. A solução para este problema consiste em selecionar rotas de custos mínimos, nas quais seus respectivos veículos partam de um depósito e visitem todos os clientes, de forma que cada cliente seja visitado uma única vez por um único veículo.

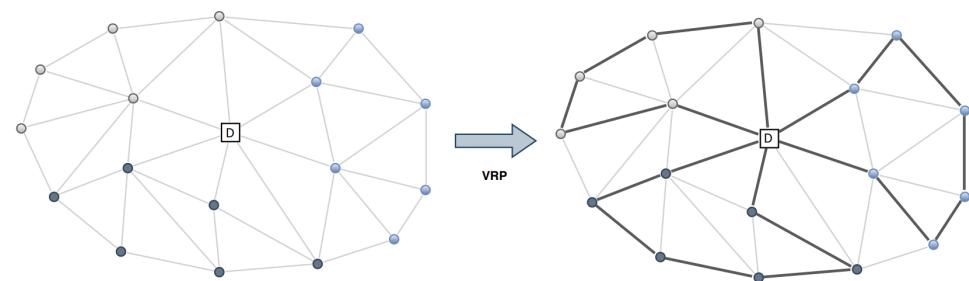


Figura 1 – Representação de Roteamento de Veículos (VRP)

Fonte: Próprio Autor

As entradas prováveis para um VRP são: localização e capacidade do(s) depósito(s), localização e demanda(s) dos clientes, tamanho da frota de veículos, tipos de veículos e suas capacidades, e custos dos deslocamentos entre os clientes. Dadas essas informações, o objetivo mais comum deste problema é estabelecer um roteamento com distância mínima. Contudo, existem outros objetivos que podem ser utilizados, de maneira a gerar modelos mais complexos e aproximados da realidade, como os objetivos de reduzir a quantidade de veículos/rotas, o consumo de combustível, a emissão de poluentes, a duração das rotas, o tempo de espera, os custos de utilização dos veículos, os custos por unidade de distância, o custo de carga, dentre outros.

As diversas variações de VRP têm recebido atenção especial dos pesquisadores, sobretudo aqueles modelos que combinam várias restrições. São exemplos dessas variações os modelos que adicionam janelas de tempo de atendimento aos clientes, utilizam tipos de veículos distintos, múltiplos depósitos, demandas variadas e atendimentos dinâmicos (imprevisíveis). A lista a seguir contém algumas variações de VRP segundo Toth e Vigo (2014) e uma breve descrição para cada variação, de acordo com as suas restrições e características.

- **Capacitated VRP (CVRP):** esta variação adiciona restrições de capacidade aos veículos homogêneos, não permitindo que ela seja ultrapassada durante a construção da rota.

- **Heterogeneous VRP (HVRP):** este problema considera frota limitada ou ilimitada com diferentes tipos de veículos (capacidade, custos fixos e custos variáveis).
- **VRP with Time Window (VRPTW):** neste problema, os clientes têm janelas de tempo que determinam o intervalo de tempo em que devem ser atendidos, não permitindo atrasos ao longo da rota.
- **Periodic VRP (PVRP):** para este tipo de problema é adicionado um horizonte de planejamento em que todos os clientes possuem uma agenda de visitas que especifica em quais dias deste horizonte eles devem ser visitados.
- **Multi-Depot VRP (MDVRP):** nesta variação, o modelo permite a existência de mais de um depósito, cada um com seu conjunto de veículos.
- **Green VRP (GVRP):** este modelo incorpora restrições relacionadas a problemas ambientais, como poluição gerada, gasto de combustíveis, etc.
- **Open VRP (OVRP):** este modelo torna não obrigatório o retorno do veículo ao depósito.
- **VRP with Pickup and Delivery (VRPPD):** para este caso, cada cliente possui um local de coleta de demanda e um respectivo local de entrega da mesma demanda.
- **VRP with Backhauls (VRPB):** este modelo divide os clientes em dois grupos (*linehaul* e *backhaul*). O primeiro são clientes referentes a demandas de entrega e o segundo correspondem aos clientes com demandas de coleta. As rotas são selecionadas de forma que os clientes do tipo *backhaul* são visitados depois dos clientes do tipo *linehaul*.
- **Dynamic VRP (DVRP):** neste problema, parte das coletas e entregas são desconhecidas *a priori*, ou seja, antes de iniciar o deslocamento dos veículos, e são disponibilizadas ao longo do tempo (*tempo real*).
- **VRP with Loading Cost (VRPLC):** este modelo considera um coeficiente de custo para o peso/volume da carga transportada entre os clientes para representar o impacto do peso no custo total. Também denominado de *Weighted Vehicle Routing Problem* (WVRP).

Dada a gama de diferentes variações de VRP, torna-se relevante estabelecer uma sistematização por meio de uma taxonomia, de modo a determinar as especificações de problemas mais bem elaborados de roteamento de veículos, chamados

de problemas enriquecidos. A Tabela 1 divide a taxonomia em características de cenários e características físicas, detalhadas na sequência para melhor entendimento. Este estudo taxonômico foi baseado em Goldbarg, Goldbarg e Luna (2015). A Tabela 1 apresenta as principais características relacionadas à formulação do VRP, sendo que as destacadas em negrito são aquelas abordadas pelo presente trabalho. Abaixo segue uma breve descrição de cada uma das características apresentadas na Tabela 1.

Características de Cenário	Características Físicas
1. Dados de entrada	1. Veículos
a) <b>Estático</b>	a) Frota Homogênea
b) Dinâmico	b) <b>Frota Heterogênea</b>
c) <b>Determinístico</b>	c) <b>Número Limitado (Fixo)</b>
d) Estocástico	d) Número Ilimitado
e) Tempo Real	e) <b>Restrições de Capacidade</b>
2. Componentes das Decisões de Gestão	2. Restrições de Tempo
a) <b>Roteamento</b>	a) <b>Restrições no consumidor</b>
b) Estoque e Roteamento	b) Restrições na vias de acesso
c) Localização e Roteamento	c) <b>Restrições no(s) depósito(s)</b>
d) Plano de produção e distribuição	d) <b>Tempo de serviço</b>
e) Planejamento de motoristas	e) <b>Latência - tempo de espera</b>
3. Número de Depósitos	f) <b>Uma única janela de tempo</b>
a) <b>Um depósito</b>	g) Múltiplas janelas de tempo
b) Múltiplos depósitos	
4. Tipo de Operação	3. Custos de Transporte
a) <b>Pickup &amp;/ou Delivery</b>	a) <b>Dependentes da Distância</b>
b) <i>Backhauls</i>	b) Dependentes do Tempo
c) <i>Dial-a-ride</i>	c) <b>Dependentes do Veículo</b>
5. Fracionamento de Itens	d) <b>Dependentes da Carga</b>
a) Fracionamento permitido	e) Dependentes da Operação
b) <b>Fracionamento não permitido</b>	f) Dependentes dos Riscos
6. Horizonte de Planejamento	g) Dependentes da Latência
a) <b>Horizonte Único</b>	
b) Múltiplos Horizontes	4. Restrições Específicas
7. Uso Múltiplos dos Veículos	a) Rotas Abertas
a) <b>Única viagem</b>	b) Restrições de Acessibilidade
b) Múltiplas viagens	c) Visitas Simultâneas
	d) Balanceamento de Rotas
	5. Função Objetivo
	a) <b>Mono-objetivo</b>
	b) Multi-objetivo

Tabela 1 – Taxonomia para o Problema de Roteamento de Veículo (VRP)

Fonte: Adaptado de Goldbarg, Goldbarg e Luna (2015)

- **Características de Cenário:** representam aspectos estratégicos e táticos de transporte e determinam os principais componentes, como locais e números de depósitos e quantidade de visitas a cada cliente.
  1. **Dados de entrada:** dados estáticos são conhecidos *a priori* e dados dinâmicos são conhecidos ou disponibilizados com o passar do tempo. O aspecto estocástico está relacionado a informações incertas e são descritas por uma distribuição aleatória, enquanto o aspecto determinístico disponibiliza a informação de maneira precisa. A característica de tempo real/*online*, em geral, está atrelada aos problemas dinâmicos, pois, denota que as informações dinâmicas se tornam disponíveis em tempo real, enquanto as operações de decisão estão ocorrendo.
  2. **Componentes das Decisões de Gestão:** problemas de roteamento dizem respeito a atender os clientes (demandas). Eventualmente mais variáveis de decisão são consideradas como aspectos de produção, planejamento dos motoristas ou tripulação, estoque, entre outros.
  3. **Número de Depósitos:** os problemas e roteamento contêm detalhes importantes sobre a quantidade e localização dos depósitos. Assim, pode-se ter variações onde sejam permitidos vários depósito de vários níveis e peculiaridades diferentes, como frota disponível, capacidade de estoque e formas de acesso.
  4. **Tipo de Operação:** diferentes tipos de operações são realizadas para as demandas específicas dos clientes. Quando os produtos são carregados no depósito e entregues nos clientes ou/e coletados nos clientes e entregues no depósito (*Pickup &/ou Delivery*). Há casos especiais em que a ordem de visita aos clientes é determinante (*Backhals*) ou em casos como *Dial-a-ride*, quando nós de entrega são distintos de nós de coleta.
  5. **Fracionamento de Itens:** nos casos em que a entrega fracionada é permitida, o cliente pode ter sua demanda atendida em partes, ou seja, coleta/entrega de um número determinado de produtos (não há fracionamento de um mesmo produto), o que pode implicar em mais visitas. Nos casos em que isso não é permitido, as demandas são atendidas na sua integralidade.
  6. **Horizonte de Planejamento (dia de trabalho):** os problemas que impõem mais de um horizonte, ou período de planejamento, dividem os seus atendimentos em períodos, de forma que os dados de entrada estão disponíveis no início de cada período programado. Em cada período de planejamento é necessário decidir quais demandas serão atendidas e quais serão atrasadas para o próximo período.

7. **Uso Múltiplos dos Veículos:** nesta classe, é permitido ou não que o mesmo veículo realize mais de uma viagem (múltiplas viagens), ou seja, ao retornar ao depósito o veículo pode ser usado para novos atendimentos.
- **Características Físicas:** referem-se a aspectos operacionais, como detalhes sobre a frota de veículos e modelo da janela de tempo.
    1. **Veículos:** as variações de tipos de veículos são impostas por diferentes necessidades dos clientes, diferentes características das vias, de preocupações ambientais, entre outros. Problemas que consideram diferentes tipos de veículos são denominados de frotas heterogêneas. As restrições de capacidade de carga estão geralmente associados a grande parte dos problemas de roteamento, pois, introduz limites de carga, o que retrata a realidade de forma fiel. Por fim, o número de veículos disponíveis é um fator importante para a solução do problema, ele pode ser dividido em números fixos ou ilimitados de veículos.
    2. **Restrições de Tempo:** as janelas de tempo são restrições que impõem um limite de tempo para os clientes serem atendidos, também associados a um tempo de serviço. Em geral, estas restrições são referentes aos clientes e ao(s) depósito(s). A latência é uma característica de alguns tipos de problemas de roteamento que permite que o veículo chegue antes da janela de tempo do cliente começar, indicando um tempo de espera.
    3. **Custos de Transporte:** os custos podem estar relacionados a diversas variáveis de um problema, o que pode tornar mais complexa a tomada de decisão. Os custos podem estar atrelados a distância, tempo de viagem, tempo de espera, tipos de veículos, peso transportado, operações nos clientes, entre outros.
    4. **Restrições Específicas:** em problemas mais elaborados de VRP, algumas restrições específicas são adicionadas no intuito de tornar o problema mais próximo a determinado caso real. A restrição de rotas abertas, por exemplo, permite que o veículo não retorne ao depósito. As visitas simultâneas permitem que um cliente receba mais de uma entrega simultaneamente, o que exige infraestrutura especial.
    5. **Função Objetivo:** a tomada de decisão final é baseada sobretudo na função objetivo, que determina um valor para uma determinada solução do problema. Em geral, as abordagens de roteamento de veículos utilizam funções mono-objetivo. Contudo, quando se deseja adicionar aspectos mais complexos a tomada de decisão, tais como: riscos ambientais, custos nas

operações aos clientes, mão de obra dos motoristas e etc., são utilizadas funções multi-objetivo.

De acordo com as variações levantadas, nas seções 2.2 à 2.5 são introduzidos os modelos de VRP abordados neste trabalho com maior profundidade, bem como suas restrições e particularidades. As versões destacadas são VRP capacitado (versão clássica), VRP com frota heterogênea, VRP com janela de tempo, VRP com custo de carga.

## 2.2 ROTEAMENTO DE VEÍCULOS CAPACITADO

O modelo clássico de VRP é denominado de Problema de Roteamento de Veículos Capacitado (CVRP), o qual introduz limites de carregamentos, que são as restrições para VRP mais estudadas segundo a literatura. Neste modelo, os veículos que realizam o transporte das mercadorias para atender as demandas dos clientes possuem uma quantidade máxima de carga transportada, que não pode ser excedida durante a rota. Os veículos possuem a mesma capacidade de carregamento (frota homogênea); os clientes, que possuem demandas previamente conhecidas, devem ser visitados apenas uma vez e as rotas devem iniciar e terminar no depósito. (TOTH; VIGO, 2014)

O VRP pode ser representado como um problema de teoria dos grafos. Seja  $G = (V, A)$  um grafo completo e não-direcionado, no qual  $V = \{0, 1, \dots, n\}$  é o conjunto de  $n+1$  vértices e  $A = \{a_{01}, a_{02}, \dots, a_{n-1,n}\}$  o conjunto de arcos. Os vértices  $i = 1, \dots, n$  correspondem aos clientes, os quais possuem demandas não negativas  $q_i$ , e o vértice  $i = 0$  se refere ao depósito. Um custo não negativo  $c_{ij}$  é associado a cada aresta  $a_{ij} \in A$ , o qual representa o custo de viagem entre os vértices  $i$  e  $j$ , em geral, relacionado a distância  $d_{ij}$  entre esses dois vértices. Se  $c_{ij} = c_{ji}$  para todo  $i, j \in V$ , então o problema é considerado simétrico; caso contrário, é chamado de assimétrico. (BURAK; VURAL; REISMAN, 2009)

O problema de roteamento de veículos capacitado consiste em encontrar um conjunto de  $k$  circuitos, que correspondam a um roteamento de custo mínimo definido pela soma dos custos dos arcos de todos os circuitos, desde que:

- Cada circuito visite o vértice 0 (depósito);
- Cada vértice  $i \in V \setminus \{0\}$  seja visitado exatamente por um circuito; e
- A soma das demandas dos vértices visitados por um circuito não excedam a capacidade  $C$  do veículo utilizado.

A formulação matemática do CVRP utiliza  $Kn^2$  variáveis binárias, dadas por:

$$x_{ijk} = \begin{cases} 1, & \text{se o veículo } k \text{ trafega pelo trecho } (i, j); \\ 0, & \text{caso contrário.} \end{cases}$$

para  $i, j \in \{0, 1, \dots, n\}$ ,  $i \neq j$  e  $k \in \{1, \dots, K\}$ , sendo  $n$  o número de clientes e  $K$  o conjunto de veículos.

A formulação matemática do CVRP é dada por:

$$\text{Minimizar} \quad \sum_{i=0}^n \sum_{\substack{j=0 \\ j \neq i}}^n \sum_{k=1}^K c_{ij} x_{ijk} \quad (2.1a)$$

$$\text{sujeito a} \quad \sum_{k=1}^K \sum_{j=1}^n x_{0jk} \leq K \quad (2.1b)$$

$$\sum_{j=1}^n x_{0jk} = \sum_{j=1}^n x_{j0k} = 1, \quad k = 1, \dots, K \quad (2.1c)$$

$$\sum_{k=1}^K \sum_{j=0}^n x_{ijk} = 1, \quad i = 1, \dots, n \quad (2.1d)$$

$$\sum_{j=0}^n x_{ijk} - \sum_{j=0}^n x_{jik} = 0, \quad k = 1, \dots, K, \quad i = 1, \dots, n \quad (2.1e)$$

$$\sum_{k=1}^K \sum_{i \in S} \sum_{j \in S} x_{ijk} \leq |S| - v(S), \quad \forall S \subseteq V \setminus \{0\}, \quad |S| \geq 2 \quad (2.1f)$$

$$\sum_{j=1}^n q_j \sum_{\substack{j=0 \\ j \neq i}}^n x_{ijk} \leq C, \quad k = 1, \dots, K \quad (2.1g)$$

$$x_{ijk} \in \{0, 1\}, \quad i, j = 0, 1, \dots, n, \quad k = 1, \dots, K. \quad (2.1h)$$

A Equação 2.1a define a função objetivo, a qual representa a soma dos custos  $c_{ij}$  referente às distâncias dos caminhos feitos pelos  $K$  veículos utilizados na solução. A restrição 2.1b garante que no máximo  $K$  veículos sairão do depósito e a restrição 2.1c assegura que todas as rotas tenham início e fim no depósito. As restrições 2.1d e 2.1e garantem que os clientes sejam visitados exatamente uma vez, e que o veículo que chega em um cliente saia dele. A restrição 2.1f evita que sejam formados subciclos que não incluam o depósito. Por fim, a restrição 2.1g garante que uma rota não exceda a capacidade do veículo e as restrições 2.1h limitam os valores que as variáveis  $x_{ijk}$  podem receber (VIEIRA, 2013).

O CVRP é um problema combinatorial da classe NP-Difícil, ou seja, para um espaço de busca complexo os algoritmos exatos não são capazes de encontrar a solução ótima em tempo polinomial. A Figura 2 ilustra o grafo  $G$  direcionado que representa um roteamento de veículos. O grafo é composto por 12 vértices, sendo que o vértice 0 representa o depósito e os vértices de 1 a 11 indicam os clientes. Cada

vértice referente a um cliente possui uma demanda não negativa  $q_i$ . Cada aresta, que representa o trecho entre os vértices, contém um custo  $c_{ij}$  referente à distância deste trecho. Na Figura 2 os clientes são atendidos a partir de três circuitos (C1, C2 e C3) que partem do depósito.

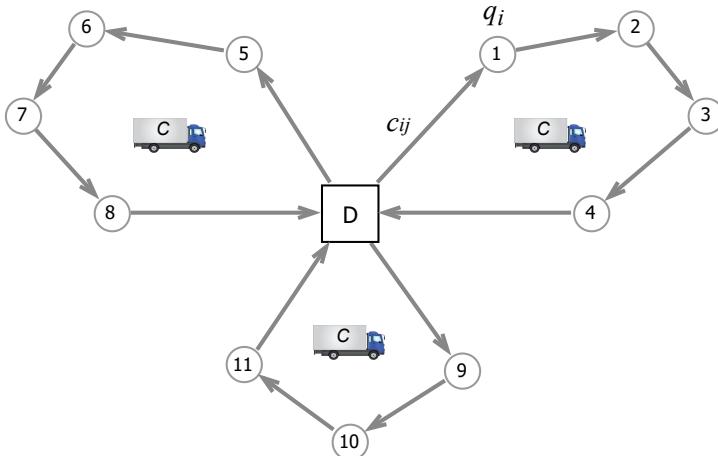


Figura 2 – Configuração do Roteamento de Veículos Capacitado (CVRP)

Fonte: Próprio Autor

De modo a atender todas as demandas dos clientes e, ao mesmo tempo, respeitar as restrições de capacidade dos veículos, conforme restrição presente no CVRP, os circuitos apresentados na Figura 2 (C1, C2 e C3) não podem conter nós (clientes) que excedam os limites do veículo, considerando as cargas já alocadas no mesmo.

### 2.3 JANELA DE TEMPO

O Problema de Roteamento de Veículos com Janela de Tempo (VRPTW) é uma extensão do problema capacitado (CVRP), apresentada por Solomon (1987), ao qual se associa a cada cliente um intervalo de tempo dentro do qual o cliente deve começar a ser atendido por algum veículo. Esse intervalo de tempo é denominado de *janela de tempo*. As janelas de tempo podem ser do tipo *hard* ou *soft*. Em uma janela de tempo do tipo *hard*, se um veículo chegar antes do início de janela de tempo de um cliente, ele deve esperar até que o cliente possa ser atendido. Em geral, este tempo de espera não gera custos adicionais. Já o tipo *soft* permite que a janela de tempo seja violada a um custo adicional (penalidade). Exemplos comuns de problemas que lidam com janela de tempo são transporte escolar e distribuição de produtos de forma agendada. Este problema também é da classe *NP-hard*, pela adição de restrições ao modelo CVRP, que pertence a esta mesma classe (TOTH; VIGO, 2014).

O problema pode ser representado como um grafo  $G = (V, A)$ , como no CVRP, sendo que  $A$  é o conjunto de arcos e  $V$  é o conjunto de  $n$  vértices, dos quais o vértice 0

é o depósito e os demais são os clientes. O conjunto de clientes devem ser atendidos por uma frota  $K$  de veículos homogêneos de capacidade  $C$ , que partem do depósito e retornam até ele ao final de rota. A cada cliente  $i$  é associado uma demanda  $q_i$ , um tempo de serviço  $s_i$  e um intervalo da janela de tempo  $[e_i, l_i]$  (período para o início do atendimento).

O depósito também possui uma janela de tempo  $[e_0, l_0]$ , em que  $e_0$  representa o momento em que os veículos podem começar as entregas ou coletas e  $l_0$  indica o limite de tempo em que todos os veículos devem estar de volta ao depósito. A cada arco  $(i, j)$  está associado um custo  $c_{ij}$  que representa a distância  $d_{ij}$  entre os vértices  $i$  e  $j$  ou o tempo necessário para trafegar entre eles, representado por  $t_{ij}$ . Nesse modelo, para simplificar, é adotado que  $(c_{ij} = d_{ij} = t_{ij})$ , o que indica que a velocidade dos veículos é constante. Além disso, são permitidas esperas nos clientes, caso o veículo chegue antes do início da janela de tempo especificada. A Figura 3 apresenta as características desse problema.

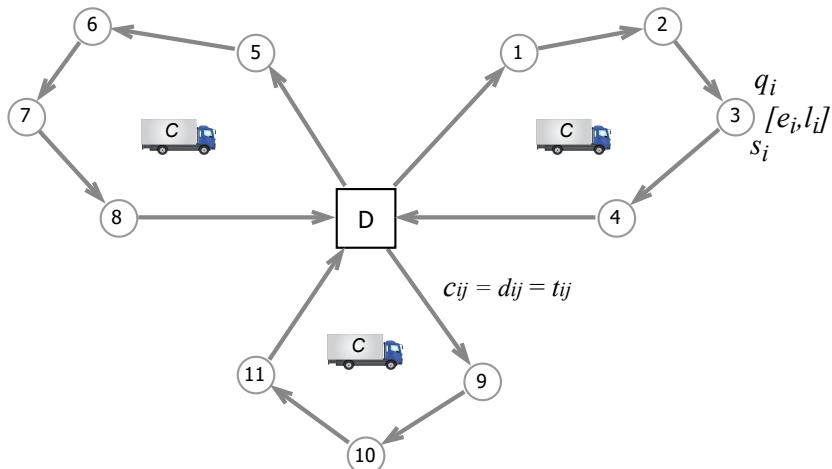


Figura 3 – Configuração do Roteamento de Veículos com Janela de Tempo (VRPTW)

Fonte: Próprio Autor

O objetivo do problema é minimizar os custos de distribuição, ou seja, o roteamento com menor distância total percorrida, de forma a respeitar as restrições de janela de tempo e capacidade dos veículos. Da mesma maneira que no CVRP, no VRPTW são utilizadas  $Kn^2$  variáveis binárias ( $x_{ijk}$ ) para indicar a presença ou não de um veículo  $k$  no arco  $(i, j)$ . Para representar o instante em que o cliente  $i$  é atendido, são utilizadas  $n$  variáveis reais, indicadas por  $b_i$  (TOTH; VIGO, 2014).

A formulação matemática do VRPTW é dada por:

$$\text{Minimizar} \quad \sum_{i=0}^n \sum_{\substack{j=0 \\ j \neq i}}^n \sum_{k=1}^K c_{ij} x_{ijk} \quad (2.2a)$$

$$\text{sujeito a} \quad \text{Restrições (2.1b a 2.1h) do CVRP} \quad (2.2b)$$

$$\sum_{k=1}^K \sum_{\substack{i=0 \\ i \neq j}}^n x_{ijk} (b_i + s_i + t_{ij}) \leq b_j, \quad j = 1, \dots, n \quad (2.2c)$$

$$e_i \leq b_i \leq l_i, \quad i = 0, \dots, n \quad (2.2d)$$

A função objetivo 2.2a e as restrições 2.1b a 2.1h são remanescentes do problema CVRP. A restrição 2.2c relaciona os instantes de início do atendimento dos clientes visitados consecutivamente por um mesmo veículo. Esta restrição impede que haja atraso ao servir o cliente  $j$ , saindo do cliente  $i$ . A folga gerada nesta restrição é considerada como um tempo de espera de veículo no cliente, até que sua janela de tempo comece e o mesmo possa ser atendido. A restrição 2.2d garante que o atendimento a um cliente  $i$  seja iniciado dentro da sua janela de tempo (BRAYSY; GENDREAU, 2005).

## 2.4 FROTA HETEROGRÊNEA

A frota de veículos em uma empresa de transporte usualmente é heterogênea, dado que ao longo do tempo veículos diferentes são incorporados a ela. Ter veículos de capacidades e características distintas resulta em uma melhor adaptação às variações de demandas dos clientes a serem servidos e permite atendimentos com um custo mais eficiente (TARANTILIS; KIRANOUDIS; VASSILIADIS, 2004).

O dimensionamento ou composição da frota é problema comum na indústria e é um *trade-off* entre adquirir/manter uma frota de veículos e subcontratar o transporte, o que se torna uma decisão importante para as empresas. O dimensionamento da frota envolve a escolha das quantidades e dos tipos de veículos a serem usados, decisão esta frequentemente embasada pela capacidade deles. Porém, outras variáveis podem afetar essa decisão, como taxas, custos de transporte, dimensão da demanda esperada e, em última instância, o próprio roteiro/percurso programado para os veículos (KOÇ et al., 2016).

O Problema de Roteamento de Veículos com frota heterogênea (HVRP) considera tipos de veículos distintos, com diferentes capacidades e custos. Este problema pode ser considerado uma generalização do CVRP e, portanto, é da classe NP-Difícil. O HVRP pode ser modelado como um grafo em que são dados  $n$  clientes e suas demandas  $q_i$ , e uma frota de veículos compostas por  $|P|$  diferentes tipos de veículos. A frota  $K$  é dividida em conjuntos de veículos homogêneos, sendo

$K = K^1 \cup K^2 \cup \dots \cup K^{|P|}$ . Cada tipo de veículo  $p = 1, \dots, |P|$  tem uma capacidade  $C^p$ , pode ter um custo fixo de utilização  $f^p$ , um custo variável  $r^p$  (em geral por unidade de distância) e pode ou não ter um limite de quantidade de veículos. A cada aresta  $(i, j) \in A$  está associado um custo  $c_{ij}^p = d_{ij} \times r_{ij}$ , em que  $d_{ij}$  é a distância do trajeto  $(i, j)$  (TOTH; VIGO, 2014). A Figura 4 representa um roteamento utilizando essas características.

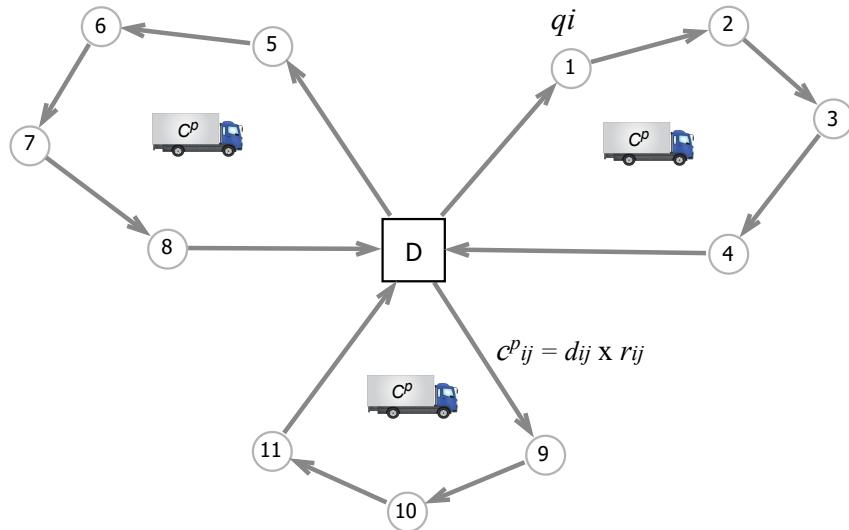


Figura 4 – Configuração do Roteamento de Veículos com Frota Heterogênea (HVRP)

Fonte: Próprio Autor

Os diferentes modelos de HVRP são determinados por características relacionadas ao limite da frota e a existência ou não de custos adicionais. Assim, as características a seguir variam de acordo com a especificidade de cada problema:

- A frota de veículos pode ser **limitada**, ou seja, no máximo  $|K^p|$  veículos do tipo  $p$  devem ser usados, ou **ilimitada** de forma que  $|K^p| \geq n$  para todo  $p$ ;
- O **custo fixo**  $f^p$ , referente cada veículo ao ser utilizado para executar uma rota, pode ser considerado ou ignorado; e
- O **custo variável**, atrelado a cada arco  $(c_{ij}^p)$ , pode ter valores diferentes para cada tipo de veículo (dependente do veículo) ou valores iguais para todos os tipos de veículos (independente do veículo), para todo  $p$ .

A formulação matemática do HVRP é dada por:

$$\text{Minimizar} \quad \sum_{i=0}^n \sum_{\substack{j=0 \\ j \neq i}}^n \sum_{k=1}^K x_{ijk} c_{ij}^p + f^p \sum_{j=1}^n \sum_{k=1}^K x_{0jk} \quad (2.3a)$$

$$\text{sujeito a} \quad \text{Restrições (2.1b a 2.1h) do CVRP} \quad (2.3b)$$

O modelo matemático do HVRP é baseado no modelo do CVRP, onde o objetivo continua sendo minimizar os custos totais de distribuição. Porém, nesse caso estes custos são relacionados ao custo variável por distância e custo fixo de utilização de cada veículo, considerando os diferentes tipos. As restrições do problema são as mesmas do problema CVRP. Se a frota de veículos for limitada este limite deve ser devidamente respeitado.

Segundo Koç et al. (2016), em geral, os problemas de HVRP consideram uma frota limitada ou ilimitada de veículos distintos, em que cada tipo de veículo tem um custo fixo ao ser utilizado para atender um conjunto de demandas conhecidas. Dois modelos principais de VRPs heterogêneos são introduzidos:

- **FSM (Fleet Size And Mix):** introduzido por Golden et al. (1984), utiliza uma frota heterogênea ilimitada. Esse tipo tem o intuito de definir qual a melhor configuração de frota e os melhores roteiros para o atendimento de uma operação. É aplicada em situações em que se deseja definir a melhor composição da frota antes de adquirir/alugar os veículos.
- **HF (Heterogenous Fixed Fleet):** definido por Taillard (1999), utiliza uma frota conhecida previamente (limitada). Portanto, ele busca definir as melhores rotas considerando recursos limitados (frota predeterminada). É aplicada em situações mais comuns do dia a dia das empresas de transporte, que já possuem a frota de veículos e desejam a definição dos roteiros de menor custo possível.

Dado que o FSM não possui limite de frota, é possível inferir que soluções que atendam todos os clientes são encontradas com facilidade. Além disso, métodos de solução menos complexos são capazes de definir os melhores veículos para cada rota, de maneira que ao definir uma rota, o tipo de veículo que melhor se encaixa (*best-fit/cheapest*) em termos de capacidade e custos ao atender os respectivos clientes pode ser escolhida sem a necessidade de verificar se há ou não veículos disponíveis do tipo selecionado (TAILLARD, 1999).

Já para o problema HFVRP o desafio é lidar com recursos limitados para o atendimento de um conjunto de demandas. Dessa maneira, dado que o atendimento de todos os clientes não é garantido, a solução não é trivial. Métodos de solução que utilizam critérios de prioridades ao definir as rotas, como, por exemplo, iniciar a criação de rotas do maior para o menor veículo ou inserir a maior demanda no veículo com maior espaço vazio, podem levar a resultados sub-ótimos e ainda soluções que não atendam todos os clientes. Dessa forma, é um desafio definir um método de solução para o HFVRP que escolha o veículo mais adequado para cada rota e que consiga atender o maior número de demandas a um custo mínimo (TAILLARD, 1999).

Outras variações de HVRP podem existir segundo dois critérios: 1) objetivo e 2) presença ou ausência de restrições de janela de tempo. A seguir, são apresentadas estas variações, segundo Koç et al. (2016). O objetivo de ambos FSM e HFVRP é minimizar o custo total que, além da distância, pode incluir custo fixo (F) de utilização dos veículos e custo variável (V) por unidade de distância percorrida. Com isso, as variações com relação a objetivos são: 1) FSM com custo fixo e custo variável  $FSM(F,V)$ ; 2) FSM com apenas custo fixo  $FSM(F)$ ; 3) FSM com apenas custo variável  $FSM(V)$ ; 4) HFVRP com custo fixo e custo variável  $HFVRP(F,V)$ ; 5) HFVRP apenas com custo variável  $HFVRP(V)$ ; e 6) HFVRP apenas com custo fixo  $HFVRP(F)$  (KOÇ et al., 2016). Em geral, problemas do mundo real são expressos pelo problema  $HF(F,V)$ , com frota definida e ambos os custos envolvidos.

O custo variável é, em geral, expresso por um custo por unidade de distância (custo por quilômetro rodado, por exemplo) em ambos os casos FSM e HFVRP. No caso do custo fixo, é importante observar que a aplicação deste custo no FSM (frota ilimitada) pode representar custo de compra do veículo, por exemplo. Já no HFVRP (frota limitada) o custo fixo pode representar o custo de aluguel, amortização ou até um custo de *setup* do veículo antes de realizar a rota, já que o custo de compra não seria aplicado pois a frota já é pré-definida.

Duas extensões das variações FSM e HFVRP surgem quando restrições de janelas de tempo são utilizadas pelos clientes. Estas são denominadas de FSMTW e HFTW, respectivamente. Com isso, duas medidas podem ser usadas para computar o custo total a ser minimizado: 1) medida baseada no tempo em rota (T), a qual é a soma dos custos fixos dos veículos com a duração das rotas, excluindo o tempo de serviço; e 2) medida baseada na distância percorrida pelas rotas (D), que consiste na soma dos custos fixos de utilização dos veículos com a distância total das rotas (caso padrão do VRPTW de Solomon (1987), como apresentado na seção 2.3). Este último pode ainda ser incrementado com o custo variável por unidade de distância (KOÇ et al., 2016).

Com isso, surgiram outras variações para FSM e HFVRP relacionadas à janela de tempo e medida base do custo: 1) FSMTW com objetivo T, denominada de  $FSMTW(T)$ ; 2) FSMTW com objetivo D, denominado de  $FSMTW(D)$ ; 3) HFTW com objetivo T, denominado  $HFTW(T)$ ; e 4) HFTW com objetivo D, denominado de  $HFTW(D)$ . O presente trabalho propõe utilizar a versão  $HFTW(D)$  em que a frota é limitada, os clientes têm janela de tempo e a distância é a medida base de custo.

## 2.5 CUSTO DE CARGA

O custo do carregamento é uma parte importante no consumo de combustível ao longo das rotas e esse consumo muda ao longo das rotas devido a mudança da carga (TANG et al., 2013). Em casos de *delivery*, por exemplo, a rota inicial no depósito com o total de entregas a serem feitas e à medida que os atendimentos são realizados a carga dentro dos veículos diminui e, consequentemente, o consumo de combustível diminui. Assim, como a carga dentro dos veículos não é constante, o impacto dela no consumo de combustível (custo da rota) também não é constante.

Na prática, ao transportar produtos especiais como materiais perigosos, alimentos perecíveis, animais, entre outros, podem ser cobrados custos extras por esse transporte. Esta parte do custo em geral está relacionada com o peso da carga transportada e pode representar até 50% do custo total (TANG; ZHANG; PAN, 2010).

### 2.5.1 VRP with Loading Cost

O problema clássico de roteamento de veículos (CVRP) possui uma limitação quanto à assimilação dos custos de carga, pois considera a carga e o impacto dela no custo total como uma constante, ou seja, considera que o peso transportado não muda ao longo da rota, o que não acontece em operações reais de transporte. Ao negligenciar o impacto da carga os resultados da otimização acabam sendo piores do que o esperado quando estes resultados são postos em prática no mundo real, já que o peso transportado varia conforme os clientes são atendidos.

A primeira formalização de um problema de VRP que considera custo de carga foi o VRPLC (*Vehicle Routing Problem with Loading Cost*), feito por Tang, Zhang e Pan (2010). Nele, um coeficiente de custo de carga foi introduzido, fazendo com que o objetivo seja minimizar os custos totais, compostos por: 1) custo variável, que é a combinação do custo por distância percorrida e o custo por unidade de peso por unidade de distância e 2) custo fixo de utilização dos veículos. O custo variável por distância e o custo fixo de utilização dos veículos são os mesmos utilizados no VRP com frota heterogênea apresentados na Seção 2.4. Nesse modelo, a carga é considerada uma variável de fluxo, em que a cada trecho entre clientes possui um valor distinto, já o peso transportado muda ao longo da rota ao passo que os clientes são atendidos (TANG; ZHANG; PAN, 2010).

Ao considerar a carga como variável, este modelo se torna mais eficiente que o VRP padrão, pois apresenta maior economia de custo, já que o modelo pode escolher atender demandas maiores no início da rota para evitar realizar um roteiro muito distante carregando um volume/peso muito grande, o que causaria maior consumo de combustível (TANG et al., 2014).

Para exemplificar, a Figura 5 demonstra o comportamento esperado para o problema com a representação da carga variável. Supondo um depósito D e três clientes cujas demandas devem ser entregues (*delivery*): cliente 1 com demanda  $q = 10$ , cliente 2 com demanda  $q = 20$  e cliente 3 com demanda  $q = 15$  igualmente distantes entre si e entre o depósito (distância igual a 10).

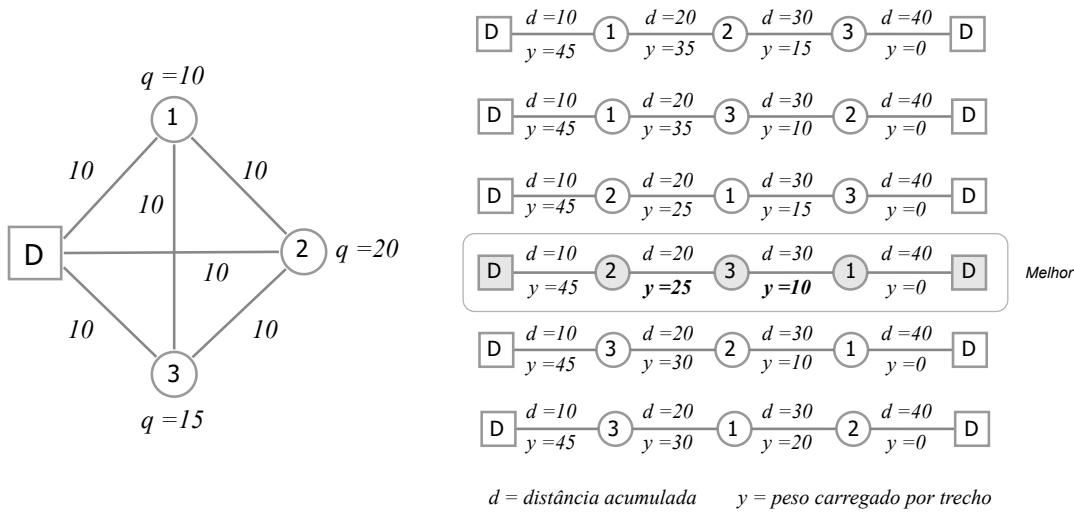


Figura 5 – Exemplo do Roteamento de Veículos com Custo de Carga (VRPLC)

Fonte: Próprio Autor

A Figura 5 apresenta possíveis soluções para o problema, destacando a distância acumulada  $d$  e o peso  $y$  referente a carga contida no veículo entre os clientes atendidos. Para o problema clássico CVRP, em que o impacto do custo de carga é negligenciado, a informação de peso acumulado  $y$  não é considerada e do ponto de vista de distância não há solução melhor nesse cenário, pois a distância acumulada é sempre a mesma em todas as possíveis soluções ( $d = 40$  ao final da rota). Porém, na prática, em cenários reais a carga é variável e transportar pesos maiores por trajetos mais longos implica em um custo total maior. Assim, para o VRPLC a solução  $D \rightarrow 2 \rightarrow 3 \rightarrow 1 \rightarrow D$  é considerada a mais adequada, pois o veículo sai do depósito com carga  $y = 45$  e, na sequência, visita o cliente 2, e passa a trafegar com  $y = 25$  até o cliente 3, que após o atendimento passa a transportar apenas  $y = 10$  até o cliente 1. Assim, nessa solução o impacto da carga é menor uma vez que percorre menores distâncias com grande quantidade/peso de demandas no veículo.

Apesar de considerar a possibilidade de atender prioritariamente as demandas maiores, esta decisão não é trivial e não pode-se afirmar que atender estas demandas primeiro levará aos menores custos, pois os custos por distância percorrida e possíveis custos fixos de utilização dos veículos não devem ser descartados no modelo. Assim, esta decisão se torna um *trade-off* em relação aos custos de carga, distância percorrida e custos da frota (TANG et al., 2014).

Este problema, como uma extensão do VRP, é considerado um problema combinatorial NP-Difícil e pode ser definido como um grafo  $G = (V, A)$ , em que  $A$  é o conjunto de arcos e  $V$  o conjunto de  $n$  vértices, dos quais o vértice 0 é o depósito e os demais são os clientes. Cada cliente possui uma demanda não negativa  $q_i$ . O conjunto de clientes devem ser atendidos por uma frota de  $K$  veículos de capacidade homogênea  $Q$ , que partem do depósito e retornam até ele ao final de rota. Uma distância  $d_{ij}$  representa a distância entre os vértices  $i$  e  $j$ . Uma variável binária  $x_{ijk}$  indica se o veículo  $k$  trafegou entre os clientes  $i$  e  $j$ . Uma variável de fluxo  $y_{ijk}$  representa quanto de carga/peso o veículo  $k$  está carregando entre os clientes  $i$  e  $j$ . Na função objetivo do problema são levados em conta três coeficientes de custo:

- $C_f$  = custo fixo por utilização de cada veículo;
- $C_d$  = custo variável por unidade de distância; e
- $C_w$  = custo por unidade de peso carregado por unidade de distância percorrida.

O VRPLC tem por objetivo minimizar a função considerando os custos relacionados à distância, peso transportado e frota utilizada, seguindo a formulação matemática a seguir:

$$\text{Minimizar} \quad \sum_{i=0}^n \sum_{\substack{j=0 \\ j \neq i}}^n d_{ij} \sum_{k=1}^K x_{ijk} (C_d + C_w y_{ijk}) + C_f \sum_{j=1}^n \sum_{k=1}^K x_{0jk} \quad (2.4a)$$

$$\text{sujeito a} \quad \text{Restrições (2.1b a 2.1h) do CVRP} \quad (2.4b)$$

$$\sum_{k=1}^K \sum_{\substack{j=0 \\ j \neq i}}^n (x_{jik} y_{jik} - x_{ijk} y_{ijk}) = q_i \quad i = 1, \dots, n \quad (2.4c)$$

$$0 \leq y_{ijk} \leq Q x_{ijk}, \quad i, j = 0, 1, \dots, n \quad (2.4d)$$

Segundo o modelo matemático, o objetivo do problema VRPLC é minimizar a função de custo composta pelos custos variáveis por distância percorrida e custo de carga transportada por unidade de peso por unidade de distância, somados aos custos fixos de utilização dos veículos. O modelo está sujeito às mesmas restrições do CVRP (2.1b - 2.1h). Além disso, restrições relacionadas a variável de fluxo de carga  $y_{ijk}$  também são impostas. A restrição 2.4c impõe a relação entre a demanda do cliente  $i$  e a carga transportada entre as arestas que ligam este cliente, ou seja, a diferença entre o fluxo de demanda que chega e sai do cliente  $i$  deve ser exatamente a demanda  $q_i$ . A restrição 2.4d representa a limitação da variável  $y_{ijk}$ , a qual deve estar entre 0 e a capacidade  $Q$  máxima do veículo.

### 2.5.2 Weighted VRP

Uma generalização do VRPLC foi introduzida por Tang et al. (2013), chamada de WVRP (*Weighted Vehicle Routing Problem*), a qual também considera o peso carregado como uma parte variável da função de custo. O termo “*weight*” é um termo geral que pode ser estendido para representar quantidade ou volume de objetos (carga/bens/passageiros) para serem entregues ou dar importância/prioridade a determinados clientes.

O modelo matemático é semelhante ao VRPLC em termos de variáveis e restrições. Porém, o WVRP não leva em consideração custos fixos e variáveis extras relacionadas aos veículos, apenas a variável de fluxo de carga  $y_{ijk}$  de forma generalizada para representar a quantidade ou peso transportado entre dois clientes. Além disso, a função objetivo é generalizada para um função mais simplificada em comparação com o VRPLC. Com essa generalização, a representação do custo de carga pode levar a otimização de outros objetivos secundários, como por exemplo priorizar entregas maiores para obter mais espaço para coletar produtos (TANG et al., 2013).

O modelo matemático para o WVRP é definido da seguinte forma:

$$\text{Minimizar} \quad \sum_{i=0}^n \sum_{\substack{j=0 \\ j \neq i}}^n \sum_{k=1}^K d_{ij} x_{ijk} y_{ijk} \quad (2.5a)$$

$$\text{sujeito a} \quad \text{Restrições do VRPLC} \quad (2.5b)$$

Segundo a modelagem matemática do WVRP, as restrições impostas são as mesmas do problema VRPLC e a diferença está na função objetivo 2.5a mais genérica, que leva em consideração a distância percorrida multiplicada pela quantidade de carga transportada em cada trecho entre clientes utilizado nas rotas.

Ao comparar os modelos de VRPLC e WVRP entende-se que o segundo é uma generalização do primeiro, embora ambos sejam definidos para atingir o mesmo propósito: representar o impacto da carga no custo total por meio de uma variável de fluxo de carga no modelo. Ambos os modelos são válidos para este propósito e a escolha de qual modelo aplicar se dá a cada caso específico. Se o objetivo é apenas representar a carga como variável ou dar prioridade para certos clientes, o modelo WVRP pode ser o mais efetivo e simples para este propósito. Porém, se o objetivo é considerar custos extras relacionados a frota de veículos (custo fixo e variável) e um coeficiente para controlar de forma mais direta o impacto da carga no custo total, o modelo mais adequada pode ser o VRPLC. Desse modo, devido à semelhança com os custos envolvidos no HVRP (custo fixo e variável) e por ser a versão mais específica e direta para tratar do custo de carga, o VRPLC será a versão utilizada neste trabalho.

## 2.6 MÉTODOS DE SOLUÇÃO

A evolução dos algoritmos e o aumento das capacidades computacionais permitiu o surgimento de diversos métodos para tratar problemas de otimização combinatória. O VRP é um problema de otimização combinatória e, portanto, pode ser resolvido de forma exata utilizando Programação Linear Inteira (PLI). Métodos exatos de solução são aqueles que por meio da modelagem matemática do problema encontram a solução ótima. Porém, para os problemas denominados NP-Difíceis, os quais representam grande parte dos problemas de PLI, a utilização de métodos exatos são inviáveis para instâncias grandes devido à explosão combinatória que acompanha estes métodos de solução (SAVELSBERGH, 1990).

Alguns exemplos de métodos exatos utilizados na solução de problemas de PLI são: *Lagrangian Relaxation*, *Column Generation*, *Branch-and-cut*, *Branch-and-bound*, entre outros. Os métodos exatos não fazem parte do escopo deste trabalho dadas as suas limitações na otimização de problemas NP-Difíceis, portanto, não serão abordados com detalhes.

Em virtude das limitações dos métodos exatos para solucionar instâncias consideradas grandes para problemas combinatórios, outras técnicas de solução receberam mais atenção da comunidade científica nas últimas décadas, tais como os métodos de solução aproximada ou sub-ótima, chamados de heurísticas. As heurísticas são métodos que denotam uma regra geral para a solução de um problema sem a aplicação de procedimentos exaustivos. Em outras palavras, são métodos de aproximação que buscam obter soluções de boa qualidade em tempo computacional viável sem necessidade de explorar toda e qualquer solução do espaço de busca antes de chegar a solução final, e sem a necessidade de ter prova matemática de sua convergência (BANDARU; DEB, 2016).

Dado que as heurísticas não garantem que a solução ótima para o problema seja encontrada, o propósito desses métodos é viabilizar a solução de problemas combinatórios para cenários inviáveis de solucionar por meio de métodos exatos. Dentre as diversas classes de heurísticas existentes estão as heurísticas de construção/inserção, destruição/remoção, refinamento e ainda as meta-heurísticas (populacionais ou de solução única), as quais serão apresentadas nas seções subsequentes.

### 2.6.1 Heurísticas de Construção/Inserção

Para os problemas de VRP, heurísticas de construção, também chamadas de heurísticas de inserção, são métodos caracterizados por realizar a criação de uma nova solução (inicialmente vazia) ou até mesmo inserir requisições (clientes) novos em uma solução já existente. Essa heurística tem o objetivo de incrementar uma solução

considerando a adição de novos clientes por meio de critérios de inserção, em geral, relacionados aos custos da solução e restrições impostas pelo problema. A Figura 6 demonstra o comportamento geral de heurísticas de inserção.

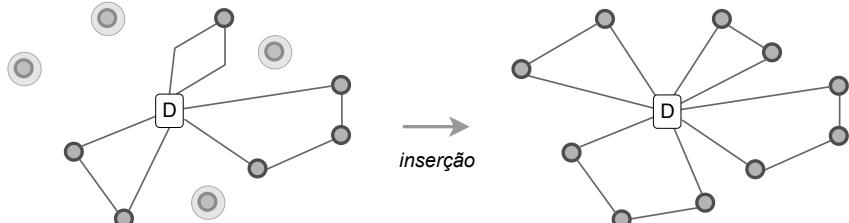


Figura 6 – Operador de inserção

Fonte: Próprio Autor

Comumente, estas heurísticas operam de maneira gulosa, inserindo sucessivamente clientes a cada iteração do processo, usando como critério o menor acréscimo de custo à solução. Embora as heurísticas construtivas sejam capazes de criar novas soluções, ou parte delas, de maneira eficiente e com baixo custo computacional, não garantem que a melhor solução global para o problema seja encontrada. Portanto, são considerados mecanismos importantes para a viabilização das soluções, porém carecem de robustez para encontrar o ótimo global.

Dentre as heurísticas construtivas existentes para a solução do VRP, podem ser destacadas as seguintes, segundo Røpke (2006), Tavares, Lopes e Lima (2009), Avdoshin e Beresneva (2019) e Laporte, Ropke e Vidal (2014):

- **Nearest Neighborhood Heuristic (NN):** este método consiste em inserir a cada passo um novo cliente na solução na posição adjacente ao último cliente inserido, partindo de uma solução vazia. Uma rota é iniciada com o cliente mais próximo do depósito de partida e o próximo cliente a ser inserido é o mais próximo do último cliente já inserido na rota. Os clientes são inseridos até que alguma restrição seja quebrada e uma nova rota seja iniciada. O processo termina quando todos os clientes foram alocados na solução (rotas). Em geral, o critério de escolha do próximo a ser inserido é baseado na distância. Porém, variações desse método também podem utilizar outros critérios, tais como, duração, tempo de espera, lotação do veículo ou até mesmo a combinação entre eles. Este método facilmente é prejudicado por mínimos locais devido à sua limitação de gerar possíveis soluções apenas analisando o custo da próxima inserção, sem analisar o custo da rota como um todo.
- **Savings Based Heuristic (SB):** este algoritmo usa estratégia de *merge* de rotas como método de construção. Inicialmente, é gerada uma rota para cada cliente e posteriormente acontecem sucessivos *merges* (junções) entre rotas, duas a

duas. Uma junção é quando o final de uma rota é conectado com o início de outra. As rotas escolhidas para a junção são aquelas que quando combinadas mantêm a solução factível e geram maior economia de custo total. Este método tem maior potencial de encontrar melhores soluções que o NN, dado que não executa a criação de uma rota de cada vez, e sim avalia possibilidade de junção de rotas quaisquer.

- **Greedy Insertion Heuristic (GI):** neste método clientes ainda não alocados podem ser inseridos em qualquer posição factível, desde que gerem o menor acréscimo de custo possível (avaliação gulosa). Dado um conjunto de clientes não inseridos, são avaliadas as rotas e melhores posições factíveis para inseri-los. O próximo cliente inserido é aquele cuja inserção gera menor aumento no custo. Após a inserção do melhor cliente, os clientes ainda não alocados são reavaliados. O processo se repete até que todos os clientes sejam inseridos. Dado que há a avaliação de todas as possíveis inserções de todos os clientes não alocados a cada passo, este método é considerado mais eficiente em comparação com o NN e o SB. Porém, há de se considerar o fato de que verificar todas as possibilidades de inserção de clientes gera um custo computacional considerável em comparação com métodos mais simples.
- **Regret- $k$ :** esta heurística é semi-gulosa, pois utiliza critérios de arrependimento na tomada de decisão. É levado em consideração o impacto futuro da modificação da solução ao avaliar a inserção de um determinado cliente. De maneira geral, opera de maneira semelhante ao GI com modificações apenas no critério de decisão de qual cliente será inserido no passo atual. A inserção de menor custo é comparada com as  $k$  alternativas de menores custo para a geração do valor de arrependimento. A inserção escolhida é a que gera o menor valor de arrependimento. Esta heurística se destaca perante ao GI devido à projeção futura de uma modificação na solução, o que não ocorre em GI.
- **Semi-parallel Heuristic (SP):** esta heurística cria rotas utilizando qualquer heurística de inserção. Contudo, para a criação de cada rota são considerados todos os clientes do problema, de modo que as rotas criadas podem conter repetição de atendimentos entre elas. Dado o conjunto de rotas criadas, esta heurística define qual a melhor rota para compor a solução final/definitiva utilizando um critério de satisfação, tal como o custo total da rota, a lotação do veículo, a duração, a distância ou até mesmo a combinação desses critérios. Após uma rota ser escolhida para compor a solução final, o processo se inicia novamente com a criação de novas rotas. O algoritmo finaliza quando todos os clientes estiverem inseridos na solução final.

### 2.6.2 Heurísticas de Destrução/Remoção

As heurísticas de remoção ou destruição para problemas de VRP removem uma quantidade determinada de clientes de uma solução (rotas) existente. Em geral, estas heurísticas são aplicadas por métodos baseados em destruição e reconstrução da solução com objetivo de gerar variações na solução. Dada uma solução existente contendo um conjunto de rotas e seus respectivos atendimentos, a heurística de remoção remove um conjunto de atendimentos (nós) da solução segundo critério definido previamente. A Figura 7 demonstra o comportamento geral das heurísticas de remoção.

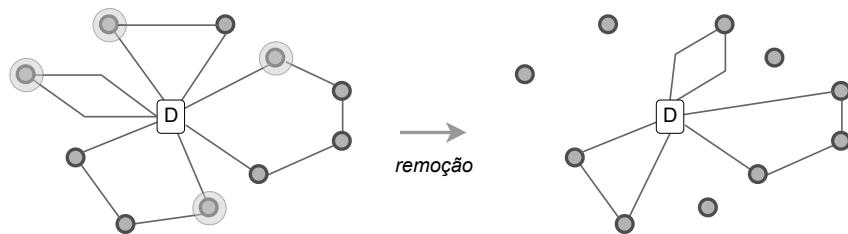


Figura 7 – Operador de remoção

Fonte: Próprio Autor

O objetivo de uma heurística de remoção, ao contrário das heurísticas de construção, não é incrementar e obter uma solução de menor custo, mas sim destruí-la de maneira a permitir a posterior reconstrução da solução de maneira mais adequada. Assim, a aplicação de métodos de remoção são viáveis apenas em combinação com métodos de inserção/construção. Dentre os métodos de remoção existentes, Pisinger e Ropke (2007) destacam os seguintes:

- **Random Removal:** esta heurística remove um número determinado de clientes da solução de maneira aleatória. Esta estratégia é utilizada para gerar maior diversidade nas soluções e eventualmente sair de mínimos locais ao longo do processo de busca.
- **Worst Removal:** este método remove os clientes que mais representam impacto no custo total da solução. É uma heurística gulosa que busca retirar da solução os piores clientes, na tentativa de inseri-los em posições melhores posteriormente.
- **Shaw/Related Removal:** esta heurística remove um conjunto de clientes baseado na similaridade que eles têm entre si. O critério de similaridade pode ser baseado em distância, janela de tempo, quantidade de demanda, entre outros. O objetivo é aumentar a chance de mudanças na solução ao inseri-los novamente nas rotas, uma vez que remover clientes muito diferentes diminui as possibilidades de trocas de posição que geram melhorias.

- **Cluster Removal:** esta heurística é semelhante a remoção Shaw, com a particularidade de buscar a remoção de todos os clientes de um determinado *cluster*, uma vez que a remoção Shaw tem uma tendência de manter na solução original alguns clientes de um mesmo *cluster*. Variações de soluções para a clusterização dos clientes podem existir e ainda serem adaptadas para utilizar diversos aspectos além da distância, tais como as janelas de tempo e demandas dos clientes.
- **Time-oriented Removal:** este método também é uma variação da remoção Shaw em que é levada em consideração apenas a remoção de clientes que são atendidos em momentos próximos. O intuito é incentivar a remoção de clientes de rotas diferentes e que sejam atendidos em momentos semelhantes para que ao serem inseridos novamente na solução, haja troca promissora entre eles em termos de custo.
- **Historical node-pair Removal:** esta heurística utiliza dados históricos entre os pares de clientes para guiar as remoções. Utilizada em métodos iterativos em que é gerada ao menos uma nova solução a cada iteração. É utilizada uma matriz entre os clientes para indicar o peso entre seus pares. A cada nova solução gerada, um valor é adicionado a cada par de clientes contidos na solução. O cliente  $i$  removido é aquele com maior somatório de pesos entre os pares  $(i, j)$ , dado  $j = 1, \dots, n$  e  $n$  o número de clientes.

### 2.6.3 Heurísticas de Melhoria/Refinamento

As heurísticas de melhoria ou refinamento, também chamadas de heurísticas de busca local, são métodos empregados para gerar melhorias em uma solução já existente, possivelmente gerada por uma heurística construtiva. O algoritmo realiza busca por soluções vizinhas em torno da solução existente e ao encontrar solução de melhor qualidade assume esta nova solução como a solução corrente.

Um fator crucial em um método de refinamento é a definição da forma como as soluções são aceitas como uma solução melhor. Assim, a regra de escolha de aceitação de uma solução passa a guiar o processo. Dentre as regras mais usadas estão: método de Descida com Melhor Melhora (*Best Improvement*), a qual aceita a solução da vizinhança que proporciona a maior melhora no valor da função objetivo; e método de Descida com Primeira Melhora (*First Improvement*), que aceita a primeira solução encontrada na vizinhança que ofereça uma melhora no valor da função objetivo (HOOS; STUTZLE, 2004).

A Figura 8 demonstra o comportamento de uma heurística de refinamento, em que a partir de uma solução inicial existente, busca através de uma estrutura de

vizinhança por solução melhor até que não haja mais melhoria (atingir ótimo local). É possível notar que apesar destas heurísticas gerarem melhorias consecutivas na solução corrente, não possui robustez suficiente a ponto de garantir que a melhor solução possível (ótimo global) seja encontrada, como visto na Figura 8.

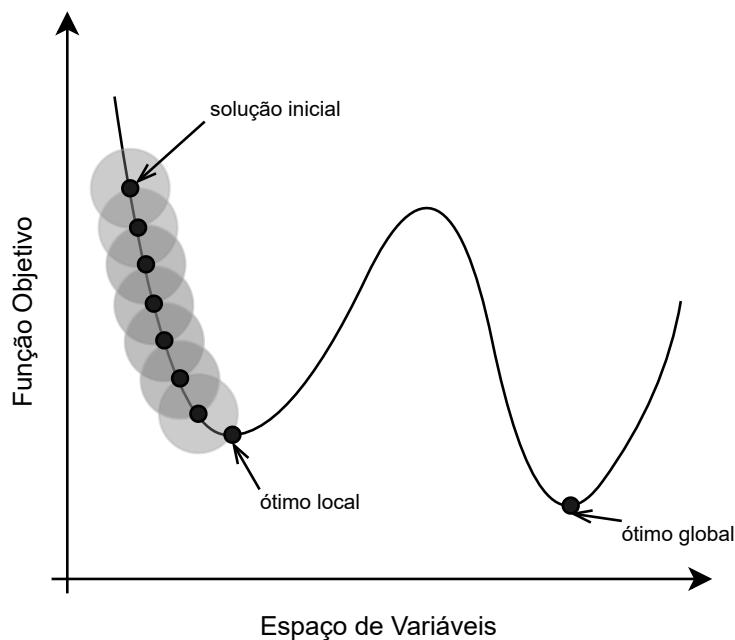


Figura 8 – Comportamento de uma heurística de refinamento

Fonte: Próprio Autor

Diversas heurísticas de estruturas de vizinhança podem ser utilizadas para gerar variações na solução corrente para um problema de VRP. A seguir estão alguns métodos expostos por Laporte e Semet (2002), Laporte, Ropke e Vidal (2014) e Toth e Vigo (2014):

- **Best Adjacent Swap:** este método visa trocar a ordem de atendimento de dois clientes adjacentes em uma mesma rota, selecionando a troca que maximiza a redução de custo da solução.
- **Best Swap / Exchange:** esta heurística tem o objetivo de realizar trocas entre as posições de dois clientes da mesma rota ou de rotas diferentes, em que a troca que causa a maior redução de custo é escolhida.
- **Best Move / Relocate:** este método visa remover um cliente e inseri-lo em outra posição da mesma rota ou de outra rota, desde que maximize a redução de custo da solução.
- **Cross Exchange:** esta heurística visa trocar segmentos entre duas rotas distintas, de maneira que sejam selecionados clientes  $i$  e  $j$  adjacentes de uma rota e os clientes  $k$  e  $l$  adjacentes de outra rota. O método troca os segmentos de

forma que o cliente  $i$  se torna adjacente ao cliente  $l$ , conectando a parte inicial da primeira rota com a parte final da segunda rota. O processo análogo ocorre ao conectar os clientes  $k$  e  $j$ . A operação é realizada se houver redução de custo da solução.

- ***I-exchange / Inverted Range Relocate***: Essa heurística visa selecionar dois atendimentos  $i$  e  $j$  em uma rota e realizar o translado desta sequência de forma invertida para outra rota ou em outra posição na mesma rota. Portanto, ela remove e adiciona de forma invertida uma sequência de atendimentos de forma que a seleção de  $i$  e  $j$  de remoção, da rota e posição de inserção, entre todas as possíveis combinações, maximizem a redução do custo da rota.
- ***2-Opt / Best Range Inversion***: esta heurística visa selecionar dois atendimentos  $i$  e  $j$  em uma rota e realizar a inversão da sequência entre esses pontos, ou seja, ela inverte uma sequência de atendimentos de um trecho da rota de forma que a seleção de  $i$  e  $j$ , entre todas as possíveis combinações, maximizem a redução do custo da rota.

#### 2.6.4 Operações com Ruído

Dadas as heurísticas de inserção, remoção e refinamento apresentadas, podem ser aplicadas variações destas heurísticas utilizando ruído (*noise*) nas operações de maneira que gerem perturbações na solução com intuito de gerar diversificação e sair de mínimos locais. Em determinadas situações o processo de busca chega a ótimos locais, sobretudo por meio da aplicação de heurísticas de refinamento. Contudo, as heurísticas gulosas não são capazes de sair desses mínimos locais. Assim, para perturbar a solução a ponto de gerar saltos consideráveis no espaço de busca e sair dos mínimos locais é necessário inserir fatores de aleatoriedade em heurísticas originalmente gulosas. A Figura 9 demonstra o comportamento desejado ao utilizar ruídos. Ao saltar de um mínimo local para um outro ponto do espaço de busca, pode-se conseguir a exploração de uma região mais promissora e eventualmente atingir o ótimo global.

Para as heurísticas de remoção, as operações de ruído podem estar na escolha dos clientes removidos da solução. No *Worst Removal* um importante ruído é na aplicação de um fator aleatório na escolha do próximo cliente removido, em que escolhe o  $k$ -ésimo melhor cliente a ser removido e não sempre o melhor, dado que  $k$  é dado segundo um fator aleatório. Já na remoção *Shaw* a escolha por similaridade também pode ser aplicada com ruídos, em que não se escolhe sempre o mais similar, mas sim o  $k$ -ésimo mais similar (ROPKE; PISINGER, 2006).

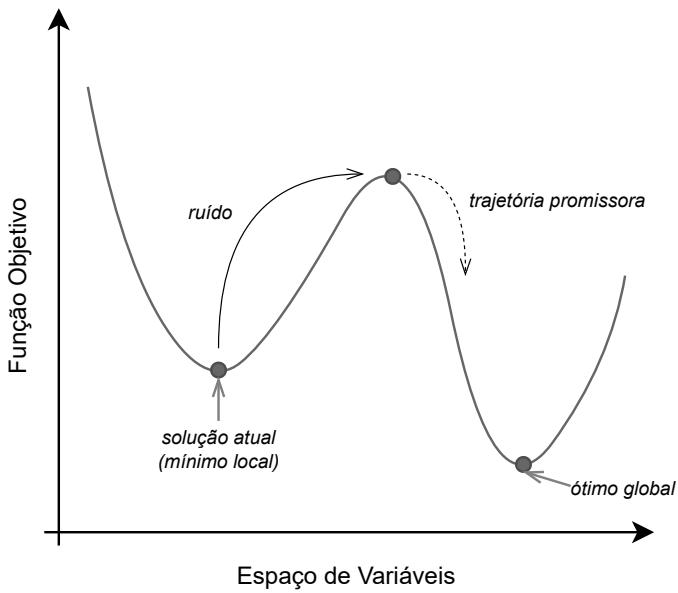


Figura 9 – Comportamento de operações de ruído

Fonte: Próprio Autor

Já para as heurísticas construtivas ou de inserção, pode-se aplicar ruídos na escolha dos clientes a serem inseridos na solução. Para as heurísticas *Greedy Insertion* e *Regret-k*, por exemplo, a escolha da próxima inserção pode ser alterada, de maneira a escolher a  $k$ -ésima inserção de menor custo (*greedy*) ou com menor valor de arrependimento (*regret*), ou até mesmo um fator de acréscimo de custo nos valores das inserções, de modo que a inserção antes dada como a menos custosa (mais atrativa), se torna menos atrativa. Por fim, nas heurísticas de refinamento, os ruídos podem ser aplicados por meio de movimentos e trocas de posição guiados por fatores aleatórios sem a necessidade de verificar se estas operações geraram melhorias na solução, dado que o objetivo é simplesmente gerar perturbações (PISINGER; ROPKE, 2007).

## 2.6.5 Meta-heurísticas

As meta-heurísticas são combinações de métodos heurísticos básicos em arquiteturas de alto nível com objetivo de promover melhor exploração do espaço de busca. De acordo com Dorigo e Stutzle (2004), uma meta-heurística pode ser vista como um método heurístico geral projetado para guiar uma heurística específica para solução de um problema. Estes métodos são iterativos por natureza e em geral usam operações estocásticas (probabilísticas) em seu processo de busca para gerar/modificar um ou mais soluções (BANDARU; DEB, 2016). As regras que compõem e guiam uma meta-heurística podem ser desenvolvidas arbitrariamente ou baseadas em analogias com fenômenos físicos, químicos, biológicos, sociais, semânticos ou até propriedades matemáticas.

As meta-heurísticas são capazes de gerar soluções de alta qualidade em tempo computacional viável, mesmo sem a garantia de que sejam ótimas. O uso de apenas heurísticas, como as construtivas e de refinamento apresentadas nas seções anteriores, não garantem a obtenção de soluções de alta qualidade. Isso ocorre porque heurísticas construtivas permitem a geração de números limitados de soluções e as buscas locais não fornecem meios robustos para escapar de ótimos locais (DORIGO; BIRATTARI; STUTZLE, 2006). Já as meta-heurísticas surgem como um balançamento entre a intensificação e diversificação das soluções no espaço de busca. Dessa maneira, as meta-heurísticas podem ser aplicadas a diferentes tipos de problemas, e não apenas a problemas específicos, dadas as devidas abstrações e robustez capazes de explorar o espaço de busca de maneira mais eficiente em comparação com heurísticas mais simples.

Por intensificação entende-se que o algoritmo meta-heurístico deve observar particularidades das várias regiões do espaço de busca de forma a intensificar a busca nestas regiões por meio de métodos de refinamento. Já em relação ao comportamento de diversificação ou exploração, o algoritmo deve ser capaz de buscar por regiões promissoras do espaço de busca por meio de perturbações e modificações consideráveis nas soluções, sem a necessidade de intensificar a pesquisa por soluções nessas regiões (BANDARU; DEB, 2016).

De forma geral, as meta-heurísticas podem ser divididas em dois tipos: de busca populacional e de solução única (também chamadas de meta-heurísticas de busca local). As meta-heurísticas de busca local exploram o espaço de busca de maneira iterativa através de movimentos sobre a vizinhança, em busca de soluções promissoras. Já as meta-heurísticas populacionais trabalham mantendo um conjunto de soluções a cada iteração, as quais são combinadas interativamente para a geração de soluções ainda melhores, o que permite a exploração de regiões promissoras do espaço de soluções (COSTA, 2012).

Outra maneira de categorizar as meta-heurísticas, segundo Goldbarg, Goldbarg e Luna (2015), é com base no modo como a solução é gerada ou aprimoradas: por heurística construtiva, em que uma ou mais soluções são geradas do zero a cada iteração; e por métodos evolutivos ou de melhoramento, em que uma ou mais soluções são modificadas a partir de soluções já existentes. Além disso, em geral as meta-heurísticas possuem mecanismos de compartilhamento de informações heurísticas de maneira a guiar a convergência no processo de busca. Estas informações são compartilhadas ao longo das iterações de maneira direta, tais como a combinação de partes de duas soluções promissoras para gerar uma solução melhor ainda, ou de maneira indireta, como a utilização de estruturas de memória ou que acumulam informações das soluções geradas ao longo da busca, e usadas para a geração de novas

soluções guiadas por estas estruturas (GOLDBARG; GOLDBARG; LUNA, 2015).

Para a solução das diversas variações de VRP, as meta-heurísticas que se destacam são apresentadas a seguir:

- **Algoritmo Genético (*Genetic Algorithm - GA*):** algoritmos genéticos são um tipo de procedimento de busca aleatória adaptativa inspirada na evolução de organismos biológicos. A partir de uma população (conjunto) de indivíduos (soluções) representados por cromossomos, diversas gerações (iterações) são criadas com operações como seleção, crossover e mutação, visando continuamente recombinar soluções existentes e guiar seu aprimoramento, melhorando a qualidade dos indivíduos. Sendo assim, é considerada uma meta-heurística populacional evolutiva em que há a troca de informação heurística diretamente entre os indivíduos, por meio da combinação de partes das melhores soluções (crossover e mutação). Por se tratar de um método que trabalha simultaneamente com múltiplas soluções, esse método possui bons resultados ao lidar com problemas com quantidade considerável de restrições e muitos máximos locais, já que é possível explorar simultaneamente várias regiões do espaço de busca (PRINS, 2004).
- **Otimização por Colônia de Formigas (*Ant Colony Optimization - ACO*):** esta categoria de meta-heurísticas são inspiradas no comportamento de formigas na busca por alimento. Nesta meta-heurística, um conjunto de formigas (soluções) são geradas a cada iteração do método e se utilizam de uma estrutura chamada de trilha de feromônio como mecanismo de comunicação indireta entre as formigas (soluções) (DORIGO; STUTZLE, 2003). Cada formiga é responsável por construir uma solução nova utilizando a trilha de feromônio como fonte de informação heurística. Esta trilha é composta/atualizada pelas informações acumuladas das soluções ao longo da busca, de modo que quanto melhor a qualidade da solução, maior a quantidade de feromônio será depositada na trilha. Assim, ao construir a solução, a formiga opta probabilisticamente por caminhos com maior concentração de feromônio. Ao final do processo os trajetos com maior concentração de feromônio devem representar a melhor solução encontrada para o problema (DORIGO; BIRATTARI; STUTZLE, 2006). Por se tratar de um algoritmo de busca construtivo, esse método apresenta boa performance para aplicações dinâmicas, tendo em vista que pode facilmente incorporar modificações. Contudo, quando o problema apresenta muitos ótimos locais, devido a natureza construtiva do método, nem sempre é possível obter qualidade satisfatória no resultado.
- **Otimização por Enxame de Partículas (*Particle Swarm Optimization - PSO*):** esse método bioinspirado em inteligência de enxame (*Swarm Intelligence*) é ins-

pirado na ação de revoada de pássaros, abstraído a um conjunto de partículas independentes (soluções) com suas próprias propriedades (posição, velocidades, inércia, mínimos) que realizam a comunicação e troca de informações (influenciam) outras partículas (BENUWA et al., 2016). Desta forma, múltiplos conjuntos de partículas percorrem um espaço de busca, sendo guiados com base nos mínimos existentes (qualidade das soluções encontradas) e pela troca de informações com outras partículas (mínimos regionais e globais), de modo que ao final do processo se espera a convergência das partículas em torno de uma solução comum (TAO et al., 2008). Assim, é considerada uma meta-heurística populacional de melhoria com troca de informações diretas entre as soluções. Por ser uma abordagem populacional, este algoritmo consegue lidar bem com problemas multimodais com várias restrições, sem que haja uma penalidade grande de tempo computacional.

- **Recozimento Simulado (*Simulated Annealing - SA*):** é uma meta-heurística de solução única que busca a melhoria de uma solução inicial, guiado pelo resfriamento de temperatura (KIRKPATRICK; GELATT; VECCHI, 1983). A temperatura é utilizada para determinar os fatores de diversificação e intensificação do processo de busca. Temperaturas mais elevadas (fases iniciais da busca) indicam que a solução atual pode variar mais em sua exploração de vizinhança, já temperaturas mais baixas (parte final da busca) indicam poucas alterações ou alterações de caráter local (refinamento). Assim, com o passar das iterações a temperatura diminui e a busca se torna mais concentrada, convergindo em um ponto específico do espaço de busca. Esse algoritmo é uma das mais simples meta-heurísticas de otimização utilizadas para tratar o VRP. Embora seja possível utilizá-lo para solucionar o VRP, muitas vezes ele não é utilizado como a rotina principal de otimização, mas sim como uma rotina auxiliar que otimiza diversos parâmetros dos algoritmos principais (FERDI; LAYEB, 2016).
- **GRASP (*Greedy Randomized Adaptive Search Procedure*):** este método visa iterativamente equilibrar os aspectos gulosos, randômicos e de busca local. Portanto, ela funciona como um algoritmo de busca em que suas decisões de construção são tomadas não de forma determinística, sempre em direção ao ótimo, mas atuando de forma semi-gulosa, onde as decisões são feitas com base em vieses provenientes das buscas locais. Esse tipo de meta-heurística é bem mais robusta e computacionalmente custosa quando comparada com uma única heurística, porém pode gerar um resultado mais otimizado, pois não é totalmente enviesada a sempre sugerir uma solução ótima local imediata. (RESENDE; RIBEIRO, 2010)

- **Busca Tabu (*Tabu Search - TS*):** é um algoritmo de solução única que considera uma memória das soluções passadas ao longo da busca. Esta memória é utilizada para prevenir que o algoritmo revisite determinadas soluções (ciclos), para evitar estagnação precoce e movimentos que não causam melhorias (GLOVER, 1986) . Utiliza operações de vizinhança para gerar soluções alternativas e gerar a convergência. Assim, é dada como uma meta-heurística de solução única e de melhorias em que a estrutura de memória é o ponto de concentração da informação heurística para guiar o processo de busca. É um método recomendado para problemas de otimização discreta como o VRP e também como mecanismo combinado com outras meta-heurística, utilizando-se da memória como estrutura auxiliar (BANDARU; DEB, 2016).
- **Busca Larga Adaptativa em Vizinhança (*Adaptive Large Neighborhood Search - ALNS*):** este método é uma busca em vizinhança com mecanismos de adaptação e balanceamento no uso dos operadores que geram modificações na solução (ROPKE; PISINGER, 2006). Esta meta-heurística surgiu para solucionar uma das principais limitações do algoritmo *Simulated Annealing*: a capacidade de alteração das soluções, em que os operadores utilizados são definidos de maneira determinística, de forma a gerar tendências e a ficar preso em mínimos locais. Já no ALNS estes operadores são escolhidos probabilisticamente a cada iteração, o que leva a maior variação e diversidade ao longo da busca e em casos de estagnação de mínimos, é capaz de gerar fugas dos mesmos. O ALNS utiliza o conceito de destruição e reconstrução da solução, inspirado no método LNS (*Large Neighborhood Search*), definido por Shaw (1998), em que operadores de remoção são utilizados para remover clientes da solução e operadores de inserção são responsáveis por inseri-los em posições mais promissoras (como as apresentadas nas seções 2.6.1 e 2.6.2). Portanto, durante o processo de execução do algoritmo existem diversos ciclos de diversificação (grandes modificações nas solução para mudança de vizinhança) e intensificação (procura por modificações melhores que melhorem a solução corrente) dentro de uma mesma vizinhança, por meio de operadores de destruição e de reconstrução da solução. Além disso, em geral, utiliza o mecanismo de resfriamento da temperatura para conduzir a busca, de forma análoga ao SA. A meta-heurística ALNS permite utilizar diversas heurísticas como operadores dentro do algoritmo. Portanto, a própria meta-heurística se torna capaz de identificar qual operador deve ser utilizado com base em seu critério de sucesso, o que permite que problemas que requerem diferentes estratégias de otimização utilizem operadores adequados, melhorando o resultado do algoritmo em diferentes cenários (PISINGER; ROPKE, 2007).

Dado que os métodos exatos se mostraram incapazes de gerar soluções para problemas de larga escala em tempo computacional aceitável, esta seção apresentou o levantamento de diversos métodos e de solução para os problemas de VRP complexos. Ou seja, os métodos heurísticos e meta-heurísticos são mecanismos robustos e eficazes para solucionar tais problemas.

Diversos métodos heurísticos foram apresentados para diferentes propósitos (remover, inserir e melhorar a solução). Porém, estes métodos individualmente não são capazes de gerar soluções eficientes, apesar do tempo computacional de processamento ser significativamente mais rápido. Portanto, as meta-heurísticas demonstram poder de convergência para soluções de maior qualidade se comparados com métodos apenas heurísticos.

Dentre as opções de meta-heurísticas, as classificadas como populacionais possuem robustez e capacidade de gerar diversidade de soluções de maneira natural, dado que mantém um conjunto de soluções a cada iteração. Contudo, por gerar ou modificar este conjunto de soluções, estes métodos se tornam computacionalmente mais caros, o que para problemas de VRP em larga escala e em cenários reais pode não ser interessante. Assim, as meta-heurísticas de solução única e de melhoria se mostram alternativa mais eficiente em termos computacionais e com capacidade de gerar soluções tão boas quanto métodos populacionais. Por estes motivos, este trabalho propõe um método de solução para o HFVRPTW-LC baseado na meta-heurística ALNS, dado que ela apresenta capacidade de alcançar soluções de boa qualidade e com bom desempenho computacional.

## 2.7 REVISÃO DOS TRABALHOS RELACIONADOS

Nesta seção são apresentadas as referências de trabalhos relacionados a esta presente dissertação. As pesquisas apresentadas possuem relação com pelo menos uma das características de VRP utilizadas nessa dissertação: frota heterogênea limitada, contendo ou não janela de tempo (HFVRP e HFVRPTW, respectivamente) e custo de carga WVRP/VRPLC.

O trabalho de Taillard (1999) foi o precursor no que se refere a problema de roteamento de veículos com frota heterogênea limitada (HFVRP), em que o objetivo é definir a configuração dos roteiros e quais tipos de veículos usados, considerando recursos limitados. Neste trabalho, é utilizada heurística baseada em geração de colunas CG (*Column Generation*) em conjunto com estratégias exatas por meio do CPLEX para solucionar o HFVRP considerando apenas custo variável por unidade de distância (sem considerar custo fixos). É apresentado um conjunto de testes contendo instâncias de 50, 75 e 100 clientes, o qual se tornou o principal *benchmark* para este

problema ao longo da literatura. Embora use técnicas exatas, o que inviabiliza a utilização para cenários maiores, os resultados foram promissores.

O trabalho de Prins (2002) soluciona o problema de HFVRP com Multi-trip para aplicações em larga escala. O método abordado é a combinação entre diversas heurísticas de inserção com métodos de busca local, em que os veículos são preenchidos em ordem crescente de capacidade (menores veículos primeiro). Neste artigo é proposta uma estratégia construtiva baseada em *merge* com economia de custo segundo o método *Savings-Based Method* de Clarke e Wright (1964). Os resultados foram satisfatórios, em especial em comparação com métodos para VRP clássico. Instâncias para testes de cenários reais de uma fabricante de móveis na França foram definidas neste trabalho.

O VRP com frota homogênea limitada (*VRP-m*) foi apresentado por Lau, Sim e Teo (2003), em que considera frota pré-definida de veículos iguais. Neste trabalho, a prioridade é maximizar o número de atendimentos, em seguida a minimização do número de veículos usados e por último a minimização da distância total, já que não possui custos bem definidos. A solução é baseada em uma TS (*Tabu Search*), que contém heurísticas construtivas e posterior intensificação por busca local. É introduzido o conceito de *Holding List*, que consiste em uma lista com clientes não atendidos. Esse conceito se tornou importante ao longo da literatura dado o limite da frota, fazendo com que nem todos os clientes possam ser atendidos em todos os cenários.

Os trabalhos de Tarantilis, Kiranoudis e Vassiliadis (2003) e Tarantilis, Kiranoudis e Vassiliadis (2004), ambos desenvolvidos pelos mesmos autores, abordam o HFVRP com custo variável, sem custo fixo e com objetivo de minimizar o custo total da operação. Utilizam o mesmo método para a geração de solução inicial, que consiste em alocar o cliente não atendido com a maior demanda no veículo com maior capacidade disponível. O processo é seguido por um método meta-heurístico em que o Tarantilis, Kiranoudis e Vassiliadis (2003) aplica um LBTA (*List-based Threshold Accepting*) o qual utiliza operações de vizinhança para atingir novas soluções sem a possibilidade aceitar pioras na solução. Já o trabalho de Tarantilis, Kiranoudis e Vassiliadis (2004) aplica uma meta-heurística BATA (*Backtracking Adaptive Threshold Accepting*), em que o objetivo é possibilitar *uphill* (piora na solução) para escapar de mínimos locais. São utilizados movimentos de vizinhança como 2-Opt, 1-1 Exchange e 1-0 Exchange para a variação da solução corrente e o operador a ser utilizado é escolhido aleatoriamente, bem como a escolha dos clientes envolvidos nestes movimentos. Ambos os trabalhos realizam testes de *benchmarks* utilizando as instâncias definidas por Taillard (1999) e a meta-heurística BATA atingiu melhores resultados (6 melhores soluções de 8 instâncias) por conseguir sair de mínimos locais.

Em Li, Golden e Wasil (2007) o autor utiliza o método HRTR (*Record-to-*

*record Travel Algorithm*) para o problema de roteamento com frota heterogênea limitada (HFVRP). Este algoritmo é uma variação determinística do SA em que usa operações de vizinhança e possibilidade de *uphill* (pioras na solução) ao longo da busca para sair dos mínimos locais. Este trabalho gerou melhores soluções para 7 das 8 instâncias de Taillard (1999).

O trabalho de Li, Tian e Aneja (2010) soluciona o problema HFVRP com custo fixo e variável, utilizando um MAMP (*Multi-start Adaptive Memory Programming*) combinado com *Path Relinking*. A cada iteração do método é gerada uma nova solução através de método construtivo, em que a inserção de cada cliente é realizada na rota com maior espaço vazio e na posição de menor custo. Após a construção da solução é aplicado um método baseado em TS (*Tabu Search*) para intensificação da solução corrente. Foram gerados testes utilizando as instâncias de Taillard (1999), considerando custo fixo e variável. O algoritmo gerou melhores resultados para 7 de 8 instâncias de teste.

O trabalho de Zheng, Lu e Sun (2010) utiliza uma PTS (*Parallel Tabu Search*) para solucionar o HFVRP considerando uma função objetivo de desperdício (*waste function*), em que é considerada a distâncias total somada ao desperdício de espaço dos veículos utilizados. Neste trabalho não são considerados custos relacionados à frota heterogênea. Os resultados frente às instâncias de Taillard (1999) não foram satisfatórios, apesar do tempo de execução ter sido um destaque positivo do trabalho.

O trabalho de Brandão (2011) propõe um algoritmo TS (*Tabu Search*) para solucionar o problema HFVRP considerando apenas custo variável (sem custo fixo). Uma solução inicial é gerada através da resolução de um *Traveling Salesman Problem* (TSP) com todos os clientes (*big tour*) utilizando o método de inserção GENI, e posteriormente é realizado um *split* em várias rotas atribuindo o tipo de veículo mais adequado a cada rota. O processo se inicia com o primeiro cliente adjacente ao depósito da solução do TSP e a partir dele é escolhido o menor veículo com capacidade suficiente para atendê-lo. Com o veículo escolhido, a rota segue sendo preenchida com os clientes adjacentes na solução *big tour* até que o veículo não tenha mais espaço vago para atender o próximo cliente e então uma nova rota é iniciada com um novo veículo. O processo finaliza quando todos os clientes forem atendidos ou não haja mais veículos disponíveis. Caso ainda existam clientes não atendidos ao final do processo, estes são inseridos em posição de menor custo com as restrições de capacidade relaxadas, permitindo gerar solução infactível. O modelo definido pelo autor, considera penalidade para soluções infactíveis geradas pela extração dos limites de carga. Dada a solução inicial, o algoritmo TS segue o processo realizando iterativamente movimentos de vizinhança utilizando heurísticas de remoção, re-inserção e *swap*. Além disso, é utilizado método de busca local baseado na heurística GENIUS e

também um processo de *shaking* para tentar sair de mínimos locais. Os experimentos foram realizados utilizando as instâncias de Taillard (1999) e outras instâncias maiores geradas pelo próprio autor. De acordo com os experimentos, o algoritmo igualou resultados da literatura para 6 das 8 instâncias definidas por Taillard (1999), o que demonstra que o método atingiu resultados satisfatórios.

O trabalho de Subramanian et al. (2012) utiliza um algoritmo híbrido com RVND (*Variable Neighborhood Descent with Random Ordering*), ILS (*Iterated Local Search*) e um formulão de SP (*Set Partitioning*) para a solução do HFVRP considerando custo fixo e variável no modelo. A modelagem do problema por SP se dá pela definição de um conjunto  $R$  de possíveis rotas para todos os tipos de veículos disponíveis. O método consiste na aplicação de métodos exatos combinados com métodos heurísticos para que a solução seja viável computacionalmente. Os resultados foram promissores frente aos benchmarks de Taillard (1999), Li, Golden e Wasil (2007) e Brandão (2011).

O trabalho de Li, Leung e Tian (2012) soluciona uma versão do HFVRP com rotas abertas com custo fixo e variável na função objetivo. É definido um algoritmo híbrido MAMP *Multi-start Adaptive Memory Programming* em conjunto com uma TS (*Tabu Search*). A cada iteração do método são geradas  $n$  soluções utilizando uma memória acumulada  $M$  contendo um conjunto de rotas promissoras (MAMP). As soluções geradas sofrem processo de intensificação por meio de uma TS e a melhor entre elas é escolhida para renovar a memória  $M$ . São definidas novas instâncias de *benchmark* para frota heterogênea limitada e rotas abertas.

O problema HFVRP com a minimização de distância e custo fixo é solucionado no trabalho de Naji-Azimi e Salari (2013) por meio de uma heurística baseada em ILP (*Integer Linear Programming Based Heuristic*), a qual pode ser usada como complemento de outras heurística existentes na literatura. Não define uma solução inicial específica, apenas assume a existência de uma solução inicial factível. A heurística trabalha com métodos de destruição e reconstrução das rotas, combinada com método de *shaking* na tentativa de sair de mínimos locais. Para os experimentos, foram utilizadas as instâncias de benchmark de Taillard (1999) e propôs a utilização das soluções de Tarantilis, Kiranoudis e Vassiliadis (2003), Tarantilis, Kiranoudis e Vassiliadis (2004) e Li, Tian e Aneja (2010) como solução inicial e o método proposto para posterior otimização. De acordo com os experimentos, foram atingidas melhorias ao utilizar este método como complemento de outras heurísticas.

O trabalho de Matei et al. (2014) soluciona o problema HFVRP com custo fixo e variável utilizando um GA (*Genetic Algorithm*) em conjunto com uma busca local (*Memetic Algorithm*). A cada iteração do algoritmo são gerados  $n$  indivíduos/soluções e aplicado um método de busca local VND (*Variable Neighborhood Descent*) contendo

6 heurísticas (2-Opt, 3-Opt, OR-Opt, 1-0 Exchange, 1-1 Exchange e Relocate). Os experimentos foram realizados com as instâncias de Taillard (1999) e o método atingiu os melhores resultados frente a literatura de 7 das 8 instâncias existentes.

O trabalho de Wu et al. (2017) utiliza um iALNS (*Improved Adaptive Large Neighborhood Search*) para a solução do HFVRP com custo fixo e variável. Uma solução inicial baseada em heurística gulosa é gerada e a partir dela são realizados mecanismos iterativos de remoção e inserção dos clientes da solução de maneira probabilística. Na fase de remoção é usado um método de clusterização DBSCAN com o intuito de remover clientes do mesmo *cluster* e aumentar a chance de haver mudanças na solução na fase de inserção, já que remover clientes de *clusters* distintos pode levar a realocação destes clientes nas mesmas posições originais. Os experimentos foram realizados utilizando instâncias de Taillard (1999) e os resultados foram considerados satisfatórios, em que 7 das 8 instâncias atingiram os melhores resultados da literatura até então.

O trabalho de Derbel, Jarboui e Bhiri (2019) soluciona o problema HFVRP com custo fixo e variável utilizando um SGVNS (*Skewed Generalized Variable Neighborhood Search*). Utiliza 14 operadores de vizinhança para as alterações da solução, a partir de uma solução inicial. O método proposto permite que soluções piores sejam aceitas com um limite aceitável de piora. Também é utilizada uma função de penalidade para a extração da carga dos veículos, já que permite soluções não factíveis. Os resultados foram satisfatórios para as instâncias de Taillard (1999) e Brandão (2011), em que foram melhorados uma série de resultados da literatura.

O problema de roteamento de veículos com frota heterogênea limitada e com janela de tempo (HFVRPTW) foi solucionado pela primeira vez por Paraskevopoulos et al. (2008), o qual também considera frota ilimitada em sua solução e apenas custo fixo e distância total em sua função objetivo. O método utilizado para solucionar esse problema foi o ReVNTS (*Reactive Variable Neighborhood Tabu Search*). Uma solução semi-paralela é gerada a fim de evitar alocar a maior demanda nos maiores veículos inicialmente, de maneira a considerar todos os tipos de veículos simultaneamente ao inserir os clientes em rotas. A partir dessa solução inicial, são aplicados movimentos de vizinhança (*shaking*) e operações de busca local para intensificar a melhoria da solução. Neste trabalho foram propostas novas instâncias de *benchmark* para o problema de VRP com frota heterogênea (limitada ou ilimitada) e janela de tempo, as quais foram amplamente utilizadas ao longo da literatura.

O trabalho de Jiang et al. (2014) soluciona o problema HFVRPTW com custo fixo e variável em seu modelo por meio de uma TS (*Tabu Search*) com uso de *Holding List* (HL) inspirado no trabalho de Lau, Sim e Teo (2003). O método consiste em operações de migração e troca da/para a HL ao passo que aumenta gradativamente o

número de veículos da solução. Em sua função objetivo hierárquica prioriza o número de atendimento como principal objetivo, seguido pelo número de veículos utilizados e por último o custo total. Neste trabalho foram definidas novas instâncias de *benchmark* a partir das instâncias de Solomon (1987) que contemplam frota heterogênea com custo fixo e variável e janela de tempo. Os experimentos mostram resultados promissores para as instâncias propostas.

O trabalho de Koç et al. (2015) utiliza um HEA (*Hybrid Evolutionary Algorithm*) para a solução do HFVRPTW com apenas custo fixo e distância/tempo percorrido. Neste trabalho também são considerados cenários com frota ilimitada. O método de solução é definido por um ALNS com operadores evolutivos (mutação, crossover e seleção). Os experimentos foram realizados para cenários com frota limitada e ilimitada utilizando instâncias definidas em Paraskevopoulos et al. (2008). Os resultados obtidos atingiram melhorias frente aos resultados precedentes em 14 das 24 instâncias de teste.

O trabalho de Fachini e Armentano (2020) solucionou o problema HFVRPTW utilizando uma decomposição do problema em subproblemas independentes de atribuição e TSP com janela de tempo. Esta decomposição foi denominada de LBBD (*Logic-based Benders Decomposition*). São utilizados métodos exatos e heurísticas construtivas e de melhoria para solucionar o problema proposto. Os experimentos foram realizados frente às instâncias de Paraskevopoulos et al. (2008) e o método atingiu resultados promissores apesar de não ter gerado nenhuma melhoria significativa comparado com a literatura existente.

O trabalho de Molina, Salmeron e Eguia (2020) soluciona uma versão do problema VRPTW com recursos limitados, incluindo o tamanho da frota de veículos (HFVRPTW). O objetivo hierárquico definido para o problema é maximizar os atendimentos, minimizar o número de veículos utilizados e minimizar a distância total percorrida somada aos custos fixos de utilização dos veículos. O método proposto consiste em um algoritmo híbrido entre um VND (*Variable Neighborhood Descent*) com uma TS (*Tabu Search*). Um conjunto de soluções iniciais construídas de maneira semi-paralela inspirada em Paraskevopoulos et al. (2008) são geradas e na sequência o método VNDTS é utilizado para explicação de vizinhança para cada solução inicial criada. Os experimentos foram realizados utilizando as instâncias de *benchmark* definidas por Paraskevopoulos et al. (2008), para as quais foram atingidos os melhores resultados para 13 das 24 instâncias. Além disso, foram gerados experimentos frente às instâncias definidas por Jiang et al. (2014), em que 44 das 56 instâncias obtiveram resultados superiores aos da literatura precedente, o que demonstra que o método proposto obteve resultados satisfatórios.

O trabalho de Molina et al. (2020) utiliza um algoritmo híbrido entre o ACS

(*Ant Colony System*) com um VNTS (*Variable Neighborhood Tabu Search*) para o problema HFVRPTW com custo fixo e variável. O método proposto inicia com uma solução inicial semi-paralela inspirada por Paraskevopoulos et al. (2008). O ACS assume o processo de busca gerando novas soluções a cada iteração e um processo de busca local VNTS é utilizado para intensificar a busca utilizando 7 estruturas de vizinhança. Os experimentos foram realizados frente às instâncias definidas por Paraskevopoulos et al. (2008) em que foram geradas 10 melhorias em relação aos resultados da literatura. Experimentos para as instâncias de Jiang et al. (2014) foram gerados o que levou a 55 melhores resultados de 56 instâncias existentes.

O trabalho de Tang, Zhang e Pan (2010) foi o primeiro registrado na literatura a solucionar o problema de roteamento de veículos considerando custos de carga (VRPLC), o qual representa o custo impactado pelo transporte de um determinado volume de carga ao longo dos trajetos percorridos, custo esse negligenciado pelas versões clássicas de VRP. Um coeficiente que quantifica o custo por unidade de peso por unidade de distância percorrida é definido como o custo variável responsável por representar o impacto do peso transportado no custo total. Um SSA (*Scatter Search Algorithm*) foi o método proposto para a solução do problema. O autor utilizou um conjunto de instâncias existentes para o VRP e adicionou o coeficiente de carga para demonstrar o impacto positivo no custo ao considerar este coeficiente no modelo. O coeficiente de carga foi definido com base em cenários locais de transporte chinês. Os experimentos demonstraram que o uso do custo de carga torna o modelo mais razoável em comparação com o VRP já que se aproxima de cenários reais de roteamento e ainda indica que o uso do custo de carga leva a uma redução do custo total da operação.

O trabalho de Zhang, Tang e Fung (2011) buscou solucionar um MDVRPWRC (*Multi-depot Vehicle Routing Problem with Weight-related cost*), uma variação da nomenclatura, porém considerando o mesmo custo de carga abordado pelo VRPLC em um cenário de múltiplos depósitos. Os autores propuseram um algoritmo SS (*Scatter Search*) para solucionar o problema. Os experimentos foram realizados com diversos cenários existentes na literatura e uma análise aprofundada sobre o efeito do uso do coeficiente de peso, também definido segundo cenários locais de transporte, revelou que o mesmo leva a um aumento da eficiência do método em encontrar soluções de menor custo.

O trabalho de Kuo e Wang (2012) soluciona o MDVRPLC (*Multi-depot Vehicle Routing Problem with Loading Cost*), em que são considerados o custo variável por unidade de distância, custo fixo de utilização do veículo e o custo variável por unidade de peso transportado por unidade de distância percorrida entre os clientes, bem como definido por Tang, Zhang e Pan (2010). Os autores utilizam um VNS (*Variable Neigh-*

*borhood Search*) para solucionar o problema. Para a realização dos experimentos são utilizadas instâncias para o problema MDVRP (*Multi-depot Vehicle Routing Problem*) e utilizadas os coeficientes de custo propostos em Tang, Zhang e Pan (2010). Os resultados demonstraram que todas as 23 instâncias utilizadas obtiveram melhora no custo total quando se utiliza o coeficiente de impacto do peso na função objetivo.

O trabalho de Tang et al. (2013) propôs uma generalização do problema VRPLC e uma nova terminologia: WVRP (*Weighted Vehicle Routing Problem*), que pode englobar diversas ponderações e priorizações dentro de modelos de VRP através de coeficientes de multiplicação na função objetivo. O principal propósito é representar o custo de carga de forma a se aproximar de cenários reais, bem como no VRPLC. Mais precisamente o trabalho soluciona uma versão do problema WVRP com entregas fracionadas *split delivery* por meio de um algoritmo baseado em colônia de formigas MMAS (*Max-Min Ant System*). Bem como os autores precedentes, os experimentos se concentraram na comparação entre os resultados obtidos sem considerar o coeficiente de custo de carga na função objetivo *versus* os resultados das mesmas instâncias, porém com o uso do coeficiente de carga. Com os resultados foram observadas melhorias na função de custo ao utilizar custo de carga para este cenário de VRP com entregas fracionadas.

O trabalho de Tang et al. (2014) utilizou um algoritmo híbrido entre o MMAS (*Max-Min Ant System*) e um *Beam Search* para a solução do WVRP. Neste trabalho também propôs novas instâncias de experimentos e demonstrou com os resultados o impacto positivo na função objetivo (custo) pela utilização do custo extra relacionado a carga na modelagem do problema. Já no trabalho de Wang, Shao e Tang (2021) é utilizado o algoritmo ILS *Iterative Local-Search* para solucionar o mesmo problema (WVRP). São utilizados 7 operadores de vizinhança para a realização da busca e ainda possui mecanismos de perturbação da solução na tentativa de sair de mínimos locais. Para os experimentos foram realizadas comparação do método proposto com resultados presentes na literatura através do trabalho de Tang et al. (2014). Os resultados demonstraram que o método proposto obteve melhoria de desempenho em 61 das 63 instâncias de teste utilizadas.

### 2.7.1 Considerações sobre os trabalhos relacionados

Dado o levantamento dos trabalhos relacionados, a Tabela 2 relaciona os trabalhos descritos nesta seção, bem como o problema tratado e o método de solução. Como pode ser observado, percebe-se a existência de um conjunto considerável de trabalhos relacionados à frota heterogênea limitada, janela de tempo e custo de carga. Portanto, é possível observar que há 1 trabalho relacionado ao VRP com frota homogênea limitada, 14 trabalhos referentes ao problema HFVRP (frota heterogênea limi-

tada), 6 trabalhos referentes ao HFVRPTW (frota heterogênea limitada com janela de tempo) e 6 trabalhos relacionados ao VRPLC ou WVRP (custo de carga), totalizando 27 trabalhos relacionados ao trabalho em tela.

Segundo a Tabela 2, não foram identificados trabalhos que contemplam a junção das características de frota heterogênea limitada, janela de tempo e custo de carga. Assim, o presente trabalho busca modelar o problema de VRP que contenha estas características e propor um método de solução capaz de solucioná-lo.

Tabela 2 – Trabalhos Relacionados

Referência	Solução	Frota Limitada	Frota Heterogênea	Janela de Tempo	Custo de Carga
Taillard (1999)	CG	X	X		
Prins (2002)	IH-LS	X	X		
Lau, Sim e Teo (2003)	TS	X			
Tarantilis, Kiranoudis e Vassiliadis (2003)	LBTA	X	X		
Tarantilis, Kiranoudis e Vassiliadis (2004)	BATA	X	X		
Li, Golden e Wasil (2007)	HRTR	X	X		
Paraskevopoulos et al. (2008)	ReVNTS	X	X		X
Li, Tian e Aneja (2010)	MAMP-TS	X	X		
Zheng, Lu e Sun (2010)	PTS	X	X		
Brandão (2011)	TS	X	X		
Subramanian et al. (2012)	RVND	X	X		
Li, Leung e Tian (2012)	MAMP-TS	X	X		
Naji-Azimi e Salari (2013)	ILP-BH	X	X		
Matei et al. (2014)	Memetic	X	X		
Wu et al. (2017)	iALNS	X	X		
Derbel, Jarboui e Bhiri (2019)	SGVNS	X	X		
Jiang et al. (2014)	TS	X	X		X
Koç et al. (2015)	HEA	X	X		X
Fachini e Armentano (2020)	LBBD	X	X		X
Molina, Salmeron e Eguia (2020)	ACS-VNTS	X	X		X
Molina et al. (2020)	VNDTS	X	X		X
Tang, Zhang e Pan (2010)	SSA				X
Zhang, Tang e Fung (2011)	SSA				X
Kuo e Wang (2012)	VNS				X
Tang et al. (2013)	MMAS				X
Tang et al. (2014)	MMAS-BS				X
Wang, Shao e Tang (2021)	ILS				X
Trabalho em Tela	ALNS-LS	X	X	X	X

Os trabalhos mais relevantes para o HFVRP são o trabalho de Taillard (1999), que deu início a esta linha de estudos de VRP com frota heterogênea limitada, e os trabalhos de Li, Golden e Wasil (2007) e Brandão (2011), que obtiveram os resultados mais consistentes até o momento. Para o HFVRPTW, que contempla restrições de janela de tempo e frota heterogênea limitada, Paraskevopoulos et al. (2008) definiu uma solução inicial semi-paralela, que inspirou a construção da solução inicial utilizada no presente trabalho, e introduziu novas instâncias de *benchmark*. O trabalho de Jiang et al. (2014) introduziu novas instâncias de teste para o problema HFVRPTW contendo custo fixo e variável. O trabalho de Molina et al. (2020) é o mais recente e com resultados mais consistentes acerca do HFVRPTW.

Para o problema de VRP com custo de carga (VRPLC ou WVRP) os principais trabalhos são de Tang, Zhang e Pan (2010), o qual foi o primeiro a formalizar o VRPLC, e o de Tang et al. (2013), que introduziu uma generalização ao problema (WVRP). O trabalho de Wang, Shao e Tang (2021) é considerado o mais recente e apresentou os resultados mais relevantes até então sobre esse tema, dado que apresentou melhorias nos resultados presentes na literatura. Contudo, nenhum trabalho identificado acerca do custo de carga, considera uma frota heterogênea de veículos e janela de tempo para os clientes, e sim outras características de roteamento como múltiplos depósitos e entregas fracionadas, não sendo possível a comparação com os mesmos devido a distância perante ao escopo dos trabalhos. Além disso, estes trabalho apresentam definição de coeficiente de custos de carga baseados em cenários locais de transporte para os experimentos.

Acerca dos métodos de solução apresentados na Tabela 2, os principais métodos são aqueles que contêm solução única ao longo do processo de busca, apesar de algumas exceções como MMAS, ACS-VNTS e Algoritmos Evolutivos, os quais operam com um conjunto de soluções a cada iteração. Dada a ausência de trabalho com as mesmas características na literatura, o presente trabalho propõe um algoritmo ALNS adaptado para a solução do problema HFVRPTW-LC, que contempla frota heterogênea limitada, janela de tempo e custo de carga, dado que este algoritmo se apresenta como uma das meta-heurísticas mais robustas e com melhor desempenho presente na literatura para outros tipos de problemas de roteamento, como demonstrados nos trabalho de Ropke e Pisinger (2006), Schmitt (2020), Gunawan et al. (2020), Mehdi, Abdelmoutalib e Imad (2021), Bayer (2021) e Wen et al. (2022). Além disso, é proposto um estudo amplo para a definição dos coeficientes de custo de carga para um *benchmark* específico.

### 3 SOLUÇÃO PROPOSTA

Este trabalho tem por objetivo propor um método de solução baseado em um algoritmo meta-heurístico de Busca Larga Adaptativa em Vizinhança (*Adaptive Large Neighborhood Search - ALNS*) para solucionar o Problema de Roteamento de Veículos com Frota Heterogênea Fixa (Limitada), Janela de Tempo e Custo de Carga (HFVRPTW-LC). Para este tipo de problema, busca-se minimizar os custos totais envolvidos, tais como, custo variável por distância percorrida, custo variável de carga transportada e custo fixo de utilização dos veículos. Este capítulo visa apresentar a modelagem do problema e o método de solução proposto.

De acordo com o levantamento dos trabalhos relacionados na Seção 2.7, não observou-se a presença de um trabalho que propõe a solução para a combinação de características propostas no presente trabalho (frota heterogênea fixa/limitada, janela de tempo e custo de carga). Dada a complexidade dos problemas de VRP, meta-heurísticas de melhoramento de solução única se apresentam como uma abordagem de solução adequada para o problema proposto, conforme abordado na Seção 2.6. Desse modo, optou-se pela utilização do ALNS como abordagem de solução.

#### 3.1 FORMALIZAÇÃO MATEMÁTICA DO PROBLEMA

O problema HFVRPTW-LC pode ser definido como um grafo  $G = (V, A)$ , em que  $A$  é o conjunto de arestas/arcos e  $V$  é o conjunto de  $n$  vértices, dos quais o vértice 0 é o depósito e os demais são os clientes. Cada cliente está associado a uma demanda não negativa  $q_i$  e uma janela de tempo  $[e_i, l_i]$  (período para o início do atendimento) e um tempo de serviço  $s_i$ . A janela de tempo do depósito é dada por  $[e_0, l_0]$ , em que  $e_0$  representa o momento em que os veículos podem começar a operar e  $l_0$  indica o limite de tempo em que todos os veículos devem estar de volta ao depósito. A cada arco  $(i, j)$  está associada uma distância  $d_{ij}$  e uma duração  $t_{ij}$  para o trajeto entre os vértices  $i$  e  $j$ .

A frota heterogênea de veículos  $K = \{1, \dots, k\}$  é dividida em conjuntos de veículos homogêneos, sendo  $K = K^1 \cup K^2 \cup \dots \cup K^{|P|}$ . Cada tipo de veículo  $p = 1, \dots, |P|$  tem uma capacidade  $C^p$ , um custo fixo de utilização  $f^p$ , um custo variável  $r^p$  (em geral por unidade de distância) e um custo variável  $w^p$  que representa o custo por unidade de peso carregado por unidade de distância percorrida pelo veículo do tipo  $p$ . Cada tipo de veículo  $p$  também possui um número limitado de veículos com no máximo  $|K^p|$  veículos. O modelo matemático contém variáveis binárias  $x_{ijk} \in \{0, 1\}$ , que indicam se o veículo  $k$  está trafegando entre os clientes  $i$  e  $j$ ; variáveis reais de

fluxo de carga  $y_{ijk}$ , que representam quanto de carga/peso o veículo  $k$  está carregando entre os clientes  $i$  e  $j$ ; e variáveis reais  $b_i$ , as quais representam o instante em que o cliente  $i$  inicia seu atendimento.

O problema tem objetivo de minimizar a função objetivo composta pelos custos relacionados à distância percorrida, peso transportado e frota utilizada, seguindo a formulação matemática a seguir (3.1):

$$\text{Minimizar} \quad \sum_{i=0}^n \sum_{\substack{j=0 \\ j \neq i}}^n d_{ij} \sum_{k=1}^K x_{ijk} (r^p + w^p y_{ijk}) + f^p \sum_{j=1}^n \sum_{k=1}^K x_{0jk} + M \left[ n - \sum_{i=0}^n \sum_{\substack{j=0 \\ j \neq i}}^n \sum_{k=1}^K x_{ijk} \right] \quad (3.1a)$$

$$\text{sujeito a} \quad \sum_{k=1}^K \sum_{j=1}^n x_{0jk} \leq K \quad (3.1b)$$

$$\sum_{j=1}^n x_{0jk} \leq 1, \quad k = 1, \dots, K \quad (3.1c)$$

$$\sum_{j=1}^n x_{j0k} \leq 1, \quad k = 1, \dots, K \quad (3.1d)$$

$$\sum_{j=1}^n x_{0jk} = \sum_{j=1}^n x_{j0k}, \quad k = 1, \dots, K \quad (3.1e)$$

$$\sum_{k=1}^K \sum_{j=0}^n x_{ijk} \leq 1, \quad i = 1, \dots, n \quad (3.1f)$$

$$\sum_{j=0}^n x_{ijk} - \sum_{j=0}^n x_{jik} = 0, \quad k = 1, \dots, K, \quad i = 1, \dots, n \quad (3.1g)$$

$$\sum_{k=1}^K \sum_{i \in S} \sum_{j \in S} x_{ijk} \leq |S| - v(S), \quad \forall S \subseteq V \setminus \{0\}, \quad |S| \geq 2 \quad (3.1h)$$

$$\sum_{j=1}^n q_j \sum_{\substack{j=0 \\ j \neq i}}^n x_{ijk} \leq C^p, \quad k = 1, \dots, K \quad (3.1i)$$

$$\sum_{k=1}^K \sum_{\substack{i=0 \\ i \neq j}}^n x_{ijk} (b_i + s_i + t_{ij}) \leq b_j, \quad j = 1, \dots, n \quad (3.1j)$$

$$e_i \leq b_i \leq l_i, \quad i = 0, \dots, n \quad (3.1k)$$

$$\sum_{k=1}^K \sum_{\substack{j=0 \\ j \neq i}}^n (x_{jik} y_{jik} - x_{ijk} y_{ijk}) = q_i \quad i = 1, \dots, n \quad (3.1l)$$

$$0 \leq y_{ijk} \leq C^p x_{ijk}, \quad i, j = 0, 1, \dots, n \quad (3.1m)$$

A Equação (3.1a) apresenta a função objetivo, composta pela soma dos custos variáveis referente à distância percorrida e ao peso transportado, custos fixos dos veículos utilizados e também uma penalidade para os não atendimentos, uma vez que

o problema abordado neste trabalho permite o não atendimento de clientes. Portanto, a modelagem considera um coeficiente de valor elevado  $M$  como fator multiplicador a quantidade de clientes não atendidos para penalizar a solução. Apesar desta penalidade estar presente neste modelo matemático, ela não é efetivamente utilizada no método de solução proposto, pois optou-se pela utilização de uma função objetivo hierárquica, apresentada na Seção 3.2.2. Desse modo, a penalidade cumpre um papel de incentivo para os atendimentos no modelo matemático, dado que não é viável trazer a noção hierárquica para esta função objetivo.

A restrição (3.1b) garante que no máximo  $K$  veículos sairão do depósito e as restrições (3.1c) e (3.1d) permitem que nem todos os veículos sejam usados. A restrição (3.1e) garante que um determinado veículo volte para o depósito ao final da rota. A restrição (3.1f) garante que cada cliente seja visitado no máximo uma vez. A restrição (3.1g) assegura que o veículo chegue e saia do mesmo cliente. A restrição (3.1h) evita que sejam formados subciclos que não incluam o depósito. A restrição (3.1i) garante que uma rota não exceda a capacidade do veículo. As restrições (3.1j) e (3.1k) garantem a factibilidade da janela de tempo. A restrição (3.1l) assegura que a diferença das cargas transportadas pelo veículo antes e depois do atendimento do cliente  $i$  seja igual a demanda deste cliente ( $q_i$ ). A restrição (3.1m) garante que a carga transportada  $y_{ijk}$  entre os clientes não supere a capacidade dos veículos.

## 3.2 MÉTODO PROPOSTO

Nesta seção é apresentado o modelo de solução proposto para o problema HFVRPTW-LC. A meta-heurística ALNS é utilizada para a solução do problema. O problema em questão tem o objetivo de minimizar custos envolvidos nas rotas, tais como custo variável por distância, custo de carga transportada e custo fixo de utilização dos veículos. Além disso, o problema apresenta um cenário de recursos limitados (frota limitada), o que traz a necessidade de lidar com situações onde os recursos disponíveis podem não ser suficientes para atender todas as demandas. Portanto, neste contexto podem haver clientes não atendidos devido a possíveis indisponibilidades de veículos e janelas de tempo não favoráveis para o atendimento completo da operação.

### 3.2.1 Representação da Solução

A representação para uma solução do problema utilizada neste trabalho é dada por um conjunto de listas encadeadas compostas por números inteiros que representam a sequência de vértices visitados de cada rota de uma roteirização. Assim, uma solução representa um conjunto de rotas de um roteamento de veículos. Como ilustrado na Figura 10, todas as rotas contidas na solução saem do depósito (vértice 0), visita os clientes (demais vértices) e retorna para o depósito ao final da rota. Neste

exemplo, os clientes 1 - 11 são atendidos por 3 rotas utilizando veículos distintos. Nesta modelagem não é permitida qualquer repetição entre vértices em uma mesma rota, com exceção do vértice 0 que indica o depósito.

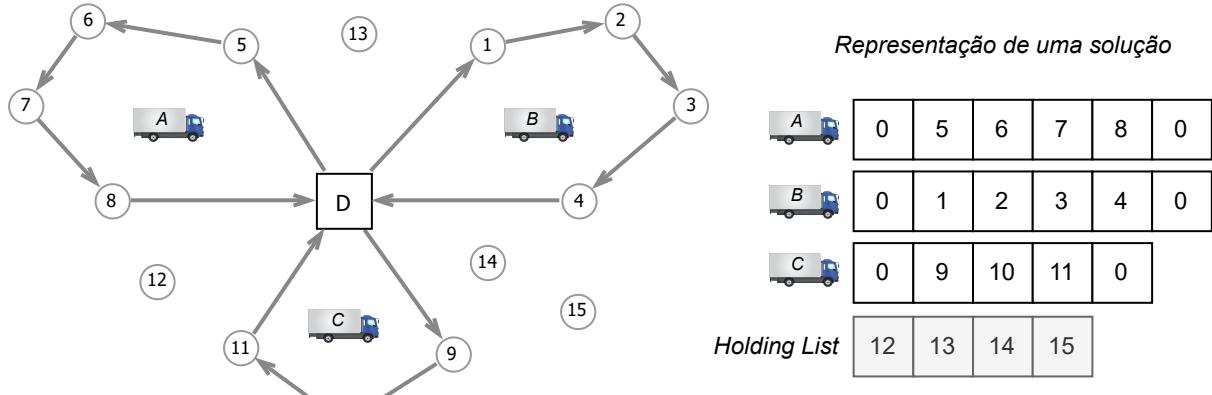


Figura 10 – Representação de uma solução do método proposto

Fonte: Próprio Autor

Para lidar de forma mais robusta com os eventuais não atendimentos, a representação da solução utilizada neste trabalho contém além das rotas (veículo utilizado e a sequência de visitas), uma lista de clientes não atendidos denominada de *Holding List* (HL). A HL foi proposta por Lau, Sim e Teo (2003) com o intuito de tornar factível uma solução com demandas/clientes não atendidas e como uma forma de navegar pelo espaço de busca com maior facilidade, já que representa um ponto de troca entre os clientes em rota (atendidos) e os clientes da HL (não atendidos). Isso aumenta as possibilidades de alterações nas soluções, o que faz com que a solução transite por regiões antes dadas como não factíveis por não atender todos os clientes.

Apesar de permitir o não atendimento de clientes, causado pelo limite de frota imposto pelo problema, no restante do método proposto todas as operações para criar ou modificar a solução são realizadas apenas se estes movimentos gerarem soluções factíveis em relação as demais restrições impostas, como o limite de capacidade dos veículos, limite de quantidade de veículos, janela de tempo e não repetição de atendimentos.

### 3.2.2 Função Objetivo Hierárquica

Em geral, problemas de VRP clássicos com frota ilimitada (infinita) e função objetivo de minimizar apenas distância percorrida ou duração da operação, ao serem solucionados por métodos heurísticos/meta-heurísticos, devem possuir algum critério para lidar com o combinação entre a minimização da distância/duração e a minimização da quantidade de rotas/veículos, dado que minimizar a distância pode ocasionar em um aumento de rotas/veículos utilizados. A solução mais comum para este combi-

nação é a utilização de uma função objetivo hierárquica, em que a prioridade é minimizar o número de veículos utilizados e em segundo lugar minimizar a distância/duração total. Outra possibilidade é a utilização de dois processos de otimização interconectados e que compartilham a solução otimizada, em que um processo fica a cargo de minimizar o número de veículos e outro processo de minimizar a distância/duração.

Contudo, no contexto de problemas de otimização de VRP proposto neste trabalho, em que são utilizados custos fixos de utilização dos veículos e custos variáveis relacionados a distância percorrida e carga transportada, a combinação destes custos na função objetivo pode ser considerada suficiente para lidar com os acréscimos de veículos, por meio da penalidade representada pelos custos fixos. Assim, se torna desnecessária a utilização de hierarquia ou meios paralelos para minimização de veículos. Ainda, a restrição de frota limitada não permite que a otimização utilize de forma deliberada a quantidade de veículos que considerar de menor custo, pois há um limite de veículos para cada tipo. Portanto, a função objetivo aplicada neste trabalho é baseada em custos e não apenas em distância/duração e já prevê o impacto da adição de um veículo a mais na solução. Além disso, para o cenário de frota heterogênea, a função objetivo também é capaz de dimensionar o impacto de usar um determinado veículo em detrimento de outro, dado que os custos de cada veículo são diferentes entre si. Assim, a minimização dos veículos passa a ser em função do seu custo e não quantidade, e fica a cargo da função objetivo representar a qualidade da solução.

Apesar da não necessidade da minimização dos veículos, a definição do problema proposto não obriga o atendimento de todos os clientes, dada a frota limitada. Com isso, na solução proposta faz-se necessário a utilização de uma função objetivo hierárquica em que o objetivo principal é maximizar o número de atendimentos (ou minimizar o número de não atendimentos - *Holding List*) e em seguida minimizar o custo total da operação. Caso contrário, o número de atendimento tenderia a zero já que não atender nenhum cliente não traria custos para a operação e não haveriam rotas na solução, sendo assim, não haveria nenhum incentivo para os atendimentos. Desse modo, soluções com maior número de atendimentos serão sempre melhor avaliadas que soluções com menos atendimentos, mesmo que esta última tenha custo total menor, o que possivelmente acontecerá já que menos atendimentos tende a resultar em menor custo.

### 3.2.3 Modelagem da Solução

A Figura 11 apresenta o diagrama de atividades do método proposto, composto pelo pré-processamento, onde acontece a geração da solução inicial, seguido pelo ciclo de otimização ALNS. A literatura apresenta diversos métodos para a criação da solução inicial que utilizam soluções gulosas com alguma espécie de ordenação

entre os veículos e as demandas. Por exemplo, alocar a maior demanda no veículo com maior espaço vazio ou criar rotas na ordem do maior para o menor veículo. Este tipo de método pode gerar uma solução menos eficiente e com pouca capacidade de melhoria por meio de um método de otimização. Logo, este trabalho visa utilizar um método alternativo para a geração da solução inicial, inspirada no método de construção semi-paralelo apresentado por Paraskevopoulos et al. (2008) e apresentada em detalhes na Seção 3.3.1.

No pré-processamento, para a criação da solução inicial, a cada passo é gerado um conjunto de rotas, uma para cada tipo de veículo disponível. O processo construtivo de cada rota é guiado por uma função gulosa e leva em consideração todos os clientes ainda não atendidos, incluindo clientes já alocados em outros veículos, podendo haver repetições de atendimento entre rotas. Dentre as rotas criadas, a mais eficiente é escolhida para compor a solução inicial definitiva. O processo se repete até que todos os clientes estejam atendidos ou não haja mais veículos disponíveis. Caso haja clientes não atendidos, estes são colocados na *Holding List (HL)*. Este método não utiliza priorização de veículos ou procedimentos semelhantes, e sim uma exploração por possibilidades de rotas com veículos diferentes e a escolha da rota mais eficiente a cada passo. Isso possibilita que a solução inicial seja mais apropriada para servir de base para o aprimoramento posterior, realizado pelo ciclo de otimização ALNS.

Após a geração da solução inicial, ela assume a posição de melhor solução corrente e melhor solução global, e é iniciado o ciclo de otimização ALNS. Dentro deste ciclo, são atualizados os valores dos pesos dos operadores e parâmetros da otimização. Na sequência, o critério de parada é verificado e, caso não seja satisfeito, inicia uma nova iteração do ciclo de otimização. A primeira operação interna deste ciclo é a seleção de um operador de remoção, um operador de inserção e um operador de ruído. A dinâmica dos operadores utilizados pelo ALNS será apresentada com mais detalhes na Seção 3.3.2. Estes operadores são aplicados na solução corrente, gerando uma nova solução.

A partir desta nova solução, é realizada uma sequência de verificações em relação a sua qualidade, com base na função objetivo hierárquica definida anteriormente. Se a nova solução for melhor que solução global, um processo de busca local é aplicado afim de melhorar a solução encontrada, e assim as soluções corrente e global são atualizadas. Caso contrário, se a nova solução for melhor apenas que a solução corrente, esta é atualizada. Caso não haja nenhuma melhora, é aplicado um critério de aceitação inspirado na meta-heurística *Simulated Annealing*, adaptado de Bayer (2021), para definir se mesmo com a piora da qualidade da solução gerada, ela será aceita. Esse mecanismo permite que o processo de busca fuja de mínimos

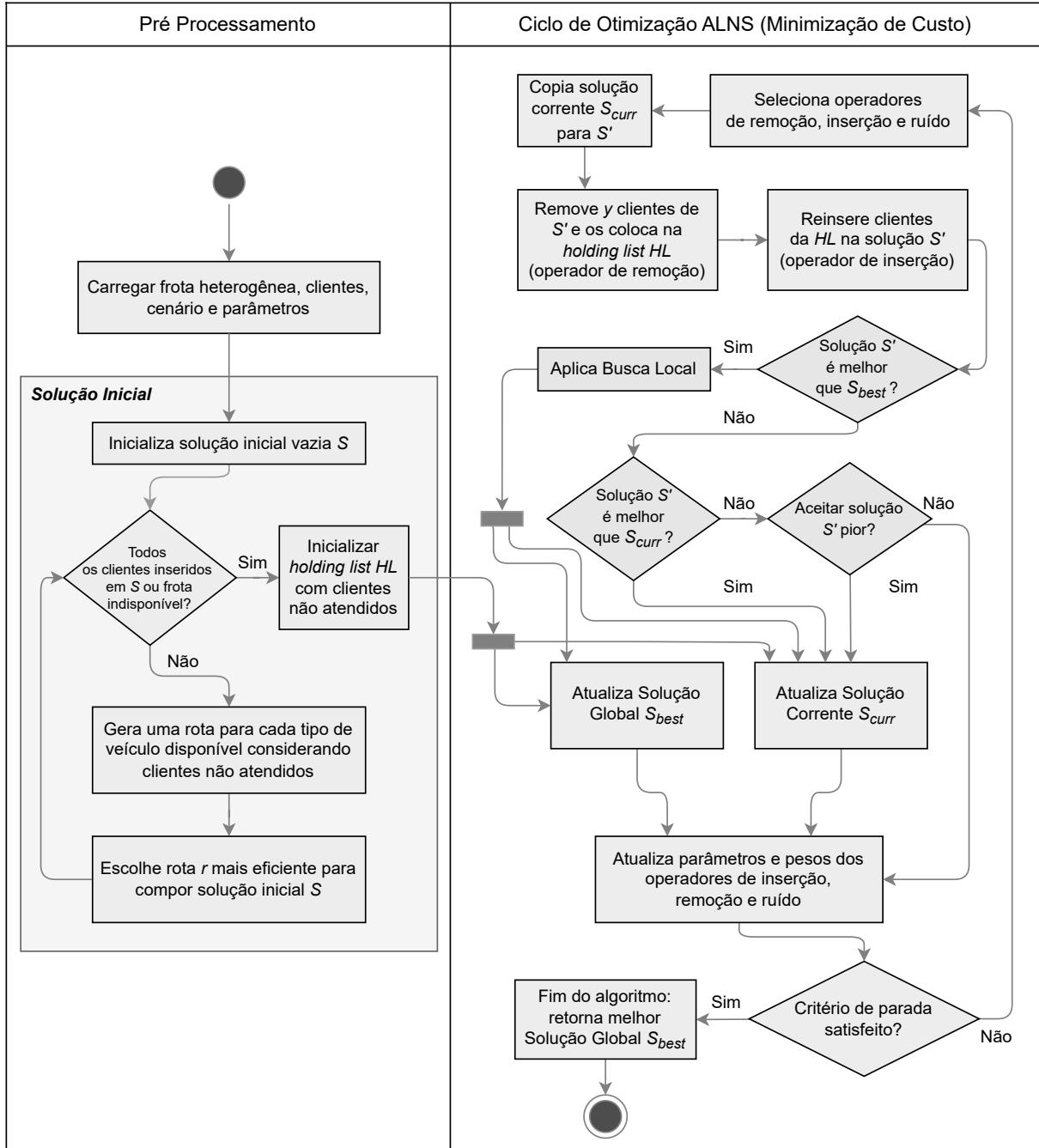


Figura 11 – Método proposto para solução do HFVRPTW-LC com ALNS

Fonte: Próprio Autor

lorais em determinados momentos. Caso a solução seja aceita, mesmo com piora, esta assume a posição da solução corrente. Com isso, são atualizados os parâmetros e pesos dos operadores com base na qualidade da solução gerada. Por fim, o critério de parada é verificado e, caso não seja satisfeito, mais uma iteração é gerada. Caso seja satisfeito, o algoritmo finaliza e a melhor solução global é retornada.

Dado o apanhado geral do modelo, algumas atividades do ciclo de otimização necessitam ser destacadas:

- **Critério de Parada:** neste trabalho o critério de parada definido é o número de iterações, se este número for atingido o algoritmo é finalizado.
- **Seleção de Operadores:** um conjunto de operadores são disponibilizados para realizar operações a partir da solução corrente. Dentre eles, estão quatro operadores de remoção, sendo eles o *Random Removal*, o *Worst Removal*, *Shaw Removal* e *Route Removal*, e cinco operadores de inserção, sendo eles o *Greedy Insertion* e quatro variações do *Regre-k Insertion*, em que  $k = \{2, 3, 4, m\}$ , dado que  $m$  é o número de rotas da solução. A cada início de iteração (ciclo de otimização) é selecionado um operador de remoção e um operador de inserção de maneira probabilística, segundo critério de seleção por roleta (detalhado na Seção 3.3.2), utilizado por Ropke e Pisinger (2006) como meio de seleção, o qual utiliza os pesos históricos dos operadores e o número de utilizações deles para medir a eficiência de cada operador ao longo das iterações. O operador mais eficiente possui maior probabilidade de ser escolhido pela roleta. É aplicada uma seleção para cada tipo de operador (remoção e inserção). Os operadores de remoção e inserção possuem mecanismos internos de geração de ruídos (descritos em mais detalhes nas Seções 3.3.2.1-3.3.2.6), com propósito de gerar variações mais drásticas na solução e evitar operações gulosas de forma constante. O uso ou não destes ruídos é determinado probabilisticamente também por uma seleção por roleta, em que é medida a eficiência histórica do uso de ruído, ou seja, se utilizar ruído se mostrou eficiente ao longo das iterações, maior a probabilidade de utilizar ruído na iteração atual. Assim, nem toda iteração do ciclo de otimização faz uso de ruído nos operadores de remoção e inserção selecionados, apenas quando escolhido pela roleta.
- **Aplicação dos operadores:** dados operadores selecionados, o operador de remoção escolhido é utilizado de modo que um conjunto de clientes atendidos seja removido e transferido para a HL. Ao remover os clientes, rotas vazias podem surgir. Se isso ocorrer, os veículos utilizados nestas rotas voltam para o conjunto de veículos ainda não utilizados. Em seguida, o operador de inserção tenta inserir de maneira factível os clientes da HL nas rotas existentes, ou seja, rotas remanescentes do processo de remoção. Após a aplicação do operador de inserção nas rotas existentes, caso haja clientes não atendidos de forma factível, é executado o método de construção semi-paralelo de Paraskevopoulos et al. (2008), utilizado na solução inicial, para criar novas rotas para atendê-los, considerando a frota livre disponível. O uso ou não de ruído, determinado no passo anterior, tem efeito sobre o operador de remoção e inserção utilizados. Ao final deste passo, uma nova solução é gerada.
- **Verificação de Melhora:** este é um ponto chave para a otimização, já que são

os critérios avaliados que determinam os rumos da convergência do algoritmo. Neste passo, é primeiramente verificado se a nova solução é melhor que a solução global. Caso isso seja verdade, é aplicada uma Busca Local à solução local e ela é atribuída como solução corrente e solução global. Se ela for melhor apenas que a solução corrente, então apenas a solução corrente é atualizada. Em caso de nenhuma melhora, um critério de aceitação de piora é verificado (próxima etapa). No processo de verificação de melhoria também são atualizadas as pontuações dos operadores, as quais são valores atrelados aos operadores de maneira cumulativa que indicam o quanto eficiente o operador foi na geração de uma nova solução. Assim, se a nova solução gerada for melhor que a solução global, os operadores recebem a pontuação mais alta, caso contrário, recebem pontuação mais baixa. A pontuação é a informação central para a calibragem dos operadores ao final da iteração.

- **Aceitação de Piora:** o critério de aceitação de uma solução pior que a solução corrente é determinado pela função de aceitação probabilística, inspirada na meta-heurística *Simulated Annealing*, adaptada da solução utilizada por Bayer (2021) para o ALNS, que utiliza a temperatura como parâmetro de controle de aceitação. Esta temperatura inicia elevada e diminui ao longo das iterações. De modo geral, a temperatura elevada aumenta a probabilidade de aceitação de uma solução pior (diversificação), enquanto que a temperatura mais baixa diminui essa probabilidade (intensificação).
- **Busca Local:** caso a solução gerada for melhor que a solução global, a mesma sofre o processo de intensificação por meio da aplicação de heurísticas de busca local. Este processo envolve um conjunto de operadores, tais como *Cross*, *Exchange* e *Relocate* nesta sequência. Estes operadores são aplicados até que não haja mais melhoria. O processo de busca local é descrito na Seção 3.3.3.
- **Atualização de pesos de operadores e outros parâmetros:** ao final da iteração a pontuação acumulada e o número de utilizações de cada operador são utilizados para atualizar seus pesos. Esse processo é detalhado na Seção 3.3.2. Neste trabalho, esta etapa é executada apenas a cada 100 iterações. Além disso, a cada iteração é realizado o decaimento do parâmetro de temperatura segundo Bayer (2021), utilizada no critério de aceitação de piores soluções. Após o término desta etapa é verificado o critério de parada.

Nesta seção foi apresentada a modelagem da solução, com a visão geral da função objetivo hierárquica utilizada, a maneira como ela lida com os potenciais não atendimentos acarretados pelo limite de frota e, por fim, as etapas principais do método de solução. Na seção subsequente o algoritmo utilizado será apresentado com

mais detalhes. Embora a modelagem da solução tenha sido definida com base em características fundamentalmente específicas para o problema proposto, foram utilizados outros trabalhos como inspiração, principalmente os de Ropke e Pisinger (2006), Pisinger e Ropke (2007), Schmitt (2020) e Bayer (2021) os quais utilizam o algoritmo ALNS para outras variações de problemas de roteamento.

### 3.3 MODELAGEM DO ALGORITMO

Nesta seção são apresentados os algoritmos utilizados na geração da solução inicial (Seção 3.3.1), na implementação do método de otimização baseado no ALNS, incluindo os operadores de remoção e inserção (Seção 3.3.2), e na busca local (Seção 3.3.3).

#### 3.3.1 Solução Inicial Semi-Paralela

Neste trabalho a geração da solução inicial é realizada utilizando uma heurística construtiva semi-paralela. Este é um método dito semi-paralelo, pois todos os tipos de veículos disponíveis são considerados simultaneamente ao longo do processo construtivo de alocação de clientes/demandas aos veículos, ao contrário dos métodos mais comuns da literatura para frota heterogênea, em que são realizadas ordenações de veículos e alocações enviesadas, como por exemplo alocar de modo iterativo a maior demanda não atendida no veículo com maior espaço vazio disponível, o que pode causar ineficiência na geração de solução. O Algoritmo 1 representa o método por completo para a geração da solução inicial.

---

#### Algorithm 1 Solução Inicial Semi-Paralela

---

```

1:  $S \leftarrow \emptyset$ 
2:  $N \leftarrow$  todos os clientes não atendidos
3:  $K \leftarrow$  tipos de veículos disponíveis
4: while  $N \neq \emptyset$  OR  $K \neq \emptyset$  do
5:    $N' \leftarrow$  cópia de  $N$ 
6:    $R \leftarrow generateCandidateRoutes(K, N')$  de acordo com o Algoritmo 2
7:    $r \leftarrow$  melhor rota de  $R$  segundo  $ACUT$  (Eq. 3.2)
8:   Insira  $r$  na solução  $S$ 
9:   Remover clientes atendidos em  $r$  do conjunto  $N$ 
10:   $K \leftarrow$  atualiza tipos de veículos disponíveis
11: end while
12: Insira clientes não atendidos na  $HL$  de  $S$ 
13: return  $S$ 

```

---

A partir dos dados de entrada de uma instância de teste contendo as informações do cenário, é inicializada uma solução vazia (linha 1), a lista de clientes não atendidos com todos os clientes do cenário (linha 2) e a lista de todos os tipos de

veículos disponíveis (linha 3). Da linha 4 até 11 é executado o *loop* de construção da solução até que a lista de não atendidos esteja vazia ou não haja mais nenhum tipo de veículo disponível. Na linha 5 é realizada uma cópia  $N'$  da lista de clientes não atendidos e em seguida são geradas possíveis rotas (linha 6) utilizando a lista  $N'$ , o conjunto de tipos de veículos disponíveis  $K$ , de acordo com o Algoritmo 2. A partir das opções de rotas geradas, a rota mais eficiente  $r$  segundo a métrica *ACUT* (Equação 3.2) é escolhida e inserida na solução definitiva  $S$  (linhas 7 e 8). Em seguida, são removidos da lista  $N$  os clientes atendidos pela rota  $r$  a lista de tipos de veículos disponíveis  $K$  é atualizada, removendo o veículo utilizado na rota  $r$  (linhas 9 e 10). Ao final do algoritmo, caso haja clientes ainda não atendidos e a frota de veículos utilizada por completo, os clientes remanescentes são transferidos para a HL da solução  $S$ . A solução  $S$  é retornada ao final do processo.

A métrica *ACUT* (*Average Cost per Unit Transferred*), segundo Paraskevopoulos et al. (2008), define a eficiência de uma determinada rota que utiliza o veículo do tipo  $k$ . O principal intuito é medir o quanto será pago pela unidade de demanda transportada. Assim, a expressão *ACUT*, dada pela Equação 3.2, é definida pelo custo total da rota, composta pelos custos variáveis por distância  $r^k$  e por carga transportada  $w^k$ , e custo fixo  $f^k$  de utilização do veículo sobre o total de demanda transportada  $q_i$  (Equação 3.3). As variáveis  $d_{ij}$  e  $y_{ij}$  são respectivamente a distância percorrida e o peso da carga transportada entre os clientes  $i$  e  $j$ . Assim, quanto menor o valor de *ACUT*, mais eficiente é a rota. Esta métrica foi adaptada neste trabalho para conter as medidas de custos fixos e variáveis.

$$ACUT_k = \frac{\left[ \sum_{i,j \in R} c_{ijk} \right] + f^k}{\sum_{i \in R} q_i} \quad (3.2)$$

$$c_{ijk} = (d_{ij} \times r^k) + (y_{ij} \times w^k) \quad (3.3)$$

Para a geração de possíveis rotas é utilizado o Algoritmo 2. Um conjunto de rotas  $R$  vazio é inicializado (linha 2). Para cada tipo de veículo  $k$  no conjunto  $K$  é gerada uma rota a partir de um conjunto de clientes (linha 3 a 13). Na linha 4 é realizada a cópia  $N''$  da lista de clientes não atendidos de  $N'$ , assim, a cada nova rota é considerado o mesmo conjunto de clientes, podendo haver repetição de atendimentos entre rotas. Com isso, é realizada a construção da uma rota de maneira gulosa segundo a Equação 3.4, em que a inserção factível de menor custo é escolhida para a rota  $r$  até que não tenham mais inserções factíveis (linha 6 a 11). A rota  $r$  é inserida no conjunto  $R$ . Ao final do processo o conjunto  $R$  é retornado. O custo de inserção do cliente  $u$  entre dois clientes adjacentes  $i$  e  $j$  é calculado pela seguinte métrica:

$$C_{ijuk} = (\alpha_1 \times C_{ijuk}^0) + (\alpha_2 \times C_{ijuk}^1) \quad (3.4)$$

$$C_{ijuk}^0 = c_{iuk} + c_{ujk} - c_{ijk} \quad (3.5)$$

$$C_{ijuk}^1 = Q_k - \left( \sum_{i \in R} q_i \right) - q_u \quad (3.6)$$

em que  $C^0$  é a diferença de custo total da rota com a inserção do cliente  $u$  entre  $i$  e  $j$  (custo dado pela Equação 3.3) e  $C^1$  é responsável por medir o espaço livre no veículo  $k$  após a inserção do cliente  $u$  a partir da capacidade  $Q_k$  do veículo e as demandas  $q$  dos clientes, ou seja, é responsável por maximizar o uso dos veículos. Os parâmetros  $\alpha 1$  e  $\alpha 2$  são responsáveis pelo balanceamento dos custos em que  $\alpha 1 + \alpha 2 = 1$ .

---

**Algorithm 2** Geração de Possíveis Rotas

---

**Require:**  $K$ : tipos de veículos disponíveis, e  $N'$ : clientes não atendidos

```

1: function GENERATECANDIDATEROUTES( $K, N'$ )
2:    $R \leftarrow \emptyset$ 
3:   for all  $k$  in  $K$  do
4:      $N'' \leftarrow$  cópia de  $N'$ 
5:      $r \leftarrow$  rota vazia iniciando no depósito
6:     repeat
7:        $c, p \leftarrow$  melhor cliente e posição factível de inserção em  $r$  (Eq. 3.4)
8:       Insira  $c$  na posição  $p$  de  $r$ 
9:       Remova  $c$  de  $N''$ 
10:      until  $c = null$ 
11:      Finaliza rota  $r$  retornando para o depósito
12:      Insira rota  $r$  em  $R$ 
13:   end for
14:   return  $R$ 
15: end function

```

---

### 3.3.2 ALNS

A partir da solução inicial gerada pelo método heurístico semi-paralelo, a meta-heurística ALNS assume o controle da otimização. O Algoritmo 3 apresenta os detalhes deste algoritmo. Conforme introduzido na Seção 2.6, o ALNS, apresentado em (ROPKE; PISINGER, 2006), é um método baseado em destruição e reconstrução da solução a cada iteração, guiados pelo peso histórico de cada operador. A cada iteração é escolhido um operador para remover clientes e outro operador para inseri-los novamente na solução em posições mais eficientes. Além disso, há a possibilidade de uso de ruído dentro de cada operador. Ao longo do processo é permitida a aceitação de piora na solução mediante critério baseado na meta-heurística *Simulated Annealing* e seu controle de temperatura.

O algoritmo ALNS recebe o conjunto de opções de operadores de remoção  $RO = \{Worst, Random, Shaw, Route\}$ , o conjunto de operadores de inserção  $IO = \{Greedy, Regret - 2/3/4/m\}$  e ruído  $NO = \{true, false\}$ . O algoritmo também requer

---

**Algorithm 3** Algoritmo de Busca Larga Adaptativa em Vizinhança (ALNS)
 

---

**Require:** Operadores ( $RO, IO, NO$ ) e Parâmetros ( $t, \theta_1, \theta_2, \theta_3, \kappa$ )

```

1:  $S \leftarrow$  solução inicial semi-paralela (Algoritmo 1)
2:  $S_{best} \leftarrow S$ 
3: Inicializa os pesos dos operadores  $\omega(RO), \omega(IO), \omega(NO)$ 
4: Inicializa as pontuações dos operadores  $\phi(RO), \phi(IO), \phi(NO)$ 
5: Inicializa a contagem dos operadores  $\mu(RO), \mu(IO), \mu(NO)$ 
6:  $itMax \leftarrow$  número máximo de iterações
7:  $it \leftarrow 0$ 
8: while  $it < itMax$  do
9:    $T \leftarrow (1 - it/itMax)^t$ 
10:  ▷ Seleção dos operadores
11:   $q_{rem} \leftarrow$  Seleciona operador de remoção usando  $\omega(RO)$  e Eq. 3.7
12:   $q_{ins} \leftarrow$  Seleciona operador de inserção usando  $\omega(IO)$  e Eq. 3.7
13:   $noise \leftarrow$  Seleciona uso de ruído usando  $\omega(NO)$  e Eq. 3.7
14:  ▷ Aplicação dos operadores
15:   $S' \leftarrow$  cópia de  $S$ 
16:   $y \leftarrow$  número de clientes a serem removidos
17:  Remova  $y$  clientes das rotas usando o operador  $q_{rem}$  e ruído  $noise$ 
18:  Insira os  $y$  clientes removidos na  $HL$  de  $S'$ 
19:  Insira clientes da  $HL$  na rotas de  $S'$  usando o operador  $q_{ins}$  e ruído  $noise$ 
20:  ▷ Minimização
21:  if  $compare(S', S_{best})$  then
22:     $localSearch(S')$  segundo Algoritmo 14
23:     $S_{best} \leftarrow S \leftarrow S'$ 
24:    Incremente as pontuações dos operadores  $\phi(q_{rem}), \phi(q_{ins}), \phi(noise)$  com  $\theta_1$ 
25:  else if  $compare(S', S)$  then
26:     $S \leftarrow S'$ 
27:    Incremente as pontuações dos operadores  $\phi(q_{rem}), \phi(q_{ins}), \phi(noise)$  com  $\theta_2$ 
28:  else if  $accept(S', S, T)$  then
29:     $S \leftarrow S'$ 
30:    Incremente as pontuações dos operadores  $\phi(q_{rem}), \phi(q_{ins}), \phi(noise)$  com  $\theta_3$ 
31:  end if
32:  Incremente as contagens dos operadores  $\mu(q_{rem}), \mu(q_{ins}), \mu(noise)$ 
33:  ▷ Atualização dos parâmetros
34:  if fim do segmento  $\kappa$  de iterações then
35:    Atualize os pesos  $\omega$  dos operadores  $RO, IO, NO$  segundo 3.8
36:    Reinicie as pontuações  $\phi$  dos operadores  $RO, IO, NO$ 
37:    Reinicie as contagens  $\mu$  dos operadores  $RO, IO, NO$ 
38:  end if
39:   $it \leftarrow it + 1$ 
40: end while
41: return  $S_{best}$ 

```

---

o fator  $t$  de resfriamento da temperatura, os valores de incremento da pontuação para os operadores ( $\theta_1, \theta_2, \theta_3$ ) e o tamanho do segmento  $\kappa$ , o qual indica a frequência de atualização dos pesos dos operadores. O primeiro passo é a geração da solução inicial  $S$  pelo método descrito na Seção 3.3.1 (linha 1). A melhor solução  $S_{best}$  recebe a cópia da solução  $S$  (linha 2). Entre as linhas 3 e 5 há a inicialização das estruturas de controle dos pesos, pontuações e contagens dos usos dos operadores de inserção, remoção e ruído. Todas estas estruturas são inicializadas em 0. Estas informações são atualizadas ao longo das iterações de maneira a dar vantagens competitivas aos operadores mais eficientes (com maior peso) no momento da seleção de operadores.

Nas linhas 6 e 7 são definidos o limite de iterações e a inicialização do contador de iterações, respectivamente. Entre as linhas 8 e 40 ocorre o processo de otimização. A cada iteração do algoritmo é realizada a definição da temperatura para a iteração atual (linha 9), segundo a função  $T \leftarrow (1 - it/itMax)^t$ , a qual leva em conta o número da iteração atual, o número máximo de iterações e o fator de resfriamento  $t$ , como definido por Bayer (2021). Esta função de resfriamento permite a não definição de uma temperatura inicial, e ainda atrela a temperatura a quantidade de iterações. Em seguida, é realizada a seleção de um operador de remoção do conjunto  $RO$  (linha 11), um operador de inserção do conjunto  $IO$  (linha 12) e se irá ou não fazer uso de ruído do conjunto  $NO$  (linha 13). Este último é realizado por meio da seleção entre duas opções: usar ruído (*true*) e não usar ruído (*false*), em que ambos são considerados operadores para que sejam incluídos no processo adaptativo do ALNS, ou seja, ao selecionar o operador de ruído *true*, os operadores escolhidos de remoção e inserção irão conter ruídos em seus processos internos e o operador de ruído será atualizado conforme a eficiência do seu uso como os demais operadores, com base no trabalho de Schmitt (2020). A seleção de um determinado operador é realizada por um método de roleta (Equação 3.7), em que cada operador tem a seguinte probabilidade de ser escolhido:

$$\frac{\omega_i}{\sum_{i \in O} \omega_i} \quad (3.7)$$

em que  $O$  é o conjunto de operadores de um determinado tipo (remoção, inserção ou ruído) e  $\omega_i$  é o peso de cada operador. Assim, os operadores com maior peso têm maior probabilidade de ser selecionado. Após a seleção dos operadores, a solução corrente  $S$  é copiada para  $S'$  (linha 15). A quantidade de clientes  $y$  a serem removidos da solução é definida na linha 16 por um valor aleatório entre  $q_{min} \leq y \leq \min(q_{max}, \delta n)$ , segundo experimentos em Ropke e Pisinger (2006), em que  $\delta$  é um parâmetro de controle e  $n$  o número de clientes atendidos (em rota),  $q_{min}$  é o limite inferior e  $q_{max}$  o limite superior. Na linha 17 são removidos os  $y$  clientes da solução utilizando o operador de remoção selecionado e com uso ou não de ruído segundo a seleção de *noise*. Na linha 18 os  $y$  clientes removidos são transferidos para a *Holding List* (HL) da solução

$S'$ . Em seguida é utilizado o operador de inserção selecionado para tentar inserir os clientes da HL de volta para as rotas de maneira mais eficaz e com possibilidade de ruído, segundo o valor de *noise* (linha 19). Com isso, uma nova solução é gerada  $S'$ . Deve-se ponderar que só são aplicadas alterações factíveis na solução, ou seja, aquelas que respeitem as restrições de carga do veículo e de janelas de tempo dos clientes e depósito.

Entre as linhas 21 e 32 são realizadas a avaliação da solução e pontuação dos operadores segundo a qualidade da solução  $S'$ , utilizando o Algoritmo 4 para comparar as soluções. Caso a solução gerada  $S'$  for melhor avaliada que a melhor solução global  $S_{best}$ , segundo a função objetivo hierárquica definida, é aplicada uma busca local (Algoritmo 14) a partir da solução  $S'$  para a intensificação da mesma e em seguida as soluções  $S_{best}$  e  $S$  (solução corrente) assumem a solução  $S'$  como melhor solução global e local, e ainda as pontuações dos operadores selecionados são incrementadas com o valor  $\theta_1$  (linhas 21-24). Caso a solução  $S'$  for melhor apenas que a solução corrente  $S$ ,  $S$  assume a solução  $S'$  e as pontuações dos operadores selecionados são incrementadas com  $\theta_2$  (linhas 25-27). Caso não houver nenhuma melhora, um critério de aceitação de piora é utilizado (Algoritmo 5), segundo definido por Bayer (2021). É importante salientar que o método de aceitação utilizado sofreu adaptações em relação ao que foi definido em Bayer (2021), pois neste trabalho é considerado o número de clientes não atendidos (tamanho da *Holding List*). Caso a HL da solução  $S'$  for maior que a da solução  $S$ , ou seja, se a nova solução gerada obtiver número de atendimentos menor que a solução original, não há possibilidade de aceitação de piora, como maneira de incentivar os atendimentos como prioridade. Caso contrário, o critério de aceitação original é aplicado conforme equação  $cost(S') \leq cost(S) \times (1 + Rand(0, 1) \times T)$ , em que  $T$  é a temperatura corrente, e a função  $cost()$  é responsável pelo cálculo de custo de uma solução, dada pela soma dos custos fixos e variáveis. Caso a piora seja aceita, a solução corrente  $S$  recebe a solução  $S'$  e as pontuações dos operadores são incrementadas em  $\theta_3$  (linhas 28-31). Na linha 32 a contagem dos operadores utilizados é incrementada.

---

#### Algorithm 4 Comparação entre Soluções - Função Objetivo Hierárquica

---

**Require:** Soluções  $S_a$  e  $S_b$

```

1: function COMPARE( $S_a, S_b$ )
2:    $HL_a \leftarrow Holding\ List\ de\ S_a$ 
3:    $HL_b \leftarrow Holding\ List\ de\ S_b$ 
4:   return  $|HL_a| < |HL_b|$  OR ( $HL_a = HL_b$  AND  $cost(S_a) < cost(S_b)$ )
5: end function

```

---

Após a avaliação da solução gerada, é verificado se o tamanho do segmento  $\kappa$  foi atingido (linha 34). Esse parâmetro tem o intuito de evitar que os pesos dos

---

**Algorithm 5** Critério de aceitação segundo *Simulated Annealing*


---

**Require:** Solução  $S'$ , Solução  $S$ , Temperatura  $T$

```

1: function ACCEPT( $S', S, T$ )
2:    $Rand \leftarrow$  Gerador de números aleatórios
3:    $HL' \leftarrow$  Holding List de  $S'$ 
4:    $HL \leftarrow$  Holding List de  $S$ 
5:   if  $|HL'| > |HL|$  then
6:     return false
7:   else
8:     return  $cost(S') \leq cost(S) \times (1 + Rand(0, 1) \times T)$ 
9:   end if
10:  end function

```

---

operadores sejam atualizados a cada iteração. Dessa maneira, as pontuações dos operadores são acumuladas por  $\kappa$  iterações para então serem usados para atualizar os pesos. Caso o fim do segmento seja atingido, os pesos  $\omega$  dos operadores são atualizados (linha 35) segundo a Equação 3.8 a seguir:

$$\omega_i = \omega_i(1 - \rho) + \rho \left( \frac{\phi_i}{\mu_i} \right) \quad (3.8)$$

em que  $0 \leq \rho \leq 1$  é um fator de reação que controla o quanto rápido os ajustes de pesos absorvem as mudanças realizadas pelos operadores no último segmento. Quanto maior o valor de  $\rho$ , mais importância é dada aos operadores utilizados recentemente, e quanto menor o seu valor mais importância para o histórico de operadores ao longo do processo de busca. Em seguida, as pontuações e contagens de todos os operadores são reiniciadas (linhas 36 e 37). Na linha 39 a iteração é incrementada. Ao final do algoritmo é retornada a solução  $S_{best}$  (linha 41).

Na sequência serão apresentados os operadores de remoção e inserção utilizados neste trabalho e suas peculiaridades frente ao problema proposto. Os operadores de remoção são: *Random Removal*, *Worst Removal*, *Shaw Removal* e *Route Removal*. Os operadores de inserção são: *Greedy Insertion* e variações do *Regret-k* (*Regret-2*, *Regret-3*, *Regret-4* e *Regret-m*), em que  $m$  é de rotas da solução

### 3.3.2.1 Random Removal

A heurística de remoção aleatória (*random removal*) é o método de remoção mais simples, em que remove  $y$  clientes de uma solução  $S$  de maneira aleatória. O principal intuito deste método é gerar variações consideráveis na solução, uma vez que não aplica procedimento guloso em seu mecanismo. Dado o comportamento adaptativo do ALNS, esta heurística se mostra importante para eventualmente liderar a busca para regiões mais distantes e sair de mínimos locais. Dado o caráter aleatório (não guloso), este método não necessita da aplicação de ruídos, ou seja, independente

do valor de  $noise$ , definido na seção anterior, não gerará impacto neste operador. O Algoritmo 6 apresenta de que maneira opera esta heurística.

---

**Algorithm 6** *Random Removal*


---

**Require:** Solução  $S$ , quantidade de remoções  $y$

- 1:  $N \leftarrow$  clientes atendidos em  $S$
  - 2:  $shuffle(N)$
  - 3:  $Rem \leftarrow N[N_0, \dots, N_{y-1}]$
  - 4: Remover clientes de  $Rem$  da solução  $S$
  - 5: Inserir clientes de  $Rem$  na *Holding List*  $HL$  de  $S$
  - 6: **return**  $S$
- 

O algoritmo recebe a quantidade de remoções  $y$  e a solução  $S$ . O primeiro passo é mapear os clientes atendidos pelas rotas da solução em uma lista de clientes atendidos  $N$  (linha 1) e em seguida o embaralhamento aleatório desta lista (linha 2). Na linha 3 são definidos os clientes a serem removidos ao captar os primeiros  $y$  clientes da lista  $N$ . Nas linhas 4 e 5 são realizadas as remoções dos clientes selecionados e as inserções dos mesmos na *Holding List* ( $HL$ ) de  $S$ . Ao final, é retornada a solução  $S$  com as remoções efetivadas.

### 3.3.2.2 Worst Removal

A remoção de pior custo (*worst removal*) busca remover clientes que representam maior impacto no custo da solução, ou seja, clientes que causam maior redução de custo se forem removidos. A função que calcula o impacto que a remoção de um cliente  $c$  exerce sobre a solução é dada por  $gain(S, c) = cost(S) - cost(S \setminus \{c\})$ ,  $\forall c \in S$ , a qual calcula a diferença de custo da solução antes e depois da possível remoção do cliente  $c$ . O Algoritmo 7 apresenta todos os procedimentos do método.

---

**Algorithm 7** *Worst Removal*


---

**Require:** Solução  $S$ , quantidade de remoções  $y$

- 1:  $N \leftarrow$  clientes atendidos em  $S$
  - 2:  $Imp \leftarrow$  impacto de cada cliente em  $N$  segundo  $gain(S, c)$ ,  $\forall c \in S$
  - 3:  $removed \leftarrow 0$
  - 4: **while**  $removed \leq y$  **do**
  - 5:     Ordenar de forma decrescente a lista de impactos  $Imp$
  - 6:      $r \leftarrow$  posição de remoção segundo  $r = abs(rand^\sigma \times |Imp|)$
  - 7:      $c \leftarrow Imp[r]$
  - 8:     Remova o cliente  $c$  de  $S$
  - 9:     Insira cliente  $c$  na *Holding List*  $HL$  de  $S$
  - 10:     $Imp \leftarrow$  impacto de cada em  $N$  após a remoção
  - 11:     $removed \leftarrow removed + 1$
  - 12: **end while**
  - 13: **return**  $S$
-

O algoritmo recebe a quantidade de clientes a serem removidos  $y$  e a solução  $S$ . Na linha 1 são mapeados todos os clientes atendidos pelas rotas de  $S$ . Na linha 2 é calculado o impacto de cada cliente na solução, segundo a equação  $gain(S, c)$ , mantido na lista de impactos  $Imp$ . Entre as linhas 3 e 12 é realizada a remoção iterativa dos  $y$  clientes. Na linha 5 a lista  $Imp$  é ordenada do maior para o menor impacto/ganho. Na linha 6 é escolhida a posição  $r$  da lista  $Imp$  que contém o cliente  $c$  a ser removido, de forma que  $r = abs(rand^\sigma \times |Imp|)$ , em que  $rand$  é um número aleatório entre 0 e 1,  $\sigma$  é um parâmetro de grau de ruído e  $|Imp|$  é o tamanho da lista de impactos. Desse modo, optou-se pela aplicação de ruído intrínseco ao método, de forma a evitar possíveis ciclos de remoções dos mesmos clientes ao longo das iterações do ALNS. Assim, o valor do operador *noise* não gera efeito neste método.

Nas linhas 7 a 9 é realizada a remoção do cliente  $c$  selecionado e a inserção do mesmo na *Holding List* (HL) da solução  $S$ . Em seguida, a lista de impactos  $Imp$  é atualizada, já que com a remoção de um cliente gera atualização no custo da solução, o que causa novos valores para  $gain$ . Contudo, o custo desta atualização pode ser reduzido, de maneira a atualizar apenas os impactos/ganhos dos clientes contidos na rota modificada pela remoção e evitando cálculos redundantes ou desnecessários de clientes em rotas não alteradas. Ao final, a solução  $S$  é retornada.

### 3.3.2.3 Shaw Removal

A remoção relacional ou também chamada de *Shaw Removal*, visa a remoção dos clientes com maior grau de similaridade entre si. Neste trabalho a similaridade é dada através da distância entre os clientes. O objetivo é que, ao remover tais clientes, aumente a probabilidade de serem inseridos posteriormente em posições distintas e mais eficientes em relação à solução original. Dessa forma, esta heurística se mostra eficiente para cenários em que remover clientes muito distintos, resultam em possíveis reinserções nas mesmas posições originais, o que não gera potencial melhoria. Para o cálculo da medida de relação entre clientes  $i$  e  $j$  é utilizada a equação  $Shaw(i, j) = (d_{ij} / max(d))$ , em que  $d_{ij}$  é a distância entre os clientes  $i$  e  $j$  e  $max(d)$  é a distância máxima entre dois clientes da instância de teste.

O Algoritmo 8 recebe a solução  $S$ , a quantidade de remoções  $y$  o valor do operador *noise* que indica a presença ou não de ruído no método de remoção. Na linha 1 são mapeados os clientes atendidos pelas rotas em  $S$ . Na linha 2 são calculados os valores de similaridade entre os clientes utilizando a equação  $Shaw(i, j)$ , resultando em uma matriz de similaridade  $Shaw$ . Na linha 3 é removido um cliente  $c$  de maneira aleatória da solução  $S$  e inserido na HL (linha 4) e na lista de clientes removidos  $R$  (linha 5). Entre as linhas 7 e 20 é realizada a remoção iterativa dos clientes até atingir  $y$  clientes removidos.

---

**Algorithm 8** *Shaw Removal*


---

**Require:** Solução  $S$ , quantidade de remoções  $y$  e parâmetro booleano de ruído  $noise$

- 1:  $N \leftarrow$  clientes atendidos em  $S$
- 2:  $Shaw \leftarrow$  matriz de similaridade entre os clientes em  $N$  segundo Eq.  $Shaw(i, j)$
- 3: Remova aleatoriamente um cliente  $c$  de  $S$
- 4: Insira  $c$  na *Holding List*  $HL$  de  $S$
- 5:  $R \leftarrow \{c\}$
- 6:  $removed \leftarrow 1$
- 7: **while**  $removed \leq y$  **do**
- 8:      $r \leftarrow$  escolha aleatoriamente dentre os clientes removidos  $R$
- 9:      $Shaw_r \leftarrow$  similaridade dos clientes remanecentes com o cliente  $r$
- 10:    Ordenar  $Shaw_r$  de forma crescente
- 11:     $p \leftarrow 0$
- 12:    **if**  $noise = true$  **then**
- 13:        $p \leftarrow$  posição de remoção segundo  $p = abs(rand^\sigma \times |Shaw_r|)$
- 14:    **end if**
- 15:     $c \leftarrow Shaw_r[p]$
- 16:    Remova o cliente  $c$  de  $S$
- 17:    Insira cliente  $c$  na *Holding List*  $HL$  de  $S$
- 18:     $R \leftarrow R \cup \{c\}$
- 19:     $removed \leftarrow removed + 1$
- 20: **end while**
- 21: **return**  $S$

---

Na linha 8 é selecionado aleatoriamente um cliente  $r$  da lista de clientes removidos  $R$  para servir de base para o cálculo de similaridade, de forma que o cliente em rota com maior similaridade em relação ao cliente  $r$  será o próximo a ser removido. Entre as linhas 10 e 15 é determinado o próximo cliente a ser removido. Primeiramente é realizada a ordenação da lista  $Shaw_r$  do menor para o maior valor da função  $shaw$  a partir do cliente  $r$  para todos os clientes ainda em rota. Caso o valor do parâmetro  $noise$  for verdadeiro, é aplicada a equação  $p = abs(rand^\sigma \times |Shaw_r|)$  para definir a posição da lista de  $Shaw_r$  que se encontra o próximo cliente  $c$  a ser removido, dado que  $rand$  é um valor aleatório entre 0 e 1,  $\sigma$  o parâmetro de controle de ruído e  $|Shaw_r|$  é o tamanho da lista de valores de similaridade a partir de  $r$ . Caso não haja ruído, o primeiro cliente  $c$  da lista  $Shaw_r$  é escolhido para ser removido. Na linha 16 é efetivada a remoção do cliente  $c$  da solução  $S$  e em seguida este é alocado na *Holding List* e na lista de removidos  $R$  (linhas 17 e 18). Ao final, é retornada a solução  $S$  (linha 21).

### 3.3.2.4 Route Removal

O operador *Route Removal* foi proposto neste trabalho com o objetivo de proporcionar a remoção de rotas inteiras e não apenas de parte delas, dado que os demais operadores utilizados podem nunca gerar a remoção de todos os clientes de uma rota. Este operador se mostra importante dado que o problema proposto contém uma

frota de veículos distintos, e com isso é fundamental que sejam testadas modificações os veículos das rotas existentes, ou a criação de novas rotas com tipos de veículos variados. Para isso, este operador propõe a remoção de 40% das rotas existentes de forma completa. Ou seja, se há 100 rotas, 40 delas serão removidas e os respectivos clientes serão transferidos para a *Holding List*. O valor de 40% foi definido de forma experimental para vários cenários.

---

#### Algorithm 9 Route Removal

---

**Require:** Solução  $S$

```

1:  $removed \leftarrow \text{ceil}(\text{routes}(S) \times 0.4)$ 
2: while  $removed > 0$  do
3:   Selecionar aleatoriamente uma rota  $r$  da solução  $S$ 
4:   Remover a rota  $r$  da solução  $S$ 
5:   Remover clientes da rota  $r$  e incluir na HL
6:   Incluir veículo da rota  $r$  na lista de veículos livres
7:    $removed \leftarrow removed - 1$ 
8: end while
9: return  $S$ 
```

---

O Algoritmo 9 recebe a solução  $S$  e na linha 1 define a quantidade de rotas a serem removidas, em que  $\text{routes}(S)$  representa a quantidade de rotas da solução  $S$ . Entre as linhas 2 e 8 são selecionadas rotas aleatoriamente e estas são removidas da solução, seus respectivos clientes são transferidos para a HL e o veículo utilizado é transferido para a lista de veículos livres. Este operador não utiliza o ruído *noise*.

#### 3.3.2.5 Greedy Insertion

As heurísticas de inserção tem objetivo de inserir clientes não atendidos de volta na solução, em geral após a aplicação de um método de remoção, a exemplo dos métodos apresentados nas seções anteriores. O método de inserção gulosa (*Greedy Insertion*) é algoritmo de construção simples, que insere os clientes presentes na *Holding List* na posição factível que gera o menor acréscimo de custo para a solução. O operador foi definido com base em Ropke e Pisinger (2006) e sofreu adaptações para o problema proposto. O Algoritmo 10 apresenta os passos desse métodos.

O algoritmo recebe a solução  $S$  contendo as rotas, a HL contendo os clientes não atendidos e removidos pelo método de remoção e o parâmetro de ruído (*noise*). Na linha 1 a *Holding List* é tomada. Na linha 2 é armazenado em  $K$  os tipos de veículos livres, que não estão operando em rotas. O método de inserção de fato opera entre as linhas 3 e 14, e finaliza quando a *Holding List* estiver vazia ou não houver posição factível de inserção. Na linha 4 é tomada a lista de rotas  $R$  da solução  $S$ . Em seguida, é calculada o conjunto de custos se inserção dos clientes da HL nas rotas  $R$ , através da função *getCosts* (Algoritmo 11). Na linha 7 é selecionada a inserção *Ins* de menor

---

**Algorithm 10** *Greedy Insertion*


---

**Require:** Solução  $S$  e ruído  $noise$

```

1:  $HL \leftarrow HoldingList$  de  $S$ 
2:  $K \leftarrow$  tipos de veículos com frota disponível
3: while  $|HL| > 0$  do
4:    $R \leftarrow$  rotas de  $S$ 
5:    $\Delta \leftarrow getCosts(R, HL, noise)$ 
6:   ▷ Escolha da melhor inserção
7:    $Ins \leftarrow$  inserção factível de custo mínimo em  $\Delta$ 
8:   if  $Ins = null$  then
9:     return  $S$ 
10:  end if
11:  ▷ Efetivar inserção do cliente escolhido
12:   $S \leftarrow insert(S, Ins)$ 
13:  Remova cliente  $c$  de  $Ins$  da Holding List  $HL$ 
14: end while
15:  $S \leftarrow$  construção semi-paralela a partir de  $S$ ,  $HL$  e  $K$  (Algoritmo 1)
16: return  $S$ 

```

---

custo, que contém o cliente a ser inserido, a rota e a posição de inserção. Caso não haja inserção factível o algoritmo finaliza (linhas 8-10). Caso haja posição factível, a inserção é efetivada pela função *insert* (Algoritmo 12), a qual insere o cliente  $c$  na rota  $r$  na posição  $p$  trazidos por  $Ins$ . Após a inserção, o cliente  $c$  é removido da *Holding List* e o ciclo se inicia novamente com as rotas existentes  $R$  e a atualização dos custos de inserção  $\Delta$ . Neste trabalho foi proposta uma alteração neste operador e nos demais operadores de inserção. Ao final do processo de inserção, caso ainda haja clientes na  $HL$ , devido a não existência de inserção factível, é utilizado o algoritmo usado na solução inicial para criação de rotas por meio de método de construção semi-paralelo para a criação de novas rotas (definido na Seção 3.3.1) para atender estes clientes se houver factibilidade e frota disponível para tal (linha 15). Ao final, o algoritmo retorna a solução  $S$  (linha 16) com os clientes re-inseridos nas rotas e em caso de restarem clientes não atendidos por não factibilidade, estes permanecem na *Holding List*.

Para o cálculo dos custos de inserção (Algoritmo 11) são considerados os clientes da  $HL$ , as rotas existentes  $R$  e o parâmetro de ruído  $noise$ . Nessa função são calculados os custos de inserção de cada cliente não atendido em cada rota existente na posição factível de menor custo (linhas 5-10). Para calcular o custo de inserção de um cliente  $c \in HL$  em uma rota  $r \in R$  na posição de menor custo (linhas 7), utiliza-se  $\Delta_{cr} = C_{cr} + \eta$ , em que  $C_{cr}$  representa o acréscimo de custo ao inserir um cliente em uma determinada rota na posição factível de menor custo, calculado pela Equação 3.4 introduzida no método utilizado para a geração da solução inicial, de forma a iterar sobre todas as posições de uma rota e encontrar a posição de inserção com menor impacto no custo total.

---

**Algorithm 11** Cálculo da Matriz de Custos de Inserção
 

---

**Require:** Rotas  $R$ , Holding List  $HL$  e parâmetro de ruído  $noise$

- 1: **function** GETCOSTS( $R, HL, noise$ )
- 2:      $\Delta \leftarrow$  conjunto de custos de inserção
- 3:     **for all**  $r$  in  $R$  **do**
- 4:         **for all**  $c$  in  $HL$  **do**
- 5:             **if** inserir  $c$  na rota  $r$  é factível **then**
- 6:                  $\Delta_{cr} \leftarrow$  posição de menor custo de inserção de  $c$  na rota  $r$
- 7:             **end if**
- 8:         **end for**
- 9:     **end for**
- 10:   **return**  $\Delta$
- 11: **end function**

---

Para os cálculos de custo, se o parâmetro  $noise$  for verdadeiro, um custo extra  $\eta = rand(-x, x)$  é adicionado ao custo de inserção como ruído, de modo a evitar que sempre seja selecionado o cliente de menor custo de forma gulosa, em que  $x = \gamma \times max(d)$ ,  $\gamma$  um fator de controle dos ruídos para métodos de inserção e  $max(d)$  a distância máxima entre os clientes do problema.

---

**Algorithm 12** Efetivar Inserção de Cliente na Solução
 

---

**Require:** Solução  $S$ , Inserção  $Ins$

- 1: **function** INSERT( $S, Ins$ )
- 2:     **if**  $Ins$  é de uma rota existente **then** ▷ Inseri em rota já existente
- 3:          $c, r, pos \leftarrow$  cliente, rota e posição de inserção de  $Ins$
- 4:         Insira cliente  $c$  em  $S$  na rota  $r$  na posição  $pos$
- 5:     **else** ▷ Cria nova rota
- 6:          $c, k \leftarrow$  cliente e tipo de veículo de  $Ins$
- 7:         Crie nova rota em  $S$  com veículo do tipo  $k$  para atender  $c$
- 8:     **end if**
- 9:     **return**  $S$
- 10: **end function**

---

### 3.3.2.6 Regret-k Insertion

O método de inserção por arrependimento (*regret insertion*) tenta melhorar o desempenho da heurística de inserção gulosa (*greedy insertion*) por meio de um mecanismo de previsibilidade do impacto de realizar uma inserção antes dela ser efetivamente realizada, mecanismo este guiado por um critério de arrependimento. Dessa forma, ao calcular os valores de arrependimento para cada possível inserção de cliente, é possível evitar a escolha de uma inserção que pareça promissora em um primeiro momento, mas que geraria um impacto negativo nas inserções subsequentes. O Algoritmo 13 apresenta como opera este método.

---

**Algorithm 13** *Regret-k Insertion*


---

**Require:** Solução  $S$ , ruído  $noise$

```

1:  $HL \leftarrow HoldingList$  de  $S$ 
2:  $K \leftarrow$  tipos de veículos com frota disponível
3: while  $|HL| > 0$  do
4:    $R \leftarrow$  rotas de  $S$ 
5:    $\Delta \leftarrow getCosts(R, HL, noise)$ 
6:   ▷ Escolha da melhor posição de inserção
7:    $Ins \leftarrow$  inserção factível de regret máximo em  $\Delta$  segundo Eq. 3.9
8:   if  $Ins = null$  then
9:     return  $S$ 
10:  end if
11:   $S \leftarrow insert(S, Ins)$  ▷ Efetivar inserção do cliente escolhido
12:  Remova cliente  $c$  de  $Ins$  da Holding List  $HL$ 
13: end while
14:  $S \leftarrow$  construção semi-paralela a partir de  $S$ ,  $HL$  e  $K$  (Algoritmo 1)
15: return  $S$ 

```

---

De maneira geral, este algoritmo possui operações semelhantes ao método guloso apresentado anteriormente, apenas com alterações na escolha da próxima inserção. O ciclo iterativo de inserções (linhas 3-13) só é finalizado quando todos os clientes da *Holding List* forem inseridos nas rotas ou não houver posição factível de inserção. O conjunto de custos de inserção  $\Delta$  calculado segundo o Algoritmo 11 (linha 5). Na linha 7 é selecionada a inserção  $Ins$  que será realizada. Porém, diferente do método de seleção utilizado na inserção gulosa, neste algoritmo a inserção escolhida não é baseada no menor custo e sim no critério de arrependimentos (*regret*). Considere  $x_{ck} \in \{1, 2, \dots, m\}$  como  $k$ -ésima melhor rota para a qual o cliente  $c$  pode ser inserido,  $m$  o número de possíveis rotas/veículos para inseri-lo e  $\Delta_{c,k} \leq \Delta_{c,k'}$  para  $k \leq k'$  representam a ordenação da melhor para a pior rota para inserir  $c$ . Assim,  $k = 1$  representa a melhor rota,  $k = 2$  a segunda melhor rota e assim por diante. O valor de arrependimento é calculado por  $regret = \Delta_{c,2} - \Delta_{c,1}$ , em que representa distância entre os custos de inserção do cliente  $c$  na melhor rota e na segunda melhor rota. Após o cálculo, o cliente que obtiver o maior valor de arrependimento será inserido na rota de menor custo, ou seja, na rota cujo  $k = 1$ .

O objetivo é selecionar o cliente e a rota com maior valor de arrependimento, ou seja, aquela em que a diferença de custos de inserção deste cliente entre a melhor e a segunda melhor rota é maior. Desse modo, tem-se mais segurança ao fazer o movimento de inserção deste cliente na solução e ao invés de outro cliente ainda não atendido.

A heurística pode ser estendida para o  $regret_k$ , em que calcula não só a diferença entre a primeira e segunda melhor rota, mas sim o somatório das diferenças

entre a melhor rota as demais  $k$  rotas. Neste trabalho os valores de  $k$  são 1, 2, 3 e  $m$ , sendo  $m$  o número de rotas existentes na solução  $S$ . A Equação 3.9 representa este cálculo.

$$\max_{c \in HL} \left\{ \sum_{j=1}^k (\Delta_{c,j} - \Delta_{c,1}) \right\} \quad (3.9)$$

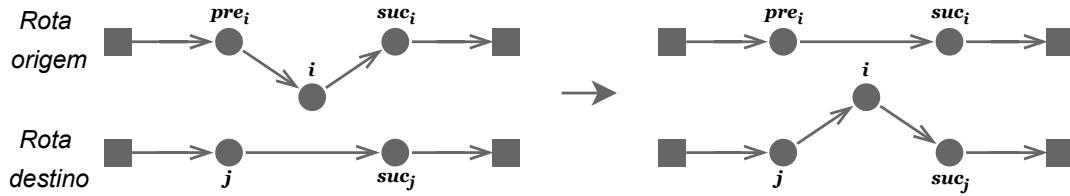
Em casos de mesmo valor de arrependimento para mais de um cliente, aquele que obtiver o menor custo de inserção será inserido. É importante pontuar que os ruídos são considerados da mesma maneira que na heurística gulosa. De forma análoga ao que é apresentado na Seção 3.3.2.5 para o *Greedy Insertion*, caso ao final do processo de inserção haja clientes ainda não atendidos de forma factível, novas rotas são criadas utilizando a estratégia semi-paralela provinda da solução inicial, cujo objetivo é criar novas rotas de forma mais eficiente com veículos apropriados (linha 14). Após esse processo, se ainda houverem clientes não atendidos de modo factível, estes permanecem na HL. Ao final do processo, a solução  $S$  é retornada (linha 15).

A utilização do método de construção semi-paralelo como complemento dos métodos de inserção têm papel fundamental no desempenho do algoritmo de modo a criar rotas novas com potencialmente veículos distintos dos utilizados na solução antes do processo de inserção. Esta operação é utilizada de forma mais intensa, em geral após o operador de remoção proposto neste trabalho *Route Removal* realizar a remoção de rotas inteiras, o que faz com que a junção destes métodos realize testes com novos tipos de veículos, o que é essencial para o bom desempenho do algoritmo para problemas de roteamento com frota heterogênea.

### 3.3.3 Busca Local

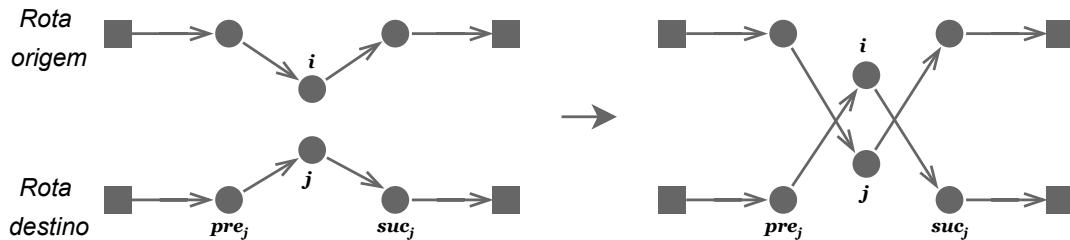
Os métodos *Relocate*, *Exchange* e *Cross* são as operações de exploração de vizinhança, introduzidos por Savelsbergh (1992), utilizados na busca local apresentada neste trabalho. Estas operações modificam simultaneamente duas rotas contidas na solução  $S$ , de forma a trocar posições dos clientes entre si. As três operações são especificadas a seguir e ilustradas pelas Figuras 12, 13 e 14, respectivamente. O cliente  $i$  sempre se refere a um cliente da rota de *origem*, assim como seu predecessor  $pre_i$  e sucessor  $suc_i$ . O cliente  $j$ , o seu predecessor  $pre_j$  e sucessor  $suc_j$ , sempre fazem parte da rota *destino*.

- **Relocate:** tenta transferir um cliente de uma rota para outra. Assim, opera sobre  $\{(pre_i, i), (i, suc_i), (j, suc_j)\} \rightarrow \{(pre_i, suc_i), (j, i), (i, suc_j)\}$ . A ordem de escolha do cliente  $i$  na rota de origem deve ser a ordem reversa ( $i = n - 1, \dots, 1$ ). Na rota destino, o cliente  $j$  também deve ser escolhido na mesma ordem ( $j = n - 1, \dots, 0$ ), em que  $n$  é o número de clientes de uma determinada rota.

Figura 12 – Operador *Relocate*

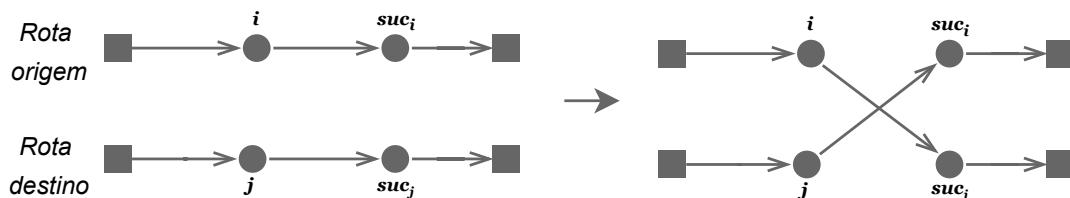
Fonte: Próprio Autor

- **Exchange:** seleciona um cliente em cada rota e os troca de posição (*swap*). Assim,  $\{(pre_i, i), (i, suc_i), (pre_j, j), (j, suc_j)\} \rightarrow \{(pre_i, j), (j, suc_i), (pre_j, i), (i, suc_j)\}$ . A ordem de escolha de  $i$  deve ser a mesma do *relocate* ( $i = n - 1, \dots, 1$ ) e a escolha de  $j$  deve seguir a mesma ordem ( $j = n - 1, \dots, 1$ ).

Figura 13 – Operador *Exchange*

Fonte: Próprio Autor

- **Cross:** tenta combinar (cruzar) duas rotas distintas de maneira que a parte de uma rota se torne parte de outra. Assim,  $\{(i, suc_i), (j, suc_j)\} \rightarrow \{(i, suc_j), (j, suc_i)\}$ . A escolha de  $i$  segue a ordem reversa em que  $i = n - 1, \dots, 0$  e a escolha de  $j$  segue a mesma ordem ( $j = n - 1, \dots, 0$ ).

Figura 14 – Operador *Cross*

Fonte: Próprio Autor

O Algoritmo 14 apresenta a estratégia de busca local realizada, em que a solução é atualizada quando há uma melhora na função objetivo hierárquica definida na Seção 3.2.2, em que leva em consideração a quantidade de clientes atendidos (tamanho da *Holding List*) e o custo total da solução. São realizadas apenas modificações que mantenham a solução factível em relação às restrições do problema.

---

**Algorithm 14** Busca Local
 

---

**Require:** Solução  $S$

```

1:  $S' \leftarrow$  cópia da solução  $S$ 
2:  $S_{ls} \leftarrow$  cópia da solução  $S$ 
3:  $improve \leftarrow true$ 
4: while  $improve = true$  do
5:    $improve \leftarrow false$ 
6:    $S' \leftarrow cross(S')$ 
7:    $S' \leftarrow relocate(S')$ 
8:    $S' \leftarrow exchange(S')$ 
9:    $S' \leftarrow relocateSingleRoute(S')$ 
10:   $S' \leftarrow exchangeSingleRoute(S')$ 
11:  if  $compare(S', S_{ls})$  then
12:     $S_{ls} \leftarrow$  cópia da solução  $S'$ 
13:     $improve \leftarrow true$ 
14:  end if
15: end while
16: return  $S_{ls}$ 
  
```

---

Nas linhas 1 e 2 são realizadas as cópias da solução  $S$  e das linhas 2 a 15 é realizado o ciclo de intensificação da solução. Os três primeiros operadores (*cross*, *relocate* e *exchange*) são aplicados percorrendo os possíveis pares de rotas e verificando as possíveis mudanças que causam diminuição do custo (linhas 6, 7 e 8). É importante pontuar que a frota de veículos ainda não utilizada também é considerada como uma possível rota, de maneira a verificar se transferir clientes de uma rota existente para uma nova rota representa redução de custo, considerando ainda os custos fixos de utilização de um novo veículo. Além disso, caso as *Holding List* HL não estiver vazia, nos operadores de *relocate* e *exchange* são realizadas operações de transferência e troca com a HL com o intuito de diminuir a quantidade de clientes não atendidos ou causar redução de custo por troca de clientes em rota e clientes não atendidos.

Em seguida são aplicados os operadores *relocate* e *exchange* individualmente em cada rota (*single route*) de maneira a realizar trocas e transferências de clientes da mesma rota, apenas em posições distintas. Após a aplicação de todos os operadores na solução  $S'$  é realizada a comparação com a melhor solução corrente  $S_{ls}$ . Caso a solução  $S'$  obteve melhoria em relação a função objetivo hierárquica, esta assume o posto de melhor solução  $S_{ls}$ . A busca local é finalizada quando não houver mais nenhuma melhora possível a ser realizada. Ao final, a solução  $S_{ls}$  é retornada.

### 3.4 IMPLEMENTAÇÃO DA TECNOLOGIA

O desenvolvimento do método proposto foi realizado em etapas e de forma incremental, em que a cada etapa, levou-se em consideração a formulação do pro-

blema atual e os métodos necessários para a sua solução. Sucessivos testes foram realizados a cada etapa cumprida para medir o impacto das modificações e reflexo das mudanças de parâmetros.

Acerca das tecnologias, o algoritmo foi implementado utilizando a linguagem de programação Java (Versão 11), escolhida pela robustez, experiência do autor com esta linguagem e ferramentas de auxílio de desenvolvimento. Para garantir a qualidade do projeto desenvolvido, foi utilizada a biblioteca de testes JUnit, de maneira a possibilitar a realização de sucessivos testes unitários. Os testes foram utilizados principalmente para validar os dados de entrada do algoritmo, verificar os operadores de modificação da solução funções de verificação das restrições. Foram utilizados recursos de paralelismo fornecidos pelas bibliotecas de *parallel stream* do Java, afim de realizar execuções do algoritmo de modo paralelo e dar agilidade nas experimentações apresentadas no próximo capítulo. Além disso, para monitorar o consumo de CPU e memória ao longo dos testes, foi utilizada a ferramenta *VisualVM*, onde os gargalos de memória puderam ser identificados e as reparações no código realizadas. Por fim, os experimentos e comparações com os *benchmarks* da literatura são descritos no Capítulo 4.

## 4 EXPERIMENTOS E ANÁLISE DOS RESULTADOS

Este capítulo apresenta os experimentos realizados segundo a execução do algoritmo proposto neste trabalho. A Seção 4.1 apresenta o protocolo de experimentação utilizado para a realização dos testes e análise dos resultados. A Seção 4.2 define a base de dados de *benchmark* utilizada nos experimentos. A Seção 4.3 apresenta a definição dos parâmetros utilizados pelo algoritmo. A Seção 4.4 detalha os resultados obtidos pela experimentação para o problema HFVRPTW, bem como sua respectiva análise quantitativa e qualitativa acerca do comportamento do algoritmo, aspectos positivos e limitações da solução. A Seção 4.5 apresenta os resultados obtidos dos experimentos para o problema HFVRPTW-LC, suas respectivas análises quantitativa e qualitativa, bem como considerações sobre o impacto do custo de carga no cenário proposto. Por fim, a Seção 4.6 apresenta as considerações finais sobre os resultados obtidos.

### 4.1 PROTOCOLO DE EXPERIMENTAÇÃO

O protocolo de experimentação definido neste trabalho visa a avaliação do algoritmo proposto frente à literatura existente acerca do problema HFVRPTW-LC, o qual contempla frota heterogênea limitada, janela de tempo e custo de carga. Ao longo da literatura, não foi identificado um trabalho composto por todas estas características e restrições juntas. Assim, para avaliar o método proposto, os experimentos são divididos em duas etapas. A primeira etapa visa comparar o método proposto com a literatura existente para o problema HFVRPTW, desconsiderando o custo de carga, já que existem diversos métodos na literatura para solucionar este problema específico. O objetivo é avaliar o método proposto perante a literatura existente antes de incluir a noção de custo de carga. Esta primeira etapa dos experimentos está descrita na Seção 4.4.

A segunda etapa visa incluir a noção de custo de carga, para avaliar o impacto na utilização deste custo no método proposto. Sendo assim, com base nos resultados obtidos na primeira etapa, são realizados experimentos para o problema HFVRPTW-LC, o qual considera frota heterogênea limitada, janela de tempo e custo de carga. A fim de avaliar o impacto de considerar o custo de carga, a proposta de experimentos é baseada na comparação dos resultados obtidos na primeira etapa para o HFVRPTW (sem custo de carga) *versus* HFVRPTW-LC (com custo de carga). Assim, é possível avaliar se considerar o custo de carga na função objetivo tem impacto no custo total. Este formato de análise é baseado na literatura, onde trabalhos como de

Tang, Zhang e Pan (2010), Zhang, Tang e Fung (2011) e Kuo e Wang (2012) compararam o algoritmo considerando e não considerando o custo de carga com o objetivo de medir seu impacto na minimização do custo.

Ao longo desta seção são realizadas análises quantitativas e qualitativas seguindo as seguintes premissas:

- A análise quantitativa visa avaliar o método proposto acerca da minimização da função objetivo de custo, de modo a comparar com os melhores resultados da literatura até então para instâncias de *benchmark* do problema HFVRPTW e o impacto no custo total com a utilização do custo de carga, para a análise do HFVRPTW-LC. Dentro destas análises estão contidas as tabelas de resultados, diagramas de caixa, análise do tempo de processamento e avaliações da significância dos resultados obtidos. Estas análises se encontram na Seção 4.4.1 (HFVRPTW) e Seção 4.5.2 (HFVRPTW-LC).
- A análise qualitativa tem objetivo de avaliar o comportamento do método em relação a convergência e atuação dos operadores ao longo do processo de busca. Esta análise está concentrada na Seção 4.4.3. Nesta análise são realizadas discussões acerca das características do processo de busca para cenários distintos, apontando possíveis pontos de melhoria.

Para a comparação em relação a minimização da função objetivo de custo, foram selecionadas as instâncias de *benchmark* definida em Jiang et al. (2014), as quais são uma adaptação das instâncias de Solomon (1987). Este *benchmark* estende o proposto por Solomon (1987), adicionando características de janela de tempo e frota heterogênea limitada. A escolha por este *benchmark* foi feita pois é a única instância presente na literatura em que a frota heterogênea de veículos é limitada e cada tipo de veículo possui um custo fixo e um custo variável. Os demais *benchmarks* encontrados possuem apenas ou custo fixo ou custo variável. Sendo assim, as instâncias de Jiang et al. (2014) são as que mais se aproximam ao cenário abordado neste trabalho. A base de dados utilizada para os testes é apresentada na Seção 4.2. Ambas os experimentos de HFVRPTW e HFVRPTW-LC utilizam as mesmas instâncias de *benchmark*. Entretanto, para o cenário HFVRPTW-LC um custo de carga  $w$  precisou ser definido para cada tipo de veículo de cada instância de teste. Os valores de custo  $w$  são definidos na Seção 4.5.1.

Os parâmetros do algoritmo foram definidos de maneira empírica/experimental e seguindo sugestões da literatura, ou seja, não foi aplicado um estudo aprofundado acerca dos parâmetros utilizados. A definição dos parâmetros é apresentada com mais detalhes na Seção 4.3.

Na primeira etapa dos experimentos, para fins de comparação dos resultados do método proposto para o problema HFVRPTW, foi escolhido o trabalho de Molina, Salmeron e Eguia (2020), o qual utiliza uma algoritmo híbrido ACS-VNTS (*Ant Colony System e Variable Neighborhood Tabu Search*), e é o trabalho mais recente sobre HFVRPTW. Ainda, este é o trabalho que apresenta o melhor desempenho para o problema HFVRPTW dentre os disponíveis na literatura, com melhorias para todas as instâncias de teste em relação aos trabalhos antecessores.

Os valores de custo da função objetivo do método proposto e do trabalho de Molina, Salmeron e Eguia (2020) são comparados por meio do cálculo do Desvio Relativo Percentual (*Relative Percent Difference - RPD*). O RDP é calculado para cada instâncias do *benchmark* e consistem na diferença entre o custo obtido pelo método proposto ( $F^{Res}$ ) e o custo apresentado no trabalho de comparação ( $F^{Best}$ ), dividido pelo valor do trabalho de comparação ( $F^{Best}$ ), conforme a seguinte fórmula,  $RPD = (F^{Res} - F^{Best})/F^{Best}$ . A métrica RPD quantifica de maneira percentual a diferença entre os valores de custo total encontrados pelo método proposto e o método a de comparação. Portanto, um RPD igual a 0% indica que ambos os métodos chegaram ao mesmo valor de custo mínimo para a instância de teste em questão, um valor de RPD positivo indica que a solução encontrada pelo método proposto obteve custo maior que a obtida pelo trabalho relacionado (pior resultado), enquanto que um RPD negativo significa que o método proposto encontrou solução com custo menor que o trabalho comparado (melhor resultado).

Para análise de significância estatística é realizado o teste de Wilcoxon (WILCOXON, 1945). Cada instância de teste foi executada 30 vezes para mitigar qualquer viés na otimização, pois se trata de uma abordagem estocástica. Ao final de cada iteração é levantado um conjunto de informações e estatísticas apresentadas ao longo do experimento.

Para a realização da segunda etapa dos experimentos foram necessárias análises aprofundadas para definir os valores do coeficiente de custo de carga  $w$  para cada instância de teste, devido ao fato de não existir na literatura tal definição. Para isso, foram geradas diversas execuções com valores distintos para avaliar os impactos gerados por estas variações. Para avaliar esse impacto, foram utilizados os valores de *RPD* para a medição da diferença percentual entre a solução que não utiliza o custo de carga em sua otimização (HFVRPTW) *versus* a solução que a utiliza (HFVRPTW-LC) e os valores de ocupação dos veículos utilizados nas soluções, como descrito na Seção 4.5.1. De modo análogo a primeira etapa, é utilizado o teste de Wilcoxon para avaliar a significância estatística dos resultados. Os experimentos também foram executados 30 vezes para cada instância de teste.

Para a análise qualitativa são selecionados resultados do método proposto so-

bre instâncias do *benchmark* que contenham características relevantes a serem destacadas. Assim, é realizada uma análise de convergência, considerando um caso com bom desempenho e um caso de mal desempenho. São considerados resultados de bom desempenho aqueles que obtiveram *RPD* negativo ou próximo de 0, caso contrário, é considerado como desempenho não satisfatório. Além disso, a fim de encontrar evidências que justifiquem o desempenho em determinadas situações, são realizadas análises e discussões acerca da atuação dos operadores de remoção e inserção do ALNS ao longo do processo de busca, por meio da análise da evolução dos pesos destes operadores. Com isso, é possível avaliar pontos específicos do comportamento do método proposto que justifiquem seus resultados.

Todos os experimentos foram executados utilizando o mesmo computador, com processador *Intel(R) Core(TM) i7-8565U CPU @ 1.80 GHz*, com 4 núcleos físicos e 8 núcleos virtuais de processamento, e 8 GB de memória RAM. Em relação ao software utilizado, os experimentos foram realizados utilizando uma *runtime environment Java OpenJDK 11.0.15*, executado em um sistema operacional *Ubuntu 18.04.6 LTS*. Dado que o volume e tamanho das instâncias não é elevado, pois são 56 instâncias de cerca de 100 clientes a serem roteados, a arquitetura não foi uma limitação para a execução dos experimentos. Porém, para algumas análises em que o número de execuções é elevado, como a análise para definição dos valores de  $w$  para cada instância, foram utilizadas estratégias de paralelismo, de modo a realizar mais de uma execução do algoritmo simultaneamente, afim de dar agilidade na execução destes experimentos.

## 4.2 BASE DE DADOS

Durante os experimentos realizados neste trabalho são utilizadas as instâncias de *benchmark* para o HFVRPTW definidas por Jiang et al. (2014). Estas instâncias utilizam como base as instâncias introduzidas por Solomon (1987), amplamente utilizadas para o VRPTW (apenas com janela de tempo), e a partir delas propõem uma tabela de veículos heterogêneos para adaptá-las ao VRPTW com frota heterogênea limitada. A escolha por este conjunto de instâncias se deu pela maior proximidade com o cenário proposto neste trabalho, se comparado com outros *benchmarks* da literatura. Esta instância propõe custo fixo, custo variável e um limite de veículos para cada tipo, sendo diretamente relevante para a proposta.

O *benchmark* definido por Solomon (1987) contém 56 instâncias, contendo 100 clientes cada, divididas em 6 categorias distintas, que combinam a disposição dos clientes no espaço e o horizonte de tempo de planejamento (tempo máximo para retorno ao depósito). A Tabela 3 apresenta as 6 categorias e suas respectivas formas de distribuição e horizonte de planejamento.

Tabela 3 – Categorias das instâncias de Solomon

Categoria	Instâncias	Distribuição	Horizonte de Tempo
HC1	9	<i>Cluster</i>	Curto
HC2	8	<i>Cluster</i>	Longo
HR1	12	<i>Random</i>	Curto
HR2	11	<i>Random</i>	Longo
HRC1	8	<i>Random &amp; Cluster</i>	Curto
HRC2	8	<i>Random &amp; Cluster</i>	Longo

Desse modo, as instâncias possuem distribuição de clientes em forma de *Cluster* em HC1 e HC2, *Random* (aleatória) em HR1 e HR2 e *Random & Cluster* (combinação dos dois primeiros) em HRC1 e RC2, como ilustrado na Figura 15. As instâncias HC1, HR1 e HRC1 possuem horizonte de tempo de planejamento mais curto, o que permite apenas poucos clientes por rota (aproximadamente 5 a 10). Por outro lado, as instâncias HC2, HR2 e HRC2 possuem horizonte de tempo mais longo, o que permite maior número de clientes por rota (mais de 30) a serem servidos pelo mesmo veículo. Além disso, dentro de cada categoria, as instâncias possuem clientes com exatamente as mesmas posições, porém com larguras diferentes de janelas de tempo, ou seja, janelas de tempo mais apertadas e outras mais espaçadas, de modo a dar maior diversidade entre as instâncias.

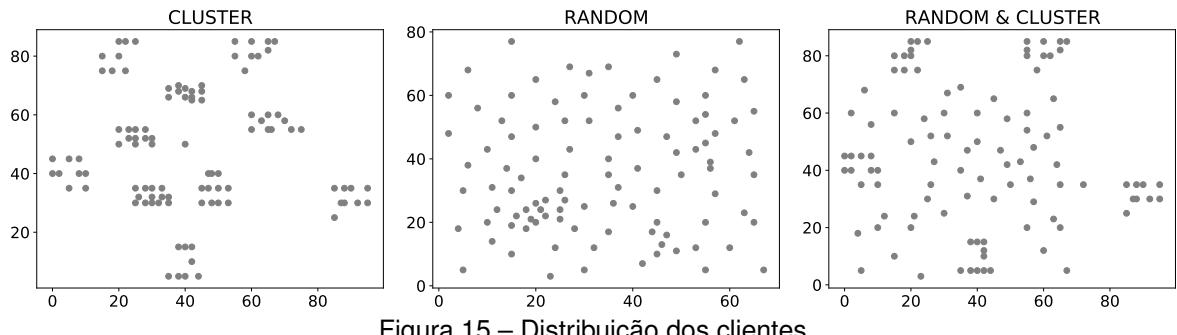


Figura 15 – Distribuição dos clientes

Para cada instância do *benchmark* a formatação do arquivo de entrada possui as informações expostas na Tabela 4. Cada cliente  $i$  possui uma coordenada  $(x_i, y_i)$ , uma demanda  $d_i$ , uma janela de tempo  $[e_i, l_i]$  e um tempo de serviço  $s_i$ .

Tabela 4 – Formatação do arquivo de entrada

ID	Coord. X	Coord. Y	Demandada Delivery	Início Janela de Tempo	Fim Janela de Tempo	Tempo de Serviço
0	$x_0$	$y_0$	$d_0$	$e_0$	$l_0$	$s_0$
1	$x_1$	$y_1$	$d_1$	$e_1$	$l_1$	$s_1$
2	$x_2$	$y_2$	$d_2$	$e_2$	$l_2$	$s_2$
...	...	...	...	...	...	...

Para incluir a noção de frota heterogênea de veículos, Jiang et al. (2014) propuseram uma tabela contendo diferentes tipos de veículos, apresentada na Tabela 5. Esta tabela contém, para cada categoria de instâncias, um conjunto de tipos de veículos, em que cada tipo possui uma quantidade limitada de veículos disponível, uma capacidade de carga, um limite de chegada ao depósito, um custo fixo por utilização do veículo e um custo variável por unidade de distância percorrida.

Tabela 5 – Tabela de Tipos de Veículos (Frota Heterogênea)

Categoria	Tipo	Qtd.	Capacidade	Limite de Chegada (depósito)	Custo Fixo	Custo Variável
HC1	A	20	100	1000	30	1.0
	B	5	200	1236	80	1.2
HC2	A	20	400	3000	100	1.0
	B	5	500	3390	140	1.2
HR1	A	10	50	180	80	1.0
	B	15	80	200	140	1.2
	C	10	120	230	250	1.4
HR2	A	10	300	800	45	1.0
	B	5	400	1000	70	1.2
HRC1	A	10	40	200	60	1.0
	B	20	80	220	150	1.2
	C	10	150	240	300	1.4
HRC2	A	10	100	900	150	1.0
	B	5	200	960	350	1.2

Fonte: (JIANG et al., 2014)

De modo geral, segundo Jiang et al. (2014), os custos fixos, custos variáveis e limite de chegada ao depósito são derivados da seguinte regra: os maiores veículos possuem os maiores custos fixos e variáveis e maior limite de chegada, enquanto que as localizações, demandas, janelas de tempo e tempos de serviço são derivados das instâncias de Solomon originais. É importante pontuar que as noções de custo de carga foram mantidas de fora dos experimentos para o HFVRPTW, ou seja, coeficiente de custo de carga  $w = 0$  para todas as instâncias e todos os tipos de veículos. Contudo, para a análise do HFVRPTW-LC o custo de carga foi incluído ( $w \neq 0$ ) e definido para cada tipo de veículo e instância na Seção 4.5.1.

#### 4.3 DEFINIÇÃO DE PARÂMETROS DO ALGORITMO

Os parâmetros utilizados no método proposto (ALNS) foram definidos com valores sugeridos em outros trabalhos presentes na literatura que utilizam o mesmo algoritmo e, de maneira experimental, alguns deles foram ajustados para o cenário proposto neste trabalho.

Os refinamentos e ajustes dos parâmetros foram realizados por meio de diversas execuções do algoritmo para um conjunto de instâncias de diferentes categorias selecionadas do conjunto de instâncias definidos na Seção 4.2. A cada execução, novos valores eram definidos para o conjunto de parâmetros e, ao final, eram anali-

sados os impactos causados pelas mudanças nos diferentes cenários propostos pelo *benchmark* em questão. Dessa forma, ao final do processo os melhores valores foram atribuídos ao conjunto de parâmetros e ao longo de toda a experimentação foram utilizados os mesmos valores.

Dado que ao longo do capítulo 3 foram descritas diversas equações que requerem parâmetros, nesta seção não serão retomadas as fórmulas destas equações, apenas os parâmetros a serem definidos. A Tabela 6 apresenta as definições dos parâmetros do algoritmo de solução inicial semi-paralela e do método ALNS implementados.

Para os experimentos realizados na primeira e na segunda etapa (HFVRPTW e HFVRPTW-LC, respectivamente), foram utilizados os mesmos valores de parâmetros, de modo a não criar possíveis vieses e manter condições semelhantes de execução para ambos os experimentos.

#### 4.4 RESULTADOS DOS EXPERIMENTOS PARA *HFVRPTW*

Nesta seção são apresentados os resultados dos experimentos acerca do método de solução proposto para o HFVRPTW utilizando as instâncias de *benchmark* de Jiang et al. (2014). Para tal, é realizada uma análise comparativa entre a solução proposta neste trabalho e a solução proposta em Molina, Salmeron e Eguia (2020), a qual soluciona o mesmo problema utilizando um ACS-VNTS (*Ant Colony System - Variable Neighborhood Tabu Search*). Para quantificar as diferenças observadas na comparação, foi calculada a métrica *RPD* e a significância estatística pelo método de Wilcoxon, conforme descrito na Seção 4.1 e apresentado na Seção 4.4.1.

##### 4.4.1 Resultados

A Tabela 7 apresenta os resultados para todas as instâncias de teste. Nesta tabela, para cada instância de teste são apresentados os resultados de Molina, Salmeron e Eguia (2020) (ACS-VNTS), tais como o custo fixo (CF), custo variável (CV) e custo total (CT) da melhor solução obtida, o qual é a soma dos dois anteriores. São apresentados os resultados correspondentes para o método proposto neste trabalho (ALNS), bem como o número de veículos utilizados (NV). Além disso, é apresentado o valor de *RPD* para cada instância de forma a comparar o custo total (CT) entre as duas soluções. Por fim, são registradas estatísticas auxiliares referentes às 30 execuções do método proposto para cada instância de teste, tais como a média e desvio padrão do custo total (Avg. CT e Dev. CT), e ainda a média dos tempos de execução de cada instância (Avg. T(s)). Os resultados em destaque, que são os melhores valores de custo total e os valores de *RPD* em que houve melhoria, estão destacados em

Tabela 6 – Definição dos parâmetros

Parâmetro	Valor	Descrição
$\alpha_1$	0,6	Importância do custo na função de inserção da Solução Inicial Semi-Paralela.
$\alpha_2$	0,4	Importância do espaço vazio no veículo na função de inserção da Solução Inicial Semi-Paralela.
$itMax$	50000	Número máximo de iterações.
$t$	4	Fator de resfriamento da temperatura.
$\theta_1$	20	Pontuação dos operadores quando uma solução melhor global é encontrada.
$\theta_2$	10	Pontuação dos operadores quando uma solução melhor local é encontrada.
$\theta_3$	2	Pontuação dos operadores quando uma solução de pior custo é aceita.
$\rho$	0,1	Fator de reação, que controla o quanto rápido o ajuste dos pesos é feito com base no histórico atual.
$\kappa$	100	Número de iterações necessárias para aplicar a atualização dos pesos dos operadores.
$q_{min}$	4	Número mínimo de clientes a serem removidos pelo operador de remoção.
$q_{max}$	100	Número máximo de clientes a serem removidos pelo operador de remoção.
$\delta$	0,4	Parâmetro de controle da quantidade de clientes a serem removidos pelo operador de remoção.
$\gamma$	0,025	Grau de ruído para os operadores de inserção.
$\sigma$	3	Grau de ruído para operador de remoção.

Fonte: Próprio Autor

negrito. Ao final da Tabela 7 é apresentado o resumo dos resultados, como o número de melhores resultados para cada método de solução e o valor médio de RPD.

Dados os resultados apresentados na Tabela 7, observa-se que a solução proposta (ALNS) apresentou melhora no custo total em 51 das 55 instâncias de teste, o que corresponde a melhorias em mais de 92% das instâncias analisadas, com uma redução média de 1,48% do custo e uma média de tempo de execução de 15,68 segundos. Já a solução de Molina, Salmeron e Eguia (2020) (ACS-VNTS), manteve menores custos em apenas 4 de 55 instâncias. Em uma das instâncias ambos obtiveram o mesmo valor de TC, resultando em um  $RPD = 0\%$ . Além disso, foi registrado o nível de ocupação médio dos veículos considerando todas as instâncias de teste, o qual resultou em 95,7% de ocupação, o que indica que os veículos estão operando com praticamente suas capacidades máximas.

Tabela 7 – Tabela de Resultados HFVRPTW

Instância	ACS-VNTS (2020)			ALNS (2023)					Estatísticas - 30 execuções		
	CF	CV	CT	CF	CV	CT	NV	RPD	Avg. CT	Dev. CT	Avg. T(s)
HC101	670	1215,33	1885,33	670	1201,87	<b>1871,87</b>	14/25	<b>-0,71 %</b>	1924,35	26,24	10,80
HC102	670	1154,59	<b>1824,59</b>	650	1193,96	1843,96	15/25	1,06 %	1894,46	31,00	12,63
HC103	650	1201,65	1851,65	670	1143,38	<b>1813,38</b>	14/25	<b>-2,07 %</b>	1879,40	38,86	13,93
HC104	670	1120,54	1790,54	670	1114,72	<b>1784,72</b>	14/25	<b>-0,32 %</b>	1822,14	30,64	14,80
HC105	670	1180,70	1850,70	670	1164,84	<b>1834,84</b>	14/25	<b>-0,86 %</b>	1897,50	25,45	11,73
HC106	670	1194,49	1864,49	670	1183,19	<b>1853,19</b>	14/25	<b>-0,61 %</b>	1932,33	32,56	12,00
HC107	670	1169,45	1839,45	670	1164,03	<b>1834,03</b>	14/25	<b>-0,29 %</b>	1897,87	22,92	12,07
HC108	670	1150,74	1820,74	670	1136,89	<b>1806,89</b>	14/25	<b>-0,76 %</b>	1883,70	26,74	13,00
HC109	670	1121,71	1791,71	670	1119,65	<b>1789,65</b>	14/25	<b>-0,12 %</b>	1855,11	25,45	13,77
HC201	520	793,25	1313,25	500	760,24	<b>1260,24</b>	5/25	<b>-4,04 %</b>	1326,10	28,31	14,90
HC202	500	758,83	1258,83	500	701,37	<b>1201,37</b>	5/25	<b>-4,56 %</b>	1293,56	26,78	19,50
HC203	540	717,36	1257,36	500	695,03	<b>1195,03</b>	5/25	<b>-4,96 %</b>	1282,68	24,90	23,13
HC204	500	752,08	1252,08	500	713,20	<b>1213,20</b>	5/25	<b>-3,11 %</b>	1276,77	21,12	27,67
HC205	540	779,88	1319,88	500	765,29	<b>1265,29</b>	5/25	<b>-4,14 %</b>	1314,98	26,80	16,00
HC206	540	722,39	1262,39	500	710,31	<b>1210,31</b>	5/25	<b>-4,13 %</b>	1269,03	24,87	17,73
HC207	540	721,39	1261,39	500	718,38	<b>1218,38</b>	5/25	<b>-3,41 %</b>	1285,12	21,02	18,00
HC208	500	686,67	1186,67	500	686,67	<b>1186,67</b>	5/25	<b>0,00 %</b>	1199,83	11,78	18,43
HR101	2760	2209,52	<b>4969,52</b>	2920	2063,19	4983,19	20/35	0,28 %	5098,93	58,87	11,00
HR102	3040	1874,35	4914,35	2830	1948,90	<b>4778,90</b>	19/35	<b>-2,76 %</b>	4833,05	25,47	12,00
HR103	2790	1857,81	4647,81	2810	1743,35	<b>4553,35</b>	18/35	<b>-2,03 %</b>	4593,03	17,05	12,97
HR104	2740	1650,16	4390,16	2680	1659,27	<b>4339,27</b>	19/35	<b>-1,16 %</b>	4376,65	20,59	14,00
HR105	2700	1824,06	4524,06	2540	1890,75	<b>4430,75</b>	22/35	<b>-2,06 %</b>	4477,11	25,48	11,87
HR106	2590	1796,13	4386,13	2540	1819,68	<b>4359,68</b>	22/35	<b>-0,60 %</b>	4399,24	20,41	12,93
HR107	2680	1702,37	4382,37	2520	1746,16	<b>4266,16</b>	21/35	<b>-2,65 %</b>	4313,32	25,69	13,63
HR108	2540	1709,14	4249,14	2540	1679,11	<b>4219,11</b>	22/35	<b>-0,71 %</b>	4261,92	26,50	14,07
HR109	2540	1770,30	4310,30	2540	1758,83	<b>4298,83</b>	22/35	<b>-0,27 %</b>	4326,80	19,02	12,90
HR110	2540	1522,31	<b>4062,31</b>	2540	1698,36	4238,36	22/35	4,33 %	4286,71	20,60	13,30
HR111	2650	1678,41	4328,41	2540	1746,60	<b>4286,60</b>	22/35	<b>-0,97 %</b>	4315,85	15,62	13,37
HR112	2540	1694,62	4234,62	2520	1676,74	<b>4196,74</b>	21/35	<b>-0,89 %</b>	4249,01	22,70	14,10
HR201	250	1391,31	1641,31	270	1311,56	<b>1581,56</b>	6/15	<b>-3,64 %</b>	1656,09	37,33	15,70
HR202	270	1184,66	1454,66	225	1174,53	<b>1399,53</b>	5/15	<b>-3,79 %</b>	1456,69	36,23	18,63
HR203	270	1053,08	1323,08	225	1022,22	<b>1247,22</b>	5/15	<b>-5,73 %</b>	1292,06	27,06	22,10
HR204	250	858,79	1108,79	225	828,70	<b>1053,70</b>	5/15	<b>-4,97 %</b>	1100,59	28,90	24,73
HR205	270	989,74	1259,74	225	1010,46	<b>1235,46</b>	5/15	<b>-1,93 %</b>	1253,59	8,86	18,00
HR206	225	953,20	1178,20	225	923,12	<b>1148,12</b>	5/15	<b>-2,55 %</b>	1170,22	9,95	20,07
HR207	225	867,23	1092,23	225	842,69	<b>1067,69</b>	5/15	<b>-2,25 %</b>	1086,66	8,63	22,40
HR208	225	772,52	997,52	225	759,48	<b>984,48</b>	5/15	<b>-1,31 %</b>	997,81	7,66	24,97
HR209	225	891,98	1116,98	225	873,04	<b>1098,04</b>	5/15	<b>-1,70 %</b>	1115,29	9,94	19,40
HR210	225	1047,79	1272,79	225	1025,35	<b>1250,35</b>	5/15	<b>-1,76 %</b>	1300,10	24,53	20,07
HR211	225	777,71	1002,71	225	772,20	<b>997,19</b>	5/15	<b>-0,55 %</b>	1013,31	8,50	21,87
HRC101	3270	2342,57	<b>5612,57</b>	3270	2361,22	5631,22	22/40	0,33 %	5720,14	42,61	12,10
HRC102	3360	2189,20	5549,20	3360	2144,40	<b>5504,40</b>	19/40	<b>-0,81 %</b>	5577,04	35,18	12,60
HRC103	3390	2034,35	5424,35	3360	2031,64	<b>5391,64</b>	18/40	<b>-0,60 %</b>	5459,45	32,07	13,03
HRC104	3390	1916,52	5306,52	3390	1822,57	<b>5212,57</b>	16/40	<b>-1,77 %</b>	5276,06	46,29	14,00
HRC105	3390	2249,81	5639,81	3300	2248,31	<b>5548,31</b>	21/40	<b>-1,62 %</b>	5625,33	43,34	12,00
HRC106	3390	2047,39	5437,39	3240	2194,46	<b>5434,46</b>	22/40	<b>-0,05 %</b>	5504,58	39,28	12,40
HRC107	3390	1971,36	5361,36	3330	1985,40	<b>5315,40</b>	20/40	<b>-0,86 %</b>	5396,84	34,03	13,00
HRC108	3450	1831,05	5281,05	3360	1848,42	<b>5208,42</b>	18/40	<b>-1,38 %</b>	5284,50	42,44	14,00
HRC201	2900	1600,72	4500,72	2900	1589,21	<b>4489,21</b>	14/15	<b>-0,26 %</b>	4513,37	16,81	13,07
HRC202	2950	1420,70	4370,70	2900	1453,73	<b>4353,73</b>	14/15	<b>-0,39 %</b>	4392,64	20,17	14,50
HRC203	2900	1412,26	4312,26	2900	1395,52	<b>4295,52</b>	14/15	<b>-0,39 %</b>	4315,91	13,97	15,10
HRC204	2950	1308,25	4258,25	2900	1339,79	<b>4239,79</b>	14/15	<b>-0,43 %</b>	4261,15	12,80	15,97
HRC205	2900	1545,61	4445,61	2900	1510,85	<b>4410,85</b>	14/15	<b>-0,78 %</b>	4434,40	16,30	14,03
HRC206	2900	1488,86	4388,86	2900	1467,50	<b>4367,50</b>	14/15	<b>-0,49 %</b>	4407,22	21,90	14,23
HRC207	2900	1416,65	4316,65	2900	1411,32	<b>4311,32</b>	14/15	<b>-0,12 %</b>	4344,48	17,70	15,03
HRC208	2900	1361,43	4261,43	2900	1332,16	<b>4232,16</b>	14/15	<b>-0,69 %</b>	4257,49	16,78	15,97

<b>Melhores</b>	4 / 55	51 / 55	-1,48 %		
-----------------	--------	---------	---------	--	--

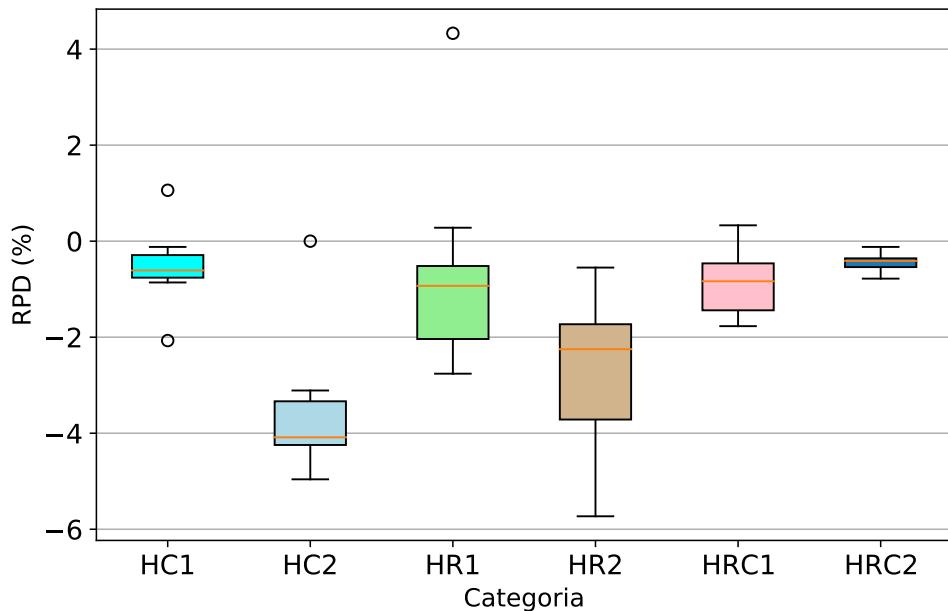
Fonte: Próprio Autor

Ao observar os resultados, nota-se que para a categoria HC1, apenas uma instância não conseguiu *RPD* negativo (HC102), com uma piora de 1,06%. Para a categoria HC2, todas as instâncias obtiveram melhoria, com exceção da instância HC208, a qual obteve  $RPD = 0\%$ . Dentre as instâncias da categoria HR1 apenas as instâncias HR101 e HR110 não obtiveram melhoria, com *RPD* de 0,28% e 4,33%, respectivamente. Para o caso HR2, todas as instâncias alcançaram melhorias. Já para as categorias HRC1 e HRC2, todas as instâncias obtiveram melhorias, exceto a instância HRC101, com *RPD* de 0,33%.

A Figura 16 apresenta um diagrama de caixa da comparação entre o *RPD*(%) para cada uma das 6 categorias de instâncias presentes no *benchmark*. A categoria HC1 obteve média de *RPD* de  $-0,52\%$ , enquanto a categoria HC2 alcançou  $-3,54\%$ , a categoria HR1 obteve  $-0,79\%$ , HR2 obteve  $-2,74\%$ , HRC1 obteve  $-0,85\%$  e por fim a categoria HRC2 obteve  $-0,44\%$  de *RPD* médio. Com isso, é possível observar que todas as categorias alcançaram *RPD* médio negativo. Ainda, verifica-se que as instâncias das categorias HC2 e HR2 obtiveram maior redução de *RPD* em média, em comparação com as demais categorias. Isso indica uma maior facilidade do método proposto em lidar com soluções de rotas longas (horizonte de tempo maior). Além disso, é possível perceber que a categoria HRC2 obteve redução média de *RPD* não significativa em comparação com as outras duas categorias de tempo longo (HC2 e HR2), isso pode ter ocorrido pela dificuldade em lidar com instâncias com distribuição randômica e em *cluster* ao mesmo tempo, ou até mesmo pela possibilidade de estarem se aproximando dos melhores resultados globais para estas instâncias, ou seja, sem muito espaço para melhorias em comparação com os resultados de Molina, Salmeron e Eguia (2020). É importante salientar que para todos os resultados obtidos nos experimentos, todos os clientes foram atendidos e por este motivo nenhuma análise específica será realizada acerca de possíveis não atendimentos.

Para verificar a significância das melhorias obtidas com relação ao trabalho relacionado, foi aplicado o teste de Wilcoxon. Este teste busca refutar ou aceitar a hipótese nula e mostrar se os resultados obtidos são significativamente equivalentes ou diferentes. Com isso, foi executado o teste sobre a métrica de Custo Total (CT) para ambas as soluções comparadas, a qual reflete o valor da função objetivo das soluções. Ao aplicar o teste comparando a solução proposta neste trabalho com o trabalho de Molina, Salmeron e Eguia (2020), foi obtido um  $p - valor = 2,6 \times 10^{-8}$  para um nível de confiança de 99% ( $p < 0,01$ ). Portanto, como o  $p - valor$  obtido é menor que 0,01, é possível refutar a hipótese nula e, portanto, constatar que ambas as soluções são significativamente diferentes. Assim, pode-se atestar que a solução proposta neste trabalho (ALNS) obteve melhoria perante os resultados de Molina, Salmeron e Eguia (2020).

Figura 16 – Diagrama de Caixa RPD (%)



Fonte: Próprio Autor

Fazendo uma análise mais profunda sobre os resultados obtidos em relação ao *RPD*, eles podem estar associados à utilização do operador de remoção *Route Removal* e a utilização do método de construção semi-paralelo, utilizado na solução inicial e como extensão dos métodos de inserção para casos de criação de novas rotas, contribuições propostas neste trabalho. De fato, a utilização destes métodos faz com que haja destruição de rotas inteiras pelo *Route Removal* e ao tentar reinseri-las, caso os clientes removidos não possam ser inseridos nas rotas remanescentes, utiliza-se o método semi-paralelo para a criação de novas rotas com veículos mais apropriados. Isso permite a modificação de tipos de veículos das rotas, o que é fundamental para a exploração de regiões promissoras do espaço de busca para problemas com frota heterogênea limitada.

#### 4.4.2 Tempo de Execução

Em termos de tempo de processamento, o método proposto obteve 15,68 segundos de tempo médio de execução, executado em um processador *Intel(R) Core(TM) i7-8565U CPU @ 1.80 GHz (Max Turbo Frequency 4.60 GHz)*, enquanto Molina, Salmeron e Eguia (2020) obteve tempo médio de execução de 797,40 segundos, segundo os próprios autores, executado em um processador *Intel(R) Core(TM) i5-2400 CPU @ 3.10 GHz (Max Turbo Frequency 3.40 GHz)*. Com base na frequência máxima dos processadores utilizados em ambos os trabalhos, nota-se que o processador utilizado no presente trabalho é cerca de 35% mais rápido que o utilizado por Molina, Salmeron e Eguia (2020).

Ao reduzir 35% do tempo médio de execução alcançado por Molina, Salmeron e Eguia (2020), obtém-se um tempo médio de 518,31 segundos. Portanto, considerando condições de processamento semelhantes, pode-se observar que o método proposto possui tempo médio de execução aproximadamente 33 vezes menor que o método de Molina, Salmeron e Eguia (2020). Essa diferença de tempo médio de execução pode ser explicada pelo fato do método proposto ser baseado no algoritmo ALNS, que é uma meta-heurística de melhoria de solução única, enquanto o de Molina, Salmeron e Eguia (2020) é baseado no ACS-VNTS, que é uma meta-heurística de construção populacional, baseada em colônia de formigas híbrida com um algoritmo de melhoria baseado em busca tabu, o que leva a uma maior quantidade de avaliações da função objetivo e verificações de restrições, consequentemente, maior tempo de processamento.

#### 4.4.3 Análise do Comportamento do Método Proposto

Com o objetivo de analisar o comportamento do algoritmo, são apresentados gráficos de peso dos operadores e convergência média para cada uma das 6 categorias de instâncias de *benchmark* (HC1, HC2, HR1, HR2, HRC1, HRC2). Para isso, foi escolhida uma instância representativa de cada categoria e seus respectivos gráficos de convergência e pesos dos operadores de remoção e inserção. O gráfico de convergência apresenta a média das 30 execuções para a instância escolhida, já os gráficos de pesos de operadores representam apenas a primeira execução da mesma, dado que as demais execuções tem comportamento semelhante. Esta análise visa entender a diferença de comportamento entre os diferentes cenários de instâncias.

Para tornar a discussão e análise mais clara, é importante resgatar os conceitos básicos de cada operador utilizado e de que forma ele impacta na solução. Os operadores de remoção são: *Random Removal*, o qual remove um conjunto de clientes da solução atual de maneira aleatória; *Worst Removal*, que remove os clientes que possuem o maior impacto no custo total; *Shaw Removal*, o qual remove os clientes por similaridade por distância entre si; e o *Route Removal*, que remove 40% da frota utilizada, ou seja, são removidas rotas completas da solução. Já os operadores de inserção são: *Greedy Insertion*, o qual insere os clientes de forma gulosa na rota e posição de menor custo; e *Regret-k Insertion*, que insere os clientes segundo critério de arrependimento com base no custo de inserção. Já a convergência é dada pelas curvas de melhor solução local da iteração atual, melhor solução global dentre todas as iterações e o resfriamento da temperatura. O gráfico é dado pelo custo total da solução em função das iterações. Dito isso, a análise da convergência e dos operadores são embasadas segundo as Figuras 17 - 22. As instâncias de teste utilizadas para representar cada categoria são: HC108, HC206, HR105, HR211, HRC102, HRC208.

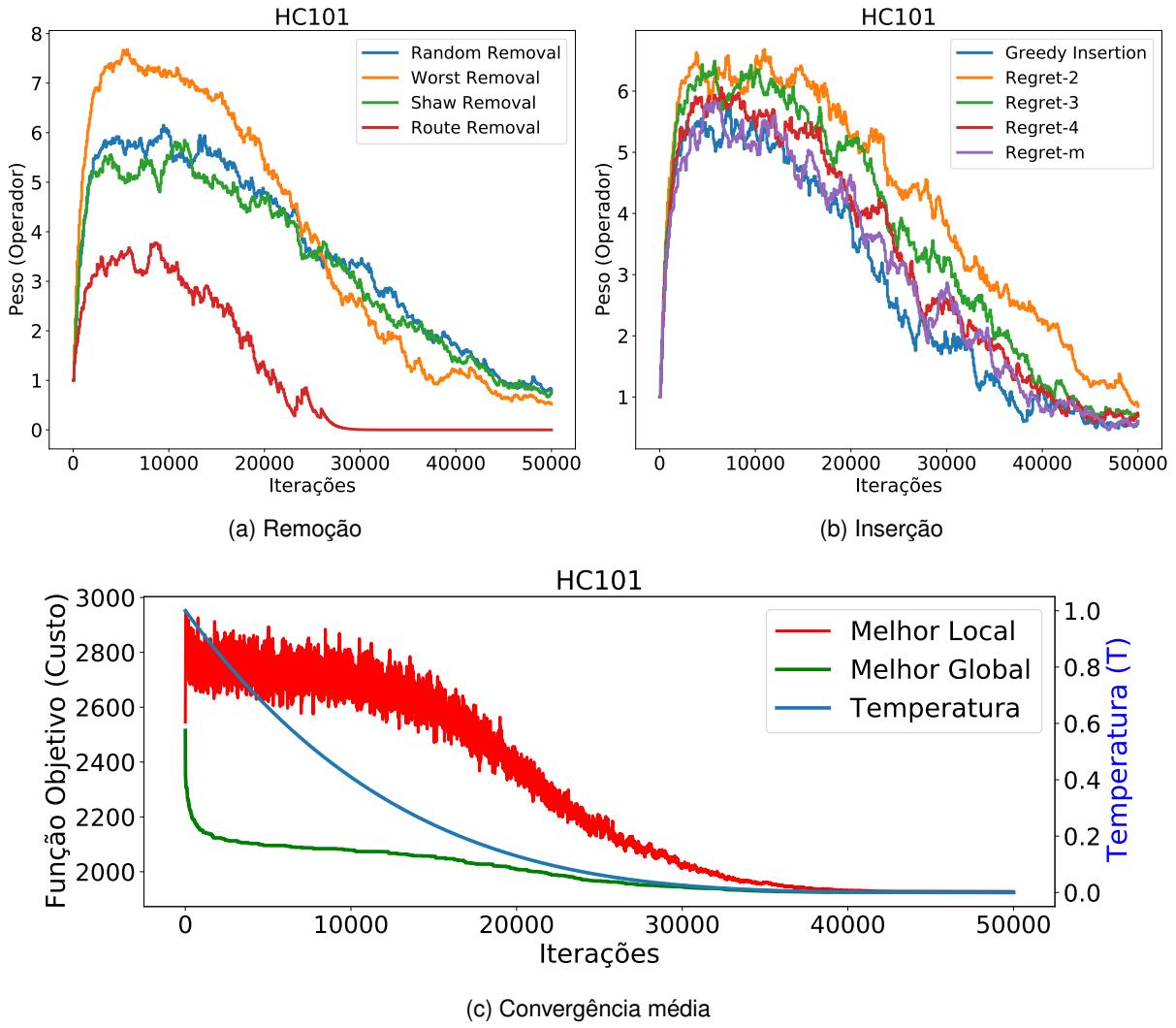


Figura 17 – Gráficos convergência e de pesos dos operadores de remoção e inserção (HC101)

Conforme os gráficos de pesos dos operadores apresentados, de forma geral, os pesos crescem de maneira abrupta no início da busca, pois é nessa fase que eles apresentam melhor efeito na solução e então recebem as maiores pontuações. Esse comportamento se justifica pois, no início da otimização, há mais espaço para alcançar melhorias por meio dos operadores. Com o passar das iterações, o processo de convergência começa a se estagnar e os operadores não apresentam mais alterações de impacto positivo na solução, perdendo sua relevância, fazendo o fator de reação histórica  $\rho$  diminuir seus pesos gradativamente. Já a convergência é dada por uma alta exploração de soluções nas iterações iniciais dada a alta temperatura, e uma intensificação e estagnação nas iterações finais.

Ao observar o comportamento dos pesos de modo geral nos gráficos das Figuras 17 - 22, nota-se que alguns operadores se destacam em comparação com outros. Isso ocorre pois estes operadores se mostram mais relevantes e eficientes ao serem utilizados em detrimento dos outros operadores. Desse modo, quando um operador é

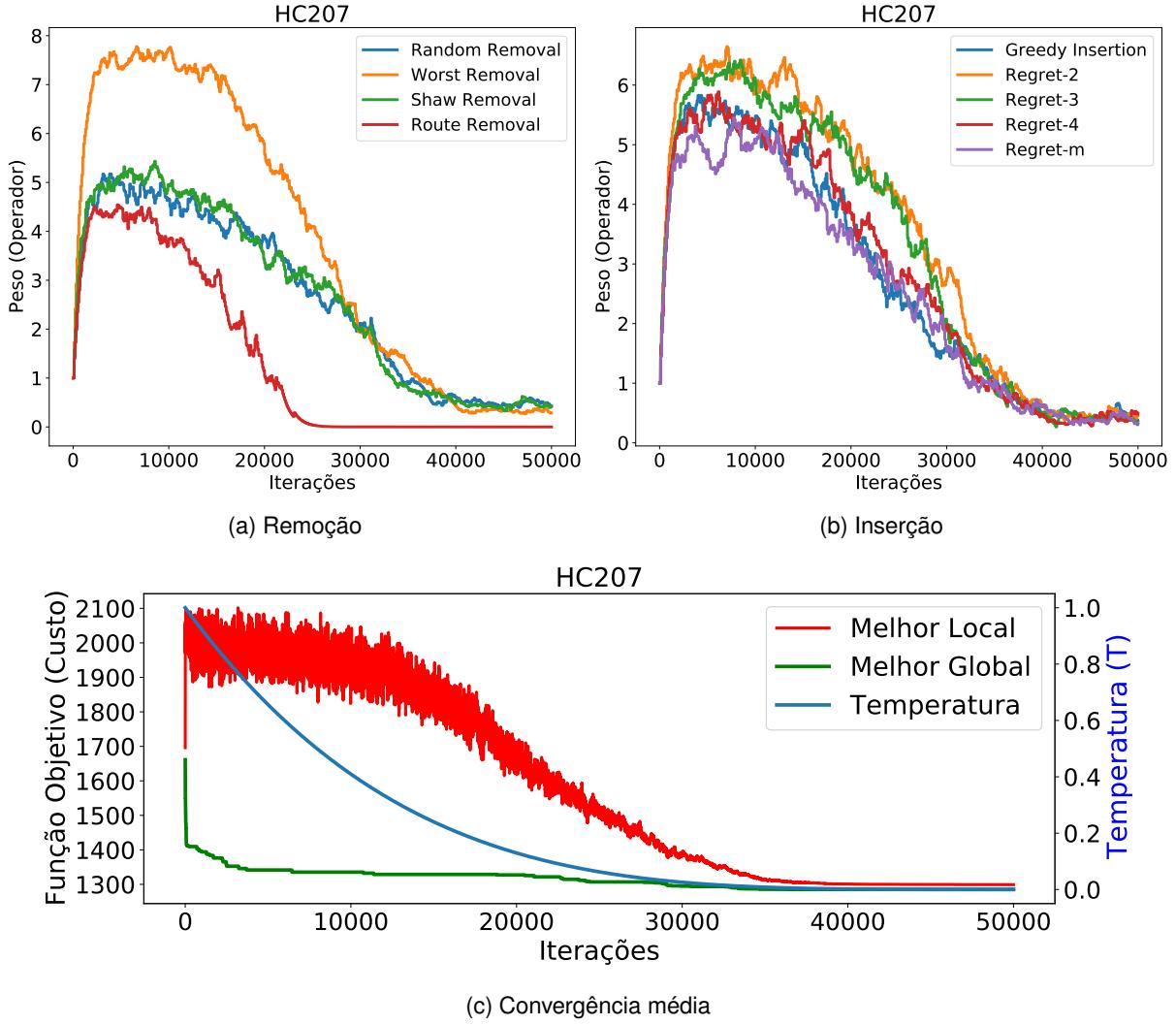


Figura 18 – Gráficos convergência e de pesos dos operadores de remoção e inserção (HC207)

utilizado com frequência e causa melhorias na solução, gera um aumento de peso e aumenta a probabilidade de utilizá-lo na iterações seguintes, caso contrário este perde peso e, por consequência, relevância para a sequência da otimização.

Diante desta premissa, percebe-se que em alguns casos, determinado operador assume posição de maior relevância na busca, o que demonstra o caráter adaptativo do algoritmo e que o método agiu de forma mais intensa em uma região mais específica do espaço de busca, liderada pelo operador em destaque. Em contrapartida, em alguns casos as curvas dos operadores permanecem próximas e equilibradas com alto valor de peso, o que significa que estes operadores contribuíram de forma semelhante para o desempenho da solução ao longo da busca e que o algoritmo conseguiu fazer uma exploração maior de diferentes regiões do espaço de busca.

É importante salientar que o comportamento das curvas de pesos não refletem necessariamente o bom ou mal desempenho final em relação a otimização da função objetivo. Estes comportamentos significam que, a depender do cenário de

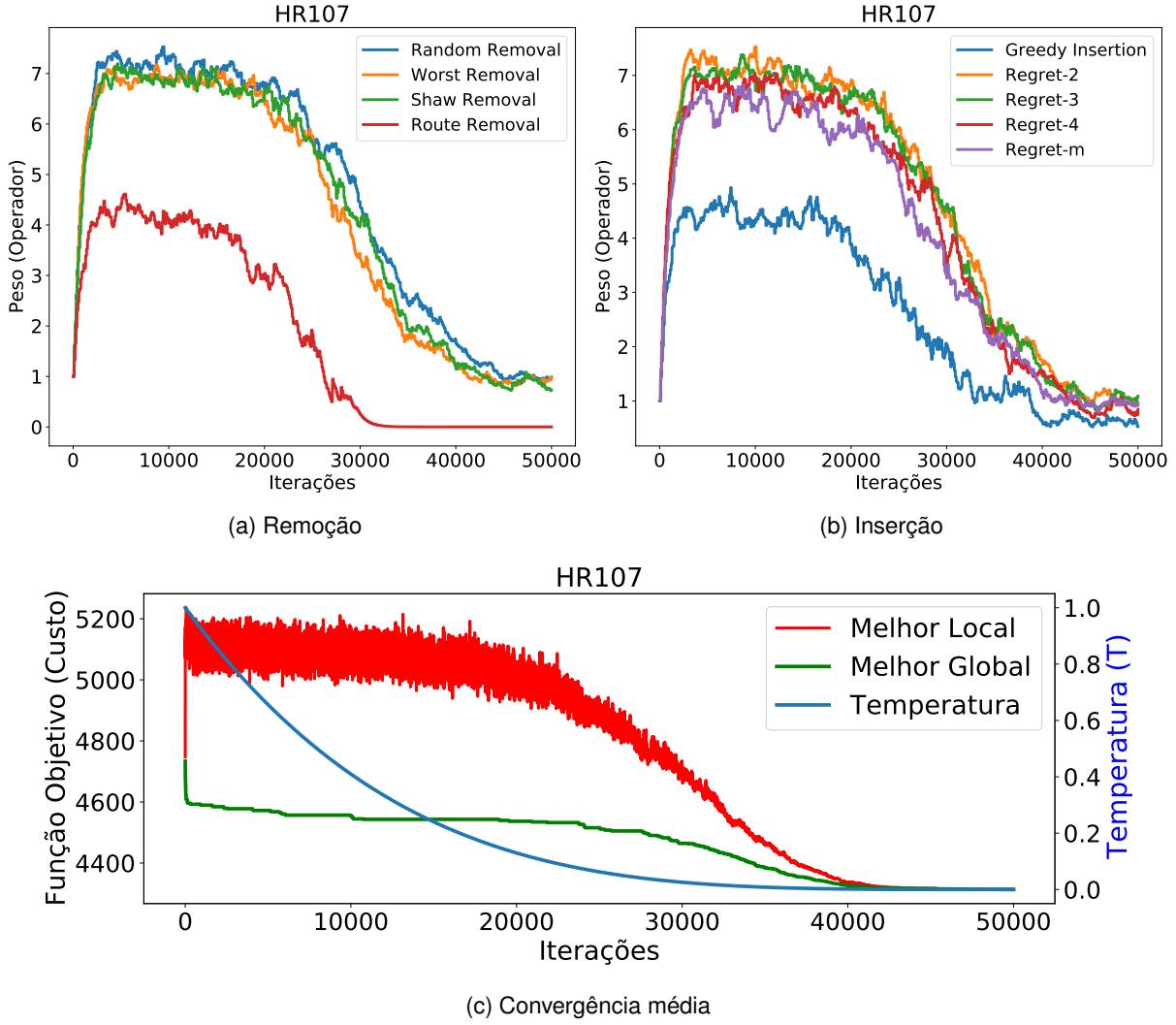


Figura 19 – Gráficos convergência e de pesos dos operadores de remoção e inserção (HR107)

teste, o algoritmo explorou e/ou intensificou o processo de busca em regiões do espaço de soluções. Cada caso de teste pode ter seu comportamento específico dada a distribuição dos clientes, disposição das janelas de tempo e demais características do cenário de otimização. Desse modo, o algoritmo mostrou capacidade de explorar/diversificar a busca e também de intensificá-la dependendo de cada cenário, o que mostra versatilidade e capacidade de adaptação a diferentes casos.

Ao analisar de modo geral a atuação dos operadores de remoção, nota-se que nas categorias de teste HC1, HC2, HR2, HRC2 (Figuras 17a, 18a, 20a e 22a), o operador de maior destaque foi o *Worst Removal*, o que demonstra que a busca foi intensificada por este operador em uma região mais específica do espaço de busca. Os operadores *Random Removal* e *Shaw Removal* se mantiveram equilibrados entre si para todas as categorias de teste. Para os cenários HR1 e HRC1 (Figuras 19a e 21a) os operadores *Worst*, *Random* e *Shaw Removal* se equilibraram durante todo o processo de otimização, o que indica que foi realizada uma busca mais diversifi-

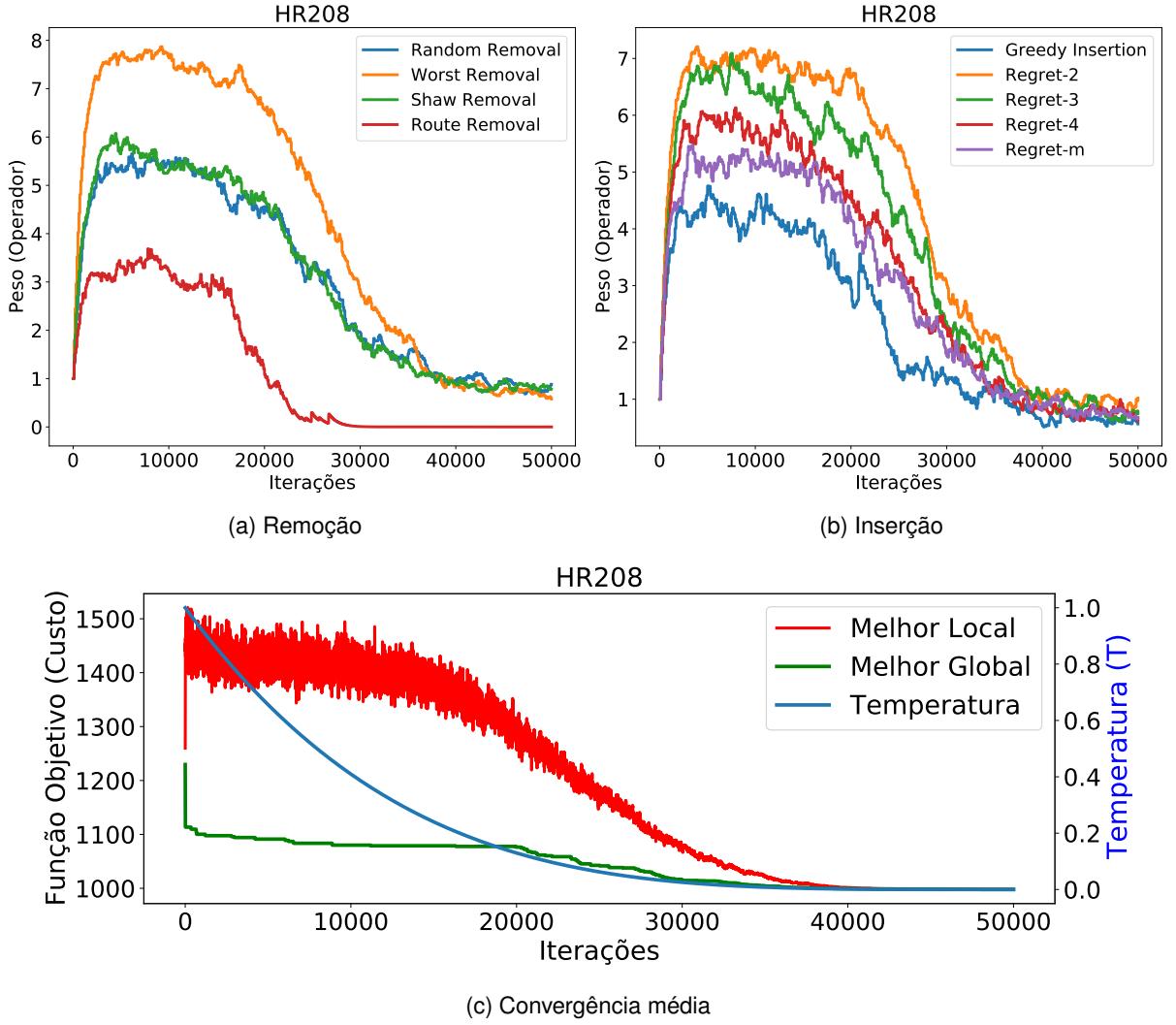


Figura 20 – Gráficos convergência e de pesos dos operadores de remoção e inserção (HR208)

cada utilizando estes operadores. Para todos os cenários, o operador proposto neste trabalho *Route Removal* obteve relevância maior nas iterações iniciais e teve uma diminuição de peso considerável nas iterações finais. O comportamento deste operador reflete o propósito da sua implementação, a qual se baseia em dar mais diversificação no processo de busca para sair de mínimos locais por meio da remoção de 40% das rotas da solução, o que permite que na fase de inserção de clientes removidos o algoritmo possa reconstruir uma parcela considerável da solução e assim atingir regiões distintas do espaço de busca.

Em relação aos operadores de inserção, *Greedy Insertion* se mostrou com pior relevância em comparação com as variações do operador *Regret-k Insertion* para os cenários HR1, HR2, HRC1 e HRC2 (Figuras 19b, 20b, 21b e 22b). Em contrapartida, para as instâncias do tipo HC1 e HC2 (Figuras 17b e 18b), o operadores *Greedy* e *Regret-k Insertion* obtiveram certo equilíbrio, o que demonstra maior exploração de opções de inserção nestes cenários. Contudo, em todos os cenários os métodos de

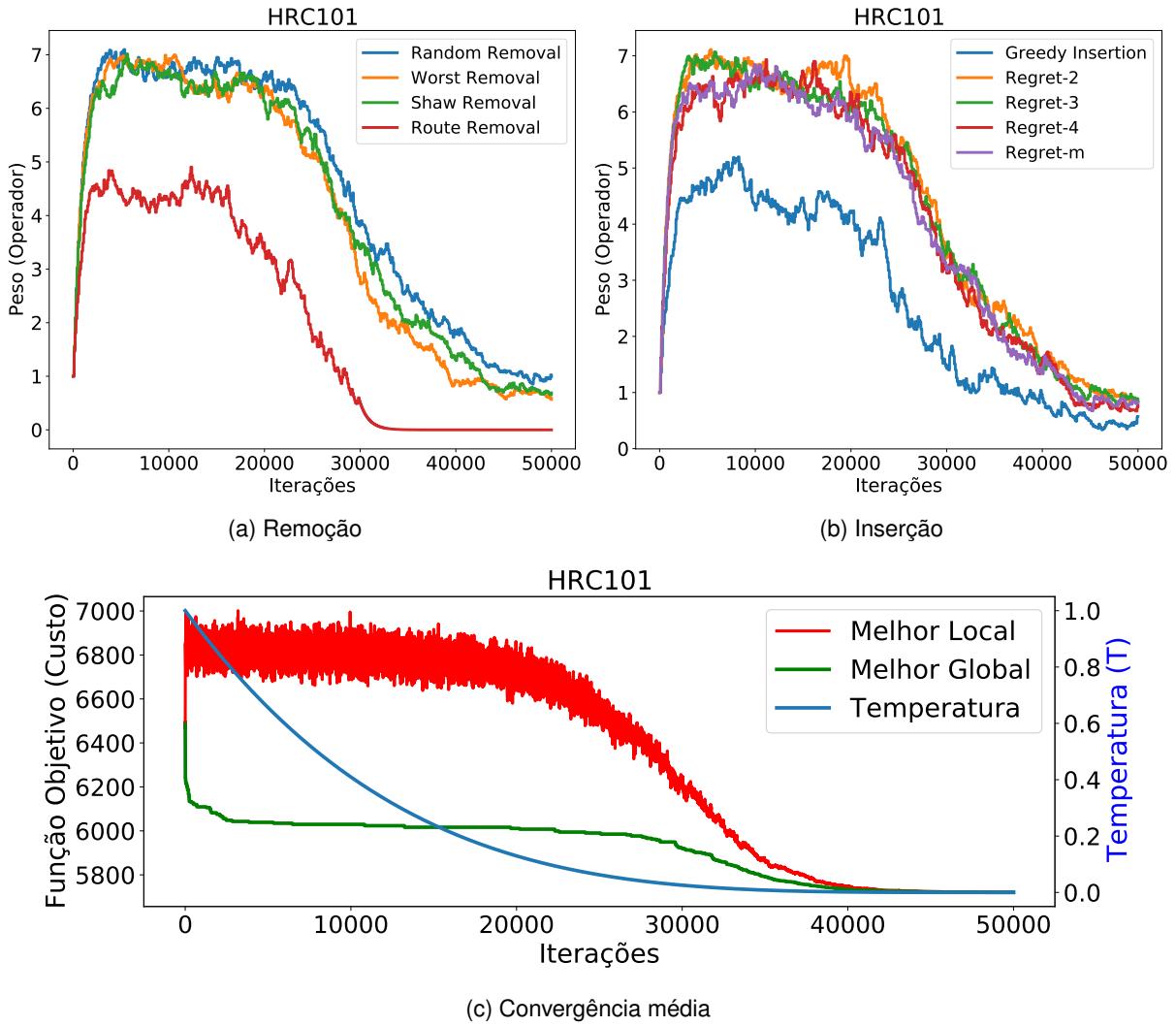


Figura 21 – Gráficos convergência e de pesos dos operadores de remoção e inserção (HRC101)

*Regret-k Insertion* obtiveram maior relevância em detrimento do *Greedy Insertion*, em especial os métodos de *Regret-2* e *Regret-3*. Isso pode se dar devido ao fato de que os métodos *regret* foram propostos para melhorar o desempenho proposto pelo *Greedy Insertion*. Assim, espera-se que naturalmente os métodos *regrets* tenham relevância igual ou superior ao método *greedy*.

De modo geral, nota-se que para os cenários onde há maior equilíbrio entre os operadores (HR1 e HRC1), o gráfico de convergência demonstra uma queda mais lenta do valor da melhor solução global, o que mostra que foi realizada uma exploração maior nas iterações iniciais causada pelo uso de forma semelhante de vários operadores. Em contrapartida, nos cenários em que um operador se destaca mais que outros (HC1, HC2, HR2 e HRC2), a convergência aponta uma queda mais abrupta e estagnação precoce da melhor solução global, causado pelo uso acentuado de um operador específico, o que leva a uma busca mais localizada e intensa em uma região específica do espaço de soluções.

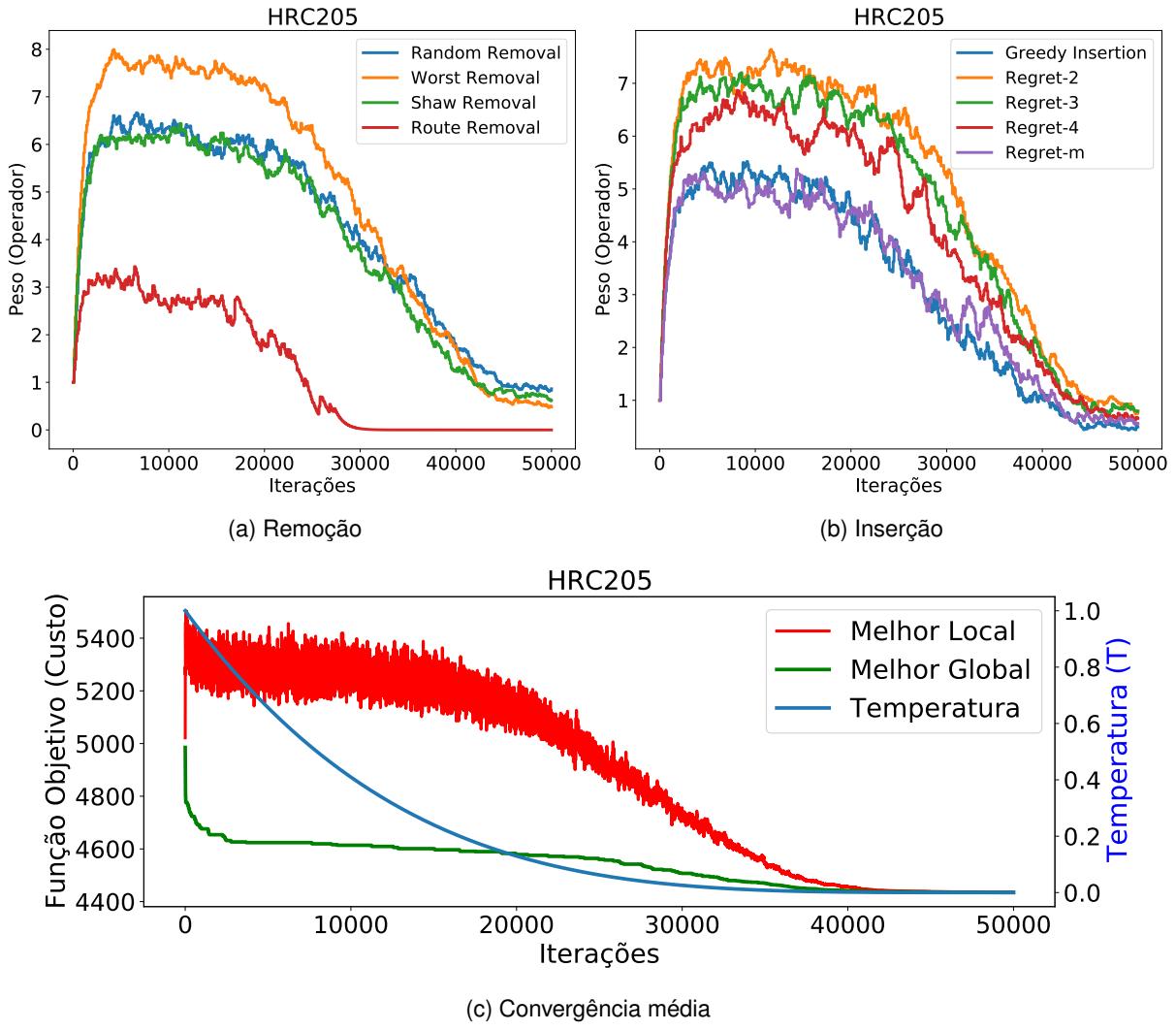


Figura 22 – Gráficos convergência e de pesos dos operadores de remoção e inserção (HRC205)

Dadas as análises realizadas acerca dos operadores de remoção e inserção, observa-se que, embora o desempenho em relação a função objetivo tenha sido satisfatória e significante perante a literatura, em grande parte das categorias de teste há um desequilíbrio em relação aos operadores de remoção, dada a maior relevância do método *Worst Removal* perante os demais operadores, o que causa uma forte intensificação e menor diversificação na fase inicial da otimização, segundo os gráficos de convergência, causando precoce estagnação. Isso pode significar que o algoritmo ainda carece de maior variação de operadores, para assim tentar uma maior diversificação no processo de busca. Nota-se também que a utilização de operadores de inserção de natureza diferente dos métodos utilizados, os quais se baseiam sempre no ganho de custo, ou até mesmo operadores de troca direta de posições, podem trazer maior diversidade para o leque de opções de operadores de inserção. Além disso, realizar testes com outros cenários diversos podem evidenciar outros comportamentos do algoritmo e trazer novas carências a serem exploradas.

## 4.5 RESULTADOS DOS EXPERIMENTOS PARA HFVRPTW-LC

Nesta seção são apresentados os resultados dos experimentos realizados para o HFVRPTW-LC utilizando as mesmas 56 instâncias de teste utilizadas na primeira etapa dos experimentos, com o HFVRPTW. Para analisar o desempenho do algoritmo para este problema é proposta uma comparação entre os resultados sem levar em conta o custo de carga (*loading cost*) e os resultados considerando esta característica na função objetivo. Entretanto, para que isso seja possível, dado que a literatura não apresenta uma definição de valores de custo de carga para as instâncias de Solomon, este trabalho faz um estudo aprofundado para definir os melhores valores para o coeficiente de custo de peso  $w$  para cada instância de teste, que são apresentados na Seção 4.5.1. A partir da definição dos valores de  $w$  para cada instância, o experimento final com os resultados para cada uma das 56 instâncias de teste são expostos na Seção 4.5.2. De acordo com os resultados, são realizadas análises de desempenho frente ao problema HFVRPTW-LC, especialmente observando os impactos gerados em considerar o custo de carga no método de otimização proposto (ALNS).

### 4.5.1 Definição do coeficiente de custo de carga ( $w$ )

A inclusão da noção do custo de carga no VRP trouxe consigo a ideia de que considerar este custo na função objetivo a ser otimizada traria uma redução de custos totais da operação, se comparados com uma otimização que não o considere. Considerar o custo de carga na otimização faz com que o método desenvolvido incorpore uma característica presente em cenários reais de transporte de carga e que é negligenciada pelas versões mais difundidas de VRP. Na prática, como introduzido na Seção 3.1, o custo de carga neste trabalho é representado por um coeficiente  $w$  por tipo de veículo, que é medido em unidade de peso por unidade de distância percorrida e faz parte do custo variável da operação, que é composto também pelo custo variável por distância. A existência deste custo no método de otimização faz com que sejam priorizadas entregas mais pesadas caso a distância compense esta priorização, o que torna o modelo mais complexo por ter que equilibrar estes dois fatores para encontrar uma solução de qualidade.

Dada a literatura acerca de VRPs que envolvem custo de carga, observou-se que, em geral, o coeficiente de carga  $w$  é definido com base em investigações do cenário local de transporte, ou seja, não há uma exploração aprofundada dos valores mais adequados para cada cenário. Além disso, não foi identificado um trabalho composto pelas mesmas características de VRP tratadas no presente trabalho: frota heterogênea limitada, janela de tempo e custo de carga, que definisse os valores de  $w$  para as instâncias de Solomon. A definição de forma apropriada do coeficiente  $w$

pode influenciar na diminuição dos custos de transporte e levar a soluções mais adequadas ao considerar o custo do peso transportado como um fator na função objetivo da otimização. Porém, a definição inapropriada deste coeficiente pode causar soluções com aumento na distância total percorrida, diminuição da ocupação dos veículos utilizados e até mesmo aumento do custo total da operação. Assim, como se trata de um coeficiente sensível, uma análise exploratória foi conduzida para a definição deste coeficiente  $w$  para cada instância de teste, que são apresentados na Tabela 9.

Diante do cenário de frota heterogênea limitada, proposto neste trabalho, é importante salientar que os coeficientes  $w$  de cada instância apresentados na Tabela 9 são utilizados como coeficiente base para a definição dos valores de  $w^p$  específico para cada tipo de veículo, segundo a definição matemática na Seção 3.1. Isso se deve ao fato do problema utilizar uma frota de veículos distintos, onde cada tipo receberá um valor distinto de  $w$ , que é proporcional a sua capacidade. Desse modo, o menor tipo de veículo recebe o valor base de  $w$  definido pelo estudo e os demais tipos de veículos recebem o valor  $w$  proporcional a capacidade. Por exemplo, supondo um  $w$  base igual a 0,04 definido pela análise exploratória para a instância HR101, o tipo de veículo A assume um  $w^p = 0,04$ , por ser o menor tipo de veículo em capacidade. Já os veículos do tipo B e C, recebem valores de  $w$  proporcionais a suas respectivas capacidades, como apresentado na Tabela 8.

Tabela 8 – Exemplo de  $w$  proporcional a capacidade (HR101)

Instância	Tipo	Qtd.	Capacidade	$w^p$
HR101	A	10	50	0,04 ( $w$ base)
	B	15	80	0,064
	C	10	120	0,096

Fonte: Próprio Autor

Optou-se por utilizar a proporção com base na capacidade como forma de buscar maior eficiência e rentabilidade da operação. Ao atribuir um custo mais alto para o maior tipo de veículo, o custo adicional de utilizar um veículo maior pode ser compensado pela rentabilidade de carregar mais demandas/peso. Caso este maior veículo receba um custo mais baixo que os menores veículos, os maiores veículos se tornariam mais atrativos, mesmo que estes não operem com a sua capacidade não utilizada da maneira ideal, causando subutilização dos menores veículos. Isso pode levar a uma operação ineficiente e mais custosa, pois rotas que poderiam ser executadas por veículos menores são executadas por veículos maiores e, na prática, mais caros. Assim, a proporção utilizada ajuda a alocar veículos mais adequados as rotas. Contudo, é importante frisar que em situações reais podem ser utilizados outros fatores para a definição deste custo de carga, como eficiência de combustível, salários dos motoristas e manutenções dos veículos.

Ao longo das análises, o método proposto foi executado diversas vezes para diferentes valores de  $w$  base para todas as instâncias de teste a fim de analisar os impactos dos valores nos resultados finais da otimização. As execuções levaram em conta os mesmos dados de entrada das 56 instâncias de Jiang et al. (2014), utilizados na primeira parte dos experimentos, apenas com o acréscimo de um coeficiente de carga para cada tipo de veículo, junto ao custo variável por distância e custo fixo de utilização do veículo, pertencentes à instância original. Além disso, é importante frisar que as condições de experimentação, ou seja, formato do algoritmo e parâmetros utilizados permanecem os mesmos da executada para o HFVRPTW apresentado na Seção 4.4.

Para a execução dos experimentos, foi definida uma faixa de possíveis valores de  $w$  base, compreendendo o intervalo  $[0.0005, 0.05]$ , compostos por valores incrementados em 0.0005, totalizando 100 diferentes valores de  $w$ . Esta faixa de valores foi definidas também de modo experimental, realizando algumas execuções do algoritmo e analisando seus impactos. Com base nisso, foram realizadas 100 execuções do algoritmo ALNS (uma para cada  $w$ ) para cada uma das 56 instâncias, o que levou a 5600 execuções do método. É importante mencionar que os custos fixos e variáveis por distância se mantiveram iguais aos definidos nas instâncias originais.

Para analisar os impactos do uso de  $w$ , além das execuções do algoritmo utilizando o coeficiente  $w$  na otimização, foram utilizados os melhores resultados de cada instância de teste na primeira etapa de experimentos para o HFVRPTW (sem custo de carga), expostos na Tabela 7. Uma vez que estes resultados foram obtidos sem a utilização do custo de carga ( $w = 0$ ), as soluções correspondentes aos resultados da Tabela 7 foram reprocessadas utilizando o valor de  $w$  na função objetivo segundo as definições acima descritas, para igualar as condições de comparação com as soluções para o HFVRPTW-LC. Assim, para uma mesma instância e um mesmo valor de  $w$ , tem-se uma solução otimizada sem a utilização do custo de carga  $w$  (HFVRPTW) e uma solução que utiliza este custo na otimização (HFVRPTW-LC), contudo ambas as soluções finais estão calculadas utilizando o custo de carga. Assim, a comparação entre HFVRPTW e HFVRPTW-LC indica os impactos de utilizar o custo de carga na otimização do problema. Portanto, dado que não há na literatura um trabalho diretamente correlacionado e que permita uma comparação direta, optou-se pela comparação com os resultados já obtidos sem a utilização do custo de carga. Esta prática de comparação já foi utilizada em trabalhos com de Tang, Zhang e Pan (2010), Zhang, Tang e Fung (2011) e Kuo e Wang (2012).

Para representar a diferença relativa ao custo total entre ambas as soluções, é proposta a utilização do RPD como indicador. Desse modo, a diferença percentual entre as duas soluções é dada por  $RPD^{LC} = (F^{HFVRPTW-LC} - F^{HFVRPTW})/F^{HFVRPTW}$ .

Caso o  $RPD^{LC}$  seja negativo, isso significa que a utilização do custo de carga no método de otimização causou uma redução de custo total. Caso contrário, indica que a utilização deste custo gerou aumento no custo total.

Com o intuito de analisar os impactos de crescimento do valor de  $w$  para cada cenário de teste, os indicadores de  $RPD^{LC}$  e ocupação dos veículos, a qual é definida pela soma total de demandas atendidas dividida pelo total de capacidade utilizada na operação. Os gráficos das Figuras 23 ilustram o comportamento destes indicadores para algumas instâncias ao passo que aumenta-se o valor de  $w$ . Durante as análises, observou-se que ao aumentar o valor de  $w$ , o  $RPD^{LC}$  tende a diminuir. Porém, a ocupação dos veículos também sofre impacto, de modo que em alguns cenários a ocupação tende a diminuir drasticamente, como nos nas instâncias HC201 e HR201 representados pelas Figuras 23.

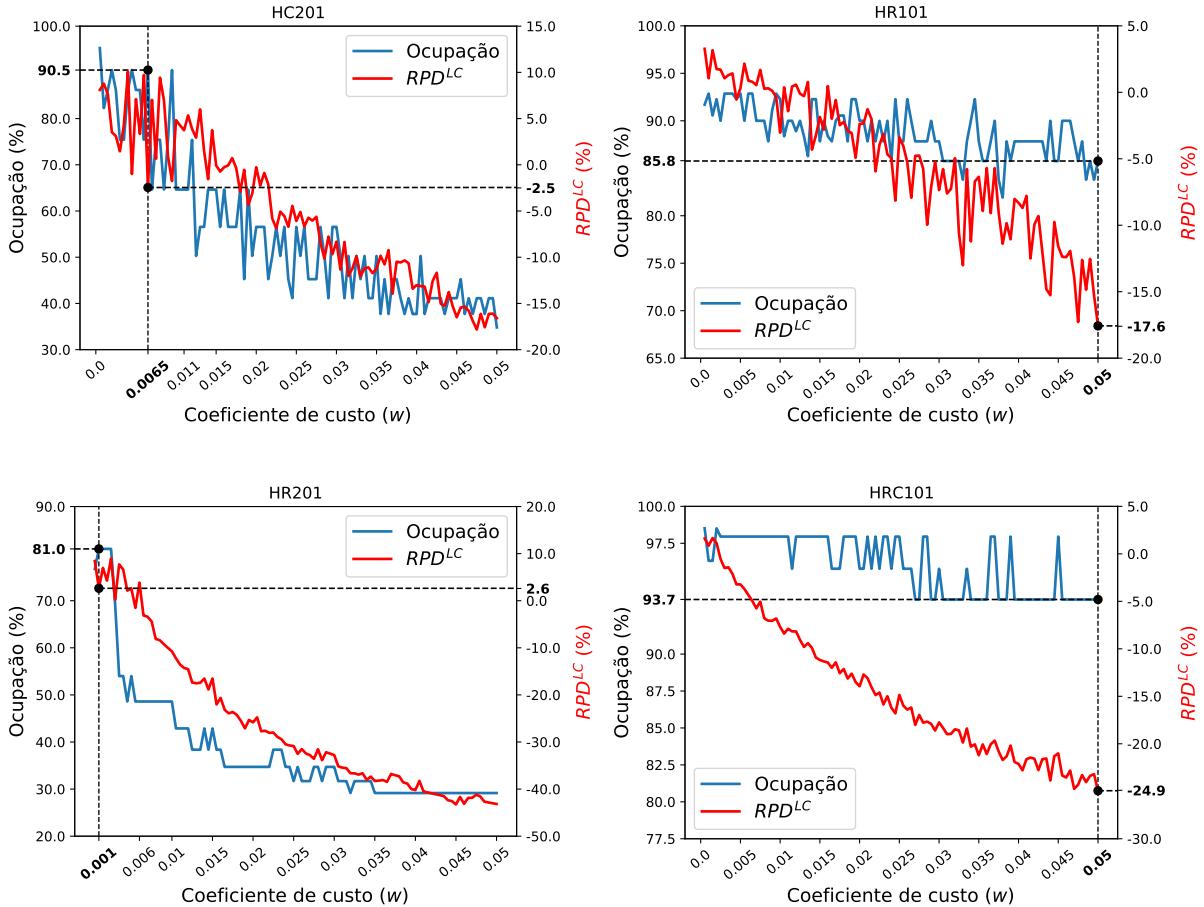


Figura 23 – Exemplos de análise e seleção dos valores de  $w$

Esta queda de ocupação pode se dar pela alocação de veículos extras de forma deliberada pelo método, o que causa uma diluição das demandas entre uma maior quantidade de veículos. Isso mostra que o custo  $w$  gera um impacto na proporção entre a parte referente ao custo fixo e ao custo variável da função objetivo definida na Seção 3.1. Uma vez que a porção variável é consideravelmente elevada em com-

paração com o custo fixo, devido a um custo  $w$  elevado, o método pode considerar ser vantajoso alocar um veículo a mais, sem que a penalidade (custo fixo) desta alocação seja relevante para o custo total.

Os gráficos das Figuras 23 também ilustram a análise e aplicação do critério de seleção do  $w$ . Dado os comportamentos ilustrados, observa-se que aumentar o custo  $w$  pode trazer ganhos consideráveis em relação a redução de custos da operação. Em contrapartida, pode levar a uma queda considerável de ocupação dos veículos. Em cenários reais de transporte é comum que a ocupação relativamente alta dos veículos seja tratada como uma das prioridades, já que rotas com poucas demandas transportadas seria ineficiente. Desse modo, para avaliar qual o valor de  $w$  mais adequado para cada instância, foi escolhido o  $w$  que resultasse em uma ocupação mínima de 80%, com base em cenários reais de transporte, e que obtivesse o mínimo  $RPD^{LC}$  possível. O método de seleção de  $w$  está exemplificado nos gráficos da Figura 23 de forma destacada através das linhas tracejadas e dos pontos de intersecção das mesmas. Esta metodologia foi aplicado para cada instância individualmente.

Diante dos resultados desta primeira análise, em que foi definido um  $w$  base para cada instância, um refinamento na busca por valores de  $w$  mais adequados foi realizada. Nesta etapa foram criadas variações de  $\pm 0.01$  sobre cada valor de  $w$  definido na etapa anterior, resultando em 3 variações:  $(w - 0.01, w, w + 0.01)$ , com valor mínimo de 0.0005. Além disso, para remover vieses de otimização, dado que o ALNS é um método estocástico, foram realizadas 30 execuções para cada variação de  $w$ , o que resultou em 5040 execuções ( $3 \times 30 \times 56$ ).

Diante dos resultados desta segunda etapa de exploração de valores de  $w$ , foram utilizados critérios mais rigorosos, sobretudo acerca da ocupação dos veículos, para definir o  $w$  mais adequado para cada instância. De modo a se aproximar dos índices de ocupação originais atingidos na primeira etapa dos experimentos para o HFVRPTW (sem custo de carga), em que foi atingido 95,6% de ocupação média geral considerando todas as instâncias. Portanto, os novos experimentos foram realizados considerando um mínimo de 90% de ocupação. Além disso, para priorizar resultados com maiores taxas de ocupação, o valor de  $w$  escolhido foi o que obteve maior número de soluções acima de 90% de ocupação dentre as 30 execuções para cada  $w$ . Caso houvesse empate dentro deste critério, o  $w$  cujo valor médio de  $RPD^{LC}$  das 30 execuções fosse o menor, seria o escolhido. Este critério foi aplicado para cada instância de teste, e os resultados são apresentados na Tabela 9.

A Tabela 9 apresenta a definição final dos valores de  $w$  para cada tipo de veículo de cada instância de teste. Assim, trabalhos futuros que utilizem as mesmas características, poderão usufruir desta definição para avaliar seus respectivos métodos, de forma a facilitar comparações e o desenvolvimento desta área de pesquisa.

Tabela 9 – Valores de  $w$  para cada tipo de veículo para cada instância

Instância	Valor $w$ por tipo de veículo			Instância	Valor $w$ por tipo de veículo		
	A	B	C		A	B	C
HC101	0,038	0,076	-	HR112	0,0595	0,0952	0,1428
HC102	0,039	0,078	-	HR201	0,0005	0,0007	-
HC103	0,058	0,116	-	HR202	0,0005	0,0007	-
HC104	0,04	0,08	-	HR203	0,0005	0,0007	-
HC105	0,0355	0,071	-	HR204	0,0005	0,0007	-
HC106	0,035	0,07	-	HR205	0,0005	0,0007	-
HC107	0,038	0,076	-	HR206	0,0015	0,002	-
HC108	0,0595	0,119	-	HR207	0,0005	0,0007	-
HC109	0,06	0,12	-	HR208	0,0005	0,0007	-
HC201	0,0065	0,0081	-	HR209	0,0025	0,0033	-
HC202	0,0005	0,0006	-	HR210	0,0005	0,0007	-
HC203	0,0025	0,0031	-	HR211	0,0005	0,0007	-
HC204	0,0015	0,0019	-	HRC101	0,06	0,12	0,225
HC205	0,0025	0,0031	-	HRC102	0,06	0,12	0,225
HC206	0,0005	0,0006	-	HRC103	0,058	0,116	0,2175
HC207	0,006	0,0075	-	HRC104	0,059	0,118	0,2213
HC208	0,0005	0,0006	-	HRC105	0,05	0,1	0,1875
HR101	0,04	0,064	0,096	HRC106	0,059	0,118	0,2213
HR102	0,0325	0,052	0,078	HRC107	0,049	0,098	0,1838
HR103	0,038	0,0608	0,0912	HRC108	0,06	0,12	0,225
HR104	0,0495	0,0792	0,1188	HRC201	0,0365	0,073	-
HR105	0,0585	0,0936	0,1404	HRC202	0,06	0,12	-
HR106	0,059	0,0944	0,1416	HRC203	0,047	0,094	-
HR107	0,06	0,096	0,144	HRC204	0,0565	0,113	-
HR108	0,057	0,0912	0,1368	HRC205	0,0495	0,099	-
HR109	0,0555	0,0888	0,1332	HRC206	0,049	0,098	-
HR110	0,055	0,088	0,132	HRC207	0,0555	0,111	-
HR111	0,059	0,0944	0,1416	HRC208	0,0575	0,115	-

Fonte: Próprio Autor

Vale destacar que ao longo das análises foram realizados experimentos com valores de  $w$  base global para todas as instâncias e também para cada categoria, considerando um valor médio de  $w$ . Porém, dadas as características particulares de cada instância de teste, embora hajam semelhanças entre instâncias da mesma categoria, identificou-se que utilizar valores médios, tanto globais quanto por categoria, comprometeu os resultados para algumas instâncias específicas. Por este motivo, optou-se pela atribuição de um valor  $w$  base para cada instância específica.

Ambas as etapas de experimentação para a definição dos valores de  $w$  base foram realizadas utilizando paralelismo entre as execuções, em que cada unidade de processamento realiza mais de uma execução do algoritmo, de forma a acelerar os

experimentos.

De modo geral, a primeira análise experimental explorou variações de  $w$  e utilizou uma tolerância de ocupação mínima flexível (80%), priorizando a diminuição do  $RPD^{LC}$  dentro desta tolerância. Já na segunda etapa realizou um refinamento mais rigoroso, em que foi utilizada uma tolerância de 90% de ocupação mínima e ainda priorizou a ocupação frente ao  $RPD^{LC}$ , de modo a se aproximar ao máximo da ocupação média global dos experimentos para o HFVRPTW.

#### 4.5.2 Resultados

A partir dos experimentos realizados para a definição dos valores de  $w$ , apresentados na Seção 4.5.1, os melhores resultados das 30 execuções para cada valor de  $w$  para cada instância foram compilados e usados como solução para o HFVRPTW-LC. Esses resultados são apresentados em detalhes na Tabela 10.

Para a definição da melhor solução dentre as 30 execuções para cada instância é considerada a solução com maior nível de ocupação dos veículos com a melhor solução. Caso haja empate neste critério, a solução com menor  $RPD^{LC}$  é escolhida. A Tabela 10 contempla todas as instâncias de teste e, para cada uma, são apresentados os resultados do problema HFVRPTW para a função objetivo contendo custo de carga e os resultados para o HFVRPTW-LC. As colunas CF (Custo Fixo), CV (Custo Variável), CT (Custo Total), DT (Distância Total), NV (Número de Veículos) e Ocupação (Ocup.) são apresentadas para ambos os resultados. A coluna  $RPD^{LC}$  é utilizada para representar os ganhos em termos de custo por utilizar o custo de carga na otimização e a coluna Avg. T(s) se refere a média de tempo de execução referente as 30 execuções do algoritmo para o HFVRPTW-LC. É importante frisar que de forma análoga a primeira etapa de experimentos, para todos os resultados obtidos nestes experimentos, todos os clientes foram atendidos e por este motivo nenhuma análise específica será realizada acerca de possíveis não atendimentos.

Dados os resultados da Tabela 10, nota-se que em 48 de 55 instâncias o uso do custo de carga gerou  $RPD^{LC}$  negativo, o que indica uma melhora nos custos da função objetivo. O nível de ocupação para cada instância se manteve acima de 90%, segundo a priorização dada a ocupação. O nível de ocupação médio para o HFVRPTW-LC alcançou um valor de 94,3%, o que indica uma redução de apenas 1,4% em relação aos 95,7% para o HFVRPTW. O  $RPD^{LC}$  médio é de -14,04%, o que indica considerável redução nos custos. Assim, observa-se que é possível ter redução de custos ao utilizar o custo de carga, sem reduzir de forma considerável as ocupações médias dos veículos. A média geral de tempo de execução é de 18,23 segundos, 2,55 segundos a mais que nos experimentos para o HFVRPTW.

Tabela 10 – Tabela de Resultados para HFVRPTW-LC

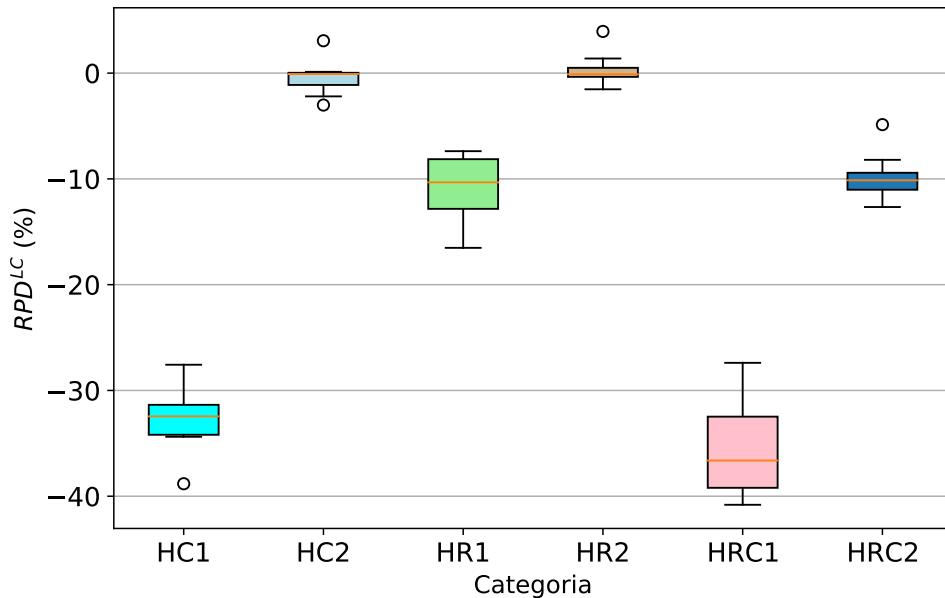
Instância	ALNS-HFVRPTW						ALNS-HFVRPTW-LC								RPD <sup>LC</sup>	Avg. T(s)
	CF	CV	CT	DT	NV	Ocup.	CF	CV	CT	DT	NV	Ocup.				
HC101	670	5995,5	6665,5	1095,78	14/25	95,26 %	600	3902,78	4502,78	1470,38	20/25	90,5 %	<b>-32,45 %</b>	12,23		
HC102	650	5519,3	6169,3	1104,11	15/25	95,26 %	570	3898,65	4468,65	1452,69	19/25	95,26 %	<b>-27,57 %</b>	14,23		
HC103	670	8515,87	9185,87	1032,83	14/25	95,26 %	600	5019,49	5619,49	1457,89	20/25	90,5 %	<b>-38,82 %</b>	15,93		
HC104	670	6068,39	6738,39	1007,03	14/25	95,26 %	570	3871,99	4441,99	1410,92	19/25	95,26 %	<b>-34,08 %</b>	16,9		
HC105	670	5557,09	6227,09	1055,1	14/25	95,26 %	600	3669,49	4269,49	1447,23	20/25	90,5 %	<b>-31,44 %</b>	14,07		
HC106	670	5522,85	6192,85	1070,17	14/25	95,26 %	570	3680,62	4250,62	1442,97	19/25	95,26 %	<b>-31,36 %</b>	14,1		
HC107	670	5662,05	6332,05	1059,14	14/25	95,26 %	570	3822,06	4392,06	1410,63	19/25	95,26 %	<b>-30,64 %</b>	14,73		
HC108	670	7983,68	8653,68	1032,55	14/25	95,26 %	600	5095,01	5695,01	1461,08	20/25	90,5 %	<b>-34,19 %</b>	15,1		
HC109	670	7962,54	8632,54	1015,85	14/25	95,26 %	600	5064,27	5664,27	1449,43	20/25	90,5 %	<b>-34,38 %</b>	16,0		
HC201	500	1580,12	2080,12	760,24	5/25	90,5 %	500	1517,22	2017,22	732,46	5/25	90,5 %	<b>-3,02 %</b>	16,77		
HC202	500	759,97	1259,97	701,37	5/25	90,5 %	500	758,3	1258,3	700,57	5/25	90,5 %	<b>-0,13 %</b>	22,9		
HC203	500	998,96	1498,96	695,03	5/25	90,5 %	500	1044,84	1544,84	727,57	5/25	90,5 %	<b>3,06 %</b>	27,7		
HC204	500	907,41	1407,41	713,2	5/25	90,5 %	500	896,74	1396,74	716,99	5/25	90,5 %	<b>-0,76 %</b>	33,7		
HC205	500	1082,0	1582,0	765,29	5/25	90,5 %	500	1047,11	1547,11	750,71	5/25	90,5 %	<b>-2,2 %</b>	18,37		
HC206	500	772,59	1272,59	710,31	5/25	90,5 %	500	774,15	1274,15	712,0	5/25	90,5 %	<b>0,12 %</b>	20,73		
HC207	500	1462,46	1962,46	718,38	5/25	90,5 %	500	1461,6	1961,6	728,92	5/25	90,5 %	<b>-0,04 %</b>	20,83		
HC208	500	745,28	1245,28	686,67	5/25	90,5 %	500	745,28	1245,28	686,67	5/25	90,5 %	<b>0,0 %</b>	21,73		
HR101	2920	6748,8	9668,8	1730,34	20/35	92,87 %	2840	5417,02	8257,02	1902,59	23/35	90,0 %	<b>-14,6 %</b>	13,0		
HR102	2830	5826,75	8656,75	1646,42	19/35	97,2 %	2920	4678,58	7598,58	1790,44	22/35	90,0 %	<b>-12,22 %</b>	14,03		
HR103	2810	6422,99	9232,99	1441,81	18/35	98,51 %	2860	5135,1	7995,1	1611,94	20/35	94,68 %	<b>-13,41 %</b>	15,07		
HR104	2680	6517,92	9197,92	1396,42	19/35	99,18 %	2700	4978,7	7678,7	1626,19	22/35	94,68 %	<b>-16,52 %</b>	16,07		
HR105	2540	6628,01	9168,01	1674,58	22/35	97,85 %	2620	5708,56	8328,56	1737,69	23/35	94,68 %	<b>-9,16 %</b>	13,83		
HR106	2540	6184,66	8724,66	1616,21	22/35	97,85 %	2620	5460,54	8080,54	1705,32	23/35	94,68 %	<b>-7,38 %</b>	15,03		
HR107	2520	6514,76	9034,76	1531,02	21/35	99,18 %	2620	5375,54	7995,54	1667,92	23/35	94,68 %	<b>-11,5 %</b>	15,83		
HR108	2540	5800,19	8340,19	1490,07	22/35	97,85 %	2620	5082,37	7702,37	1613,25	23/35	94,68 %	<b>-7,65 %</b>	16,17		
HR109	2540	5906,93	8446,93	1542,67	22/35	97,85 %	2620	5160,4	7780,4	1646,24	23/35	94,68 %	<b>-7,89 %</b>	14,7		
HR110	2540	5782,59	8322,59	1496,33	22/35	97,85 %	2620	5018,48	7638,48	1641,37	23/35	94,68 %	<b>-8,22 %</b>	15,7		
HR111	2540	6069,15	8609,15	1542,93	22/35	97,85 %	2620	5226,05	7846,05	1672,52	23/35	94,68 %	<b>-8,86 %</b>	15,83		
HR112	2520	6385,68	8805,68	1445,72	21/35	99,18 %	2540	5240,17	7780,17	1621,23	22/35	97,85 %	<b>-12,64 %</b>	16,17		
HR201	270	1392,21	1662,21	1311,56	6/15	81,0 %	250	1477,72	1727,72	1316,82	5/15	91,13 %	<b>3,94 %</b>	18,0		
HR202	225	1252,81	1477,81	1174,53	5/15	97,2 %	225	1230,21	1455,21	1153,2	5/15	97,2 %	<b>-1,53 %</b>	21,33		
HR203	225	1096,95	1321,95	1022,22	5/15	97,2 %	225	1089,64	1314,64	1014,56	5/15	97,2 %	<b>-0,55 %</b>	26,0		
HR204	225	885,71	1110,71	828,7	5/15	97,2 %	225	901,01	1126,01	841,74	5/15	97,2 %	<b>1,38 %</b>	29,7		
HR205	225	1080,6	1305,6	1010,46	5/15	97,2 %	225	1078,48	1303,48	1008,26	5/15	97,2 %	<b>-0,16 %</b>	21,17		
HR206	225	1115,37	1340,37	923,12	5/15	97,2 %	225	1117,64	1342,64	925,48	5/15	97,2 %	<b>0,17 %</b>	24,0		
HR207	225	904,68	1129,68	842,69	5/15	97,2 %	225	910,82	1135,82	849,13	5/15	97,2 %	<b>0,54 %</b>	27,03		
HR208	225	814,42	1039,42	759,48	5/15	97,2 %	225	819,22	1044,22	766,34	5/15	97,2 %	<b>0,46 %</b>	29,87		
HR209	225	1172,18	1397,18	873,04	5/15	97,2 %	225	1168,26	1393,26	878,67	5/15	97,2 %	<b>-0,28 %</b>	22,77		
HR210	225	1096,76	1321,76	1025,35	5/15	97,2 %	225	1091,14	1316,14	1021,93	5/15	97,2 %	<b>-0,43 %</b>	23,17		
HR211	225	823,52	1048,52	772,2	5/15	97,2 %	225	822,34	1047,34	770,58	5/15	97,2 %	<b>-0,11 %</b>	26,53		
HRC101	3270	14676,6	17946,6	1963,44	22/40	97,95 %	3300	9731,84	13031,84	2430,44	28/40	93,7 %	<b>-27,39 %</b>	13,47		
HRC102	3360	17673,55	21033,55	1728,95	19/40	97,4 %	3300	9586,53	12886,53	2392,08	28/40	93,7 %	<b>-38,73 %</b>	15,0		
HRC103	3360	16849,96	20209,96	1619,61	18/40	97,95 %	3300	9214,98	12514,98	2375,95	28/40	93,7 %	<b>-38,08 %</b>	15,77		
HRC104	3390	17527,88	20917,88	1385,52	16/40	98,51 %	3150	9230,74	12380,74	2359,38	27/40	97,95 %	<b>-40,81 %</b>	16,33		
HRC105	3300	14816,26	18116,26	1846,94	21/40	97,4 %	3390	8641,15	12031,15	2402,62	28/40	91,7 %	<b>-33,59 %</b>	14,43		
HRC106	3240	14646,53	17886,53	1829,99	22/40	98,51 %	3150	9523,89	12673,89	2319,15	27/40	97,95 %	<b>-29,14 %</b>	14,6		
HRC107	3330	14216,89	17546,89	1590,55	20/40	97,4 %	3150	8224,78	11374,78	2312,83	27/40	97,95 %	<b>-35,17 %</b>	15,33		
HRC108	3360	17661,83	21021,83	1441,29	18/40	97,95 %	3150	9330,23	12480,23	2290,16	27/40	97,95 %	<b>-40,63 %</b>	16,1		
HRC201	2900	6782,95	9682,95	1484,71	14/15	95,78 %	2900	6311,35	9211,35	1580,25	14/15	95,78 %	<b>-4,87 %</b>	15,07		
HRC202	2900	9835,07	12735,07	1352,78	14/15	95,78 %	2900	8524,76	11424,76	1451,99	14/15	95,78 %	<b>-10,29 %</b>	16,13		
HRC203	2900	8000,41	10900,41	1294,83	14/15	95,78 %	2900	6620,21	9520,21	1401,76	14/15	95,78 %	<b>-12,66 %</b>	17,03		
HRC204	2900	8508,53	11408,53	1252,02	14/15	95,78 %	2900	7184,3	10084,3	1354,16	14/15	95,78 %	<b>-11,61 %</b>	17,97		
HRC205	2900	8382,94	11282,94	1413,9	14/15	95,78 %	2900	7457,52	10357,52	1490,66	14/15	95,78 %	<b>-8,2 %</b>	16,03		
HRC206	2900	8244,9	11144,9	1371,49	14/15	95,78 %	2900	7148,82	10048,82	1478,41	14/15	95,78 %	<b>-9,83 %</b>	16,0		
HRC207	2900	8770,33	11670,33	1321,79	14/15	95,78 %	2900	7506,41	10406,41	1447,56	14/15	95,78 %	<b>-10,83 %</b>	16,93		
HRC208	2900	8436,81	11336,81	1247,71	14/15	95,78 %	2900	7305,72	10205,72	1409,48	14/15	95,78 %	<b>-9,98 %</b>	17,73		
Melhores		7 / 55		95,7 %				48 / 55				94,3 %	<b>-14,04 %</b>	18,23		

Fonte: Próprio Autor

Para verificar a significância das melhorias obtidas, foi aplicado o teste de Wilcoxon sobre os custos totais das soluções apresentadas na Tabela 10. Foi obtido um  $p$ -valor =  $5,64 \times 10^{-9}$  para um nível de confiança de 99% ( $p < 0,01$ ). Portanto, como o  $p$ -valor obtido é menor que 0,05, é possível refutar a hipótese nula e portanto inferir que ambas as soluções são significativamente diferentes. Assim, pode-se constatar que para nos cenários de teste apresentados o uso do custo de carga representou significativa melhoria em termos de custos totais para o problema proposto.

Para sumarizar os resultados, a Figura 24 apresenta o diagrama de caixa que

contém o  $RPD^{LC}$  por categoria de instâncias, e a Tabela 11 apresenta as médias dos principais indicadores de desempenho por categoria de instâncias, comparando os resultados para o HFVRPTW e HFVRPTW-LC, segundo a Tabela 10. Os indicadores utilizados são: veículos utilizados, distância, custo fixo, custo variável e ocupação dos veículos.



Fonte: Próprio Autor  
Figura 24 – Diagrama de Caixa  $RPD^{LC}$  (%) para HFVRPTW-LC

Tabela 11 – Tabela de média dos indicadores de cada categoria de instância

Indicadores	HC1	HC2	HR1	HR2	HRC1	HRC2
Veículos disponíveis	25	25	35	15	40	15
Veículos Utilizados	HFVRPTW	14,11	5,00	20,83	5,09	19,50
	HFVRPTW-LC	19,56	5,00	22,50	5,00	27,50
Distância	HFVRPTW	1052,51	718,81	1546,21	958,49	1675,79
	HFVRPTW-LC	1444,80	719,49	1686,39	958,79	2360,33
Custo Fixo	HFVRPTW	667,78	500,00	2626,67	229,09	3326,25
	HFVRPTW-LC	586,67	500,00	2683,33	227,27	3236,25
Custo Variável	HFVRPTW	6531,92	1038,60	6232,37	1057,75	16008,69
	HFVRPTW-LC	4224,93	1030,66	5206,79	1064,23	9185,52
Ocupação	HFVRPTW	95,26	90,50	97,77	95,73	97,88
	HFVRPTW-LC	92,62	90,50	94,16	96,65	95,58
$RPD^{LC}$ (%)	-32,77 %	-0,37 %	-10,84 %	1,14	-35,44 %	-9,78 %

Fonte: Próprio Autor

Ao observar o diagrama de caixa da Figura 24 e as Tabelas 10 e 11, nota-se que as categorias com resultados menos relevantes são HC2 e HR2. As demais categorias obtiveram impacto em seus custos de modo considerável. Em geral, para os cenários com melhor desempenho (HC1, HR1, HRC1 e HRC2) os indicadores se alteram com maior amplitude, segundo a Tabela 11, se comparados com os cenários HC2 e HR2. Nota-se que para estes cenários a quantidade veículos utilizados e/ou a distância total aumentam na comparação entre HFVRPTW e HFVRPTW-LC. Isso indica

que o algoritmo alocou mais veículos para a operação, sem comprometer a ocupação e remanejando a topologia/configuração das rotas de modo a encontrar soluções mais eficientes ao considerar o custo de carga como fator de impacto no custo total. Para estes cenários, a ocupação não foi comprometida devido à configuração da frota disponível, a qual contém veículos de pequeno e médio porte, se comparados com as demais categorias, o que significa que ao adicionar um veículo a mais, a ocupação não é impactada de forma drástica e há mais opções de modificações nas rotas para buscar uma redução de custo.

A categoria HRC2 é a que mais obteve destaques e um claro impacto positivo do uso do custo de carga. Ao observar a Tabela 11, nota-se que os indicadores de veículos utilizados, custo fixo e ocupação permanecem os mesmos na comparação entre HFVRPTW e HFVRPTW-LC, com apenas a distância sofrendo um aumento e o custo variável uma diminuição considerável. Logo, nota-se que a mesma frota foi utilizada, porém a mudança da configuração das rotas, embora tenha havido um aumento na distância, a minimização do custo de carga causou uma diminuição intensa no custo variável e, por consequência, no  $RPD^{LC}$ . Isso demonstra o impacto positivo evidente em utilizar o custo de carga no método de otimização.

Em contrapartida, nos cenários de pior resultados (HC2 e HR2) a frota de veículos possui apenas veículos de grande porte. Isso faz com que a adição de novos veículos leve a uma queda acelerada na ocupação deles, como pode ser notado nos gráficos da Figura 23. Portanto, as limitações destes cenários em decorrência das características da frota de veículos podem ter influenciado no desempenho final do algoritmo para estes casos.

Ao longo da realização dos experimentos sobre o HFVRPTW-LC foram analisados os gráficos de convergência e comportamento dos pesos dos operadores da solução proposta. Entretanto, o comportamento apresentado pelo algoritmo foi muito semelhante ao observado nos experimentos para o HFVRPTW, na Seção 4.4.3. Por este motivo, esta seção não apresenta a análise do comportamento do algoritmo utilizando na experimentação do HFVRPTW-LC.

Dados os resultados obtidos ao longo de toda a análise do HFVRPTW-LC, conclui-se que a utilização do custo de carga como parte da função objetivo causa um impacto significativo na redução do custo, além de aproximar ainda mais o VRP às operações reais de transporte.

#### 4.6 CONSIDERAÇÕES SOBRE OS EXPERIMENTOS

Este capítulo apresentou os experimentos realizados para a análise e avaliação do método proposto. O protocolo de experimentos foi apresentado de maneira

a separar a análise em duas etapas: uma que trata do problema HFVRPTW sem considerar o custo de carga, com o objetivo de avaliar o método perante a literatura existente; e outra para o HFVRPTW-LC, a qual analisa os impactos de considerar o custo de carga na otimização do HFVRPTW. Além disso, ao longo dos experimentos foram feitas análises quantitativas em relação a ambos os problemas tratados, bem como uma análise qualitativa para observar o comportamento do algoritmo perante diferentes cenários de teste.

Em seguida foi descrita a base de dados utilizada, introduzida por Jiang et al. (2014) e baseada em Solomon (1987). Este *benchmark* foi escolhido pois, dentre os *benchmarks* disponíveis na literatura, este é o que mais se encaixa com as características do problema proposto no que diz respeito à janela de tempo, frota heterogênea limitada e os custos fixo e variável relacionados a frota (HFVRPTW). Após a descrição das bases de dados, foram apresentadas as definições dos parâmetros do algoritmo proposto, que contém os parâmetros da solução inicial e do algoritmo ALNS.

O objetivo desta análise é avaliar o desempenho do método proposto para o HFVRPTW frente a um método já existente na literatura, a fim de observar a capacidade do mesmo frente a um cenário com janelas de tempo e, sobretudo, frota heterogênea limitada, já que não se identificou na literatura um trabalho com estas características juntamente com a noção de custo de carga, sendo necessário reduzir as configurações do problema da proposta final para conseguir estar em nível de comparação com a literatura atual. Desse modo, o trabalho Molina, Salmeron e Eguia (2020), o qual utiliza um algoritmo híbrido ACS-VNTS, foi selecionado para esta comparação, pois se trata do trabalho com melhores resultados até a data do presente trabalho.

Os resultados obtidos nesta primeira análise apontaram uma redução média de 1,48% nos custos totais, em que 51 das 56 instâncias foram melhoradas e uma manteve custo igual. O teste de Wilcoxon indicou significância estatística, o que indica que em média o método proposto superou o desempenho alcançado por Molina, Salmeron e Eguia (2020). Em relação ao tempo de execução médio, o método proposto atingiu 15,68 segundos de média, o que é 33 vezes mais rápido do que o do método definido por Molina, Salmeron e Eguia (2020). As categorias com melhor desempenho foram HC2 e HR2, que representam instâncias de distribuição em *cluster* ou aleatória dos clientes e rotas mais longas. Os resultados menos satisfatórios foram os da categoria HRC2, na qual a distribuição combina *clusters* e aleatórias e rotas longas.

Diante as análises qualitativas, ainda dentro da primeira etapa de experimentos, foi observado que em geral o algoritmo ALNS conseguiu se adaptar aos diversos cenários de teste. Porém, em grande parte das categorias de teste o algoritmo apresentou uma convergência mais acelerada e estagnação precoce, devido à atuação intensa de operadores específicos como *Worst Removal* e *Regret-2 Insertion*, que fa-

zem com que a busca se localize em uma área mais restrita do espaço de soluções. Com isso, embora este comportamento não reflita necessariamente de forma negativa nos resultados qualitativos apresentados, pode ser um indicativo da necessidade de aumentar a quantidade e variação dos operadores de remoção e inserção, para possivelmente obter maior exploração e diversificação nos estágios iniciais da busca.

Além disso, destaca-se a utilização do operador de remoção *Route Removal* e o método de construção semi-paralelo (utilizado na construção da solução inicial) como complemento dos operadores de inserção. A utilização destes métodos contribuiu para a destruição e reconstrução de rotas inteiras, de modo a causar modificações nos tipos de veículos destas rotas, o que pode gerar soluções promissoras, sobretudo para cenários de frota heterogênea de veículos.

Após a primeira etapa, foi realizada a inclusão da noção de custo de carga nas análises, caracterizando o problema como um HFVRPTW-LC. Para isso, uma nova experimentação foi proposta a fim de definir os coeficientes de custo de carga  $w$  individuais para cada tipo de veículo para cada instância de teste, já que a literatura atual só apresenta definição para cenários de *benchmark* particulares e não públicos de transporte.

Nesta etapa, foi promovido um estudo exploratório amplo de modo a testar diferentes valores para o coeficiente  $w$  e analisar os impactos destes valores no desempenho das soluções encontradas. Foram utilizados os resultados da primeira etapa de experimentos (HFVRPTW - sem custo de carga) como comparativo para analisar os impactos de incluir este custo. Para a avaliação de desempenho, ao passo que os valores de  $w$  eram aumentados, foi utilizado o indicador  $RPD^{LC}$  que mostra a diferença percentual do custo entre os resultados do método para o HFVRPTW e HFVRPTW-LC, e também a ocupação dos veículos. Os estudos mostram que ao aumentar o valor de  $w$  existe uma tendência de diminuição do  $RPD^{LC}$ , o que indica redução de custo, mas com uma diminuição da ocupação dos veículos, devido a alocações de novos veículos na operação. Diante disso, foram estipulados limites mínimos de ocupação de modo empírico e baseados na ocupação obtida nos experimentos sobre o HFVRPTW, que foi em torno de 95%.

Ao final deste processo, foi definida uma tabela contendo os valores de  $w$  para cada tipo de veículo por instância de teste, de modo a contribuir com trabalhos futuros que foquem no estudo de problemas com frota heterogênea e custo de carga.

Diante dos resultados obtidos, verificou-se que a utilização do custo de carga no método de otimização gera uma redução significativa no custo da função objetivo, sem impactar drasticamente na ocupação dos veículos. Os resultados apresentaram uma redução média de 14,04% dos custos e uma ocupação média de 94,3%, em

que 48 de 56 instâncias obtiveram melhorias no custo e uma instância teve custo igual. Portanto, observa-se que o uso do custo de carga gerou um impacto positivo em média para os cenários de teste apresentados.

## 5 CONCLUSÃO

A área de logística de transporte vem cada vez mais buscando por iniciativas que visem oportunizar a otimização e a eficiência na distribuição devido à necessidade de reduções de custos operacionais e consequente impacto no restante das cadeias de produção. Na academia, esse problema se enquadra na área de estudos de VRP, que podem ser especializados nos mais diversos tipos e características de problemas de transporte.

Este trabalho tem o objetivo de propor uma solução para o problema de roteamento de veículos com característica de janela de tempo, frota heterogênea limitada e custo de carga (HFVRPTW-LC), por meio de um método meta-heurístico de busca larga adaptativa em vizinhança (ALNS). Para isso, um referencial teórico foi introduzido para definir os conceitos que permeiam este trabalho, como as definições das variações de problemas de roteamento de veículos envolvidos e os conceitos de heurísticas e meta-heurísticas. Foi realizado um levantamento bibliográfico acerca dos temas envolvidos segundo as características utilizadas no problema, o que totalizou em 27 trabalhos levantados que estão diretamente relacionados ao presente trabalho. Uma modelagem matemática do problema proposto e o método de solução ALNS adaptado para o problema foi definido.

A modelagem matemática do problema, dado como um objetivo específico deste trabalho, foi proposta de modo a unificar as restrições referentes à janela de tempo, frota heterogênea limitada e custos envolvidos, bem como custo variável por distância, custo variável por carga transportada (*loading cost*) e custo fixo de utilização dos veículos. Também, na modelagem matemática está presente a possibilidade de não atendimento de clientes dado o limite de recursos (frota de veículos limitada). Desse modo, há na função objetivo de custo um incentivo para que haja o atendimento do maior número de clientes possível. A partir disso, foi proposto um algoritmo baseado na meta-heurística ALNS como método de solução, combinado com métodos de busca local e solução inicial semi-paralela.

Dentre os principais desafios na solução proposta para o problema está a forma com que novas rotas são construídas, dada a necessidade de definir os tipos de veículos de cada rota, bem como a quantidade de veículos utilizados. Ainda, lidar com possíveis não atendimentos é um desafio extra para este cenário, onde os recursos são limitados como a frota de veículos. Diante disso, um dos principais objetivos específicos deste trabalho é utilizar um método de criação de rotas sem vieses causados por possíveis priorizações ou ordenações de veículos e demandas.

Para lidar com estes desafios, foi utilizado um método de construção semi-paralelo adaptado do que foi proposto em Paraskevopoulos et al. (2008), em que todos os tipos de veículos são considerados ao mesmo tempo na criação de rotas de uma solução. Desse modo, evita-se o uso de estratégias puramente gulosas, em que são realizadas alocações ordenadas de clientes em veículos, de modo que a maior demanda é encaixada no maior veículo. Além disso, para lidar com não atendimentos foi utilizado o conceito de *Holding List*, proposto em Lau, Sim e Teo (2003), que se trata de uma estrutura que armazena clientes não atendidos ao longo de todo o processo de otimização, de modo a tornar factíveis as soluções mesmo que hajam demandas não atendidas, e também permita uma navegação mais livre pelo espaço de busca.

Do ponto de vista da meta-heurística ALNS, além dos operadores de remoção e inserção mais conhecidos pela literatura como *Worst*, *Random*, *Shaw Removal* e *Greedy* e *Regret Insertion*, neste trabalho foram propostas duas contribuições: i) a criação de um método de remoção (*Route Removal*), em que 40% das rotas são removidas de forma completa; e ii) a utilização do método de construção semi-paralelo para complementar os operadores de inserção após a tentativa de inserir clientes da *Holding List* nas rotas existentes, de modo que sejam criadas novas rotas para atender clientes ainda não atendidos. Desse modo, estas duas modificações trouxeram a possibilidade de uma destruição e reconstrução mais profunda da solução, de modo a remover e reconstruir rotas inteiras com outros tipos de veículos. Esta estratégia se mostrou vantajosa na obtenção de melhores resultados para o cenário de frota heterogênea proposto.

No que diz respeito ao custo de carga, não houve alteração do método em si, apenas a função objetivo passou a considerar este custo adicional ao longo da otimização. A hipótese de que sua utilização causa um impacto na redução de custos foi considerada na fase de experimentos.

Os experimentos realizados foram divididos em duas etapas, sendo uma a aplicação de análise quantitativa e a outra uma análise qualitativa. A primeira etapa analisa o desempenho do algoritmo ALNS para o problema HFVRPTW, o qual não considera o custo de carga na otimização. Esta análise foi realizada para validar o método proposto por meio de sua comparação com um trabalho presente na literatura. Isso porque não se identificou um trabalho que contemplasse todas as características do VRP proposto. Dados os resultados desta etapa, observou-se uma melhora estatisticamente significativa na qualidade das soluções obtidas pelo método proposto em relação ao trabalho de Molina, Salmeron e Eguia (2020) (ACS-VNTS). A redução média dos custos é de 1,48%, em que 51 das 56 instâncias foram melhoradas. No que diz respeito ao tempo de execução, o algoritmo proposto atingiu um tempo médio 33 vezes menor que o tempo utilizado no trabalho de Molina, Salmeron e Eguia (2020).

Assim, além de encontrar melhores soluções para a grande maioria das instâncias de teste, o método proposto também é mais rápido.

Para a segunda parte dos experimentos, é adicionado o custo de carga na função objetivo do método proposto. O intuito desta análise é verificar se a utilização do custo de carga tem impactos no custo e em outros indicadores. Para isso, são comparados os resultados obtidos no primeiro experimento, para o HFVRPTW, e o resultado para o HFVRPTW-LC. Dadas as análises dos resultados, nota-se uma redução média de custo de 14,04% com significância estatística e 48 de 56 instâncias melhoradas, sem comprometer a ocupação média dos veículos, que é de 94,3%.

Sobre as análises qualitativas em torno da convergência e comportamento dos pesos dos operadores do método proposto, observou-se que em grande parte dos cenários, o algoritmo realizou uma busca mais intensificada do que diversificada. Isso pode ser observado pela atuação de operadores específicos com maior intensidade e relevância em detrimento de outros para 4 das 6 categorias de teste. O que demonstra uma busca mais localizada em regiões específicas do espaço de busca. Isso pode ser um indicativo da necessidade de uma maior quantidade e variação de operadores de remoção e inserção.

A partir das análises dos experimentos, é possível observar que o método proposto atingiu desempenho satisfatório em comparação com os melhores resultados da literatura até então, no que diz respeito a frota heterogênea limitada com janela de tempo (HFVRPTW). Ainda, o uso do custo de carga (*loading cost*) causou impacto positivo na redução de custos, sem comprometer a ocupação dos veículos, para os cenários de teste utilizados. Dessa forma, demonstrou-se que a utilização deste custo em métodos de otimização de VRP pode trazer ganhos consideráveis de eficiência nas operações logísticas.

## 5.1 TRABALHOS FUTUROS

Com base nos resultados obtidos e na identificação de possíveis melhorias ou potenciais incrementos ao método proposto, abaixo são apresentados os possíveis trabalhos futuros:

- **Aumentar a quantidade e variedade dos operadores do ALNS:** levando em consideração os experimentos deste trabalho, notou-se que utilizar uma maior quantidade e variedade de operadores pode causar maior equilíbrio entre os operadores ao longo da otimização e assim trazer maior diversificação na busca. Portanto, um possível trabalho futuro pode ser a implementação de novos operadores para o método proposto.

- **Exploração de outros métodos de busca local:** fazer experimentos com outros algoritmos de busca local como *2-Opt*, *3-Opt* e outras variações de *Swap*, pode levar a uma melhoria de desempenho. Portanto, novos operadores podem ser explorados como trabalho futuro.
- **Exploração de outros métodos de seleção de operadores:** O método de seleção por roleta pode não ser o método mais adequado para seleção dos operadores do ALNS, sendo importante investigar outros métodos.
- **Experimentos com outros valores de parâmetros:** em todo algoritmo baseado em meta-heurística há a necessidade de explorar variados valores de parâmetros para tentar melhorar o desempenho obtido. Portanto, um trabalho futuro pode ser a automação da definição de alguns parâmetros ou mesmo a realização de testes intensivos para definir melhores valores para os parâmetros existentes.
- **Experimentos com cenários de não atendimento:** no presente trabalho, os cenários de teste utilizados não são os mais adequados para teste com relação aos não atendimentos, pois os recursos disponíveis eram em gerar suficientes para atender todas as demandas. Assim, uma análise específica de cenários de não atendimento pode ser abordado como trabalho futuro.
- **Demandas bi-dimensionais e tri-dimensionais:** dada a relevância da forma e do volume da carga transportada, outros cenários de VRP a serem explorados são os que levam em consideração demandas com 2 ou 3 dimensões (VRP-2L e VRP-3L). Portanto, além da otimização da ordem de visitas baseado nos custos operacionais, a alocação das demandas nos veículos deve levar em consideração o formato das encomendas. Portanto, a otimização da alocação das cargas em 2 ou 3 dimensões também é um trabalho futuro relevante.

## REFERÊNCIAS

- AVDOSHIN, S.; BERESNEVA, E. Constructive heuristics for capacitated vehicle routing problem: a comparative study. **Proceedings of the Institute for System Programming of the RAS**, v. 31, p. 145–156, 09 2019.
- BANDARU, S.; DEB, K. Metaheuristic techniques: Theory and practice. In: \_\_\_\_\_. [S.l.: s.n.], 2016. p. 693–750. ISBN 978-1-4665-6430-5.
- BAYER, F. R. **Proposta de Algoritmo de Busca Larga Vizinhança para o Problema de Roteamento de Veículos com Suporte a Múltiplas Janelas de Tempo**. 115 p. Dissertação (Mestrado) — Universidade do Estado de Santa Catarina - UDESC, 2021.
- BENUWA, B. et al. A comprehensive review of particle swarm optimization. **International Journal of Engineering Research in Africa**, v. 23, p. 141–161, 04 2016.
- BRAEKERS, K.; RAMAEKERS, K.; Van Nieuwenhuyse, I. The vehicle routing problem: State of the art classification and review. **Computers and Industrial Engineering**, Elsevier Ltd, v. 99, p. 300–313, 2016. ISSN 03608352.
- BRANDÃO, J. A tabu search algorithm for the heterogeneous fixed fleet vehicle routing problem. **Computers & Operations Research**, v. 38, n. 1, p. 140–151, 2011. ISSN 0305-0548. Project Management and Scheduling. Disponível em: <<https://www.sciencedirect.com/science/article/pii/S0305054810000924>>.
- BRASY, O.; GENDREAU, M. Vehicle routing problem with time windows, Part I: Route construction and local search algorithms. **Transportation Science**, v. 39, p. 104–118, 2005.
- BURAK, E.; VURAL, A. V.; REISMAN, A. The vehicle routing problem: A taxonomic review. **Computers & Industrial Engineering**, p. 1472–1483, 2009.
- CARLIER, M. **Mobility-as-a-Service - statistics & facts**. 2022. Disponível em: <[https://www.statista.com/topics/4664/mobility-as-a-service/#dossierContents\\_outerWrapper](https://www.statista.com/topics/4664/mobility-as-a-service/#dossierContents_outerWrapper)>.
- CLARKE, G.; WRIGHT, J. W. Scheduling of Vehicles from a Central Depot to a Number of Delivery Points. **Operations Research**, p. 568–581, 1964.
- CORDEAU, J. F. et al. Chapter 6 Vehicle Routing. **Handbooks in Operations Research and Management Science**, v. 14, n. C, p. 367–428, 2007. ISSN 09270507.
- COSTA, J. M. Estudo do Problema de Roteamento Aberto de Veículos com Janelas de Tempo Utilizando Algoritmos Colônia de Formiga. v. 1, p. 36–40, 2012.
- DANTZIG, G. B.; RAMSER, J. H. The truck dispatching problem. **Management Science**, Institute for Operations Research and the Management Sciences (INFORMS), v. 6, n. 1, p. 80–91, out. 1959. Disponível em: <<https://doi.org/10.1287/mnsc.6.1.80>>.

DERBEL, H.; JARBOUI, B.; BHIRI, R. A skewed general variable neighborhood search algorithm with fixed threshold for the heterogeneous fleet vehicle routing problem. **Annals of Operations Research**, v. 272, 01 2019.

DORIGO, M.; BIRATTARI, M.; STUTZLE, T. Ant colony optimization: Artificial ants as a computational intelligence technique. **IEEE Computational Intelligence Magazine**, v. 1, p. 28–39, 2006.

DORIGO, M.; STUTZLE, T. The ant colony optimization metaheuristic: Algorithms, applications, and advances. **Kluwer Academic Publishers**, p. 251–285, 2003.

DORIGO, M.; STUTZLE, T. **Ant Colony Optimization**. MIT Press, 2004.

FACHINI, R.; ARMENTANO, V. Logic-based benders decomposition for the heterogeneous fixed fleet vehicle routing problem with time windows. **Computers & Industrial Engineering**, v. 148, p. 106641, 07 2020.

FERDI, I.; LAYEB, A. A novel heuristic based simulated annealing for the capacitated location routing problem. In: . New York, NY, USA: Association for Computing Machinery, 2016. (MedPRAI-2016), p. 17–24. ISBN 9781450348768. Disponível em: <<https://doi.org/10.1145/3038884.3038888>>.

GENDREAU, M. et al. A tabu search heuristic for the heterogeneous Feet vehicle routing problem. **Computers & Operations Research**, p. 1153–1173, 1999.

GLOVER, F. Future paths for integer programming and links to artificial intelligence. **Computers & Operations Research**, v. 13, n. 5, p. 533–549, 1986. ISSN 0305-0548. Applications of Integer Programming. Disponível em: <<https://www.sciencedirect.com/science/article/pii/0305054886900481>>.

GOLDBARG, E.; GOLDBARG, M.; LUNA, H. **Otimização Combinatória e Metaheurísticas: Algoritmos e Aplicações**. First editon. [S.I.]: Campus, 2015.

GOLDEN, B. et al. The Fleet Size and Mix Vehicle Routing Problem. **Computers & Operations Research**, v. 11, p. 49–66, 1984.

GUNAWAN, A. et al. Adaptive large neighborhood search for vehicle routing problem with cross-docking. In: **2020 IEEE Congress on Evolutionary Computation (CEC)**. [S.I.: s.n.], 2020. p. 1–8.

HASLE, G.; KLOSTER, O. Industrial vehicle routing problems. **Geometric Modelling, Numerical Simulation and Optimization: Applied Mathematics at SINTEF**, 2007.

HOMBERGER, J.; GEHRING, H. A two-phase hybrid metaheuristic for the vehicle routing problem with time windows. **European Journal of Operational Research**, p. 220–238, 2005.

HOOS, H. H.; STUTZLE, T. **Stochastic Local Search: Foundations and Applications**. First editon. [S.I.]: Morgan Kaufmann E.U.A., 2004.

JIANG, J. et al. Vehicle routing problem with a heterogeneous fleet and time windows. **Expert Systems with Applications**, v. 41, p. 3748–3760, 06 2014.

KIRKPATRICK, S.; GELATT, C. D.; VECCHI, M. P. Optimization by simulated annealing. **Science**, American Association for the Advancement of Science (AAAS), v. 220, n. 4598, p. 671–680, maio 1983. Disponível em: <<https://doi.org/10.1126/science.220.4598.671>>.

KOÇ, Ç. et al. A hybrid evolutionary algorithm for heterogeneous fleet vehicle routing problems with time windows. **Comp. and Op. Research**, v. 64, p. 11–27, 2015.

KOÇ, Ç. et al. Thirty years of heterogeneous vehicle routing. **European Journal of Operational Research**, Elsevier, v. 249, n. 1, p. 1–21, 2016.

KUO, Y.; WANG, C.-C. A variable neighborhood search for the multi-depot vehicle routing problem with loading cost. **Expert Systems with Applications**, v. 39, n. 8, p. 6949–6954, 2012. ISSN 0957-4174. Disponível em: <<https://www.sciencedirect.com/science/article/pii/S0957417412000371>>.

LAPORTE, G.; ROPKE, S.; VIDAL, T. Chapter 4: Heuristics for the vehicle routing problem. In: \_\_\_\_\_. [S.l.: s.n.], 2014. p. 87–116. ISBN 978-1-61197-358-7.

LAPORTE, G.; SEMET, F. 5. classical heuristics for the capacitated vrp. In: \_\_\_\_\_. [S.l.: s.n.], 2002. p. 109–128. ISBN 978-0-89871-498-2.

LAU, H. C.; SIM, M.; TEO, K. M. Vehicle routing problem with time windows and a limited number of vehicles. **European Journal of Operational Research**, v. 148, n. 3, p. 559–569, 2003. ISSN 0377-2217. Disponível em: <<https://www.sciencedirect.com/science/article/pii/S0377221702003636>>.

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, p. 2734–2742, 2007. ISSN 0305-0548. Disponível em: <<https://www.sciencedirect.com/science/article/pii/S0305054805003382>>.

LI, X.; LEUNG, S. C.; TIAN, P. A multistart adaptive memory-based tabu search algorithm for the heterogeneous fixed fleet open vehicle routing problem. **Expert Systems with Applications**, v. 39, n. 1, p. 365–374, 2012. ISSN 0957-4174. Disponível em: <<https://www.sciencedirect.com/science/article/pii/S0957417411009870>>.

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, p. 1111–1127, November 2010. Disponível em: <<https://ideas.repec.org/a/eee/transe/v46y2010i6p1111-1127.html>>.

MATEI, O. et al. An improved immigration memetic algorithm for solving the heterogeneous fixed fleet vehicle routing problem. **Neurocomputing**, v. 150, 10 2014.

MCCREA, B. **TMS Update: Adoption accelerates**. 2021. Disponível em: <[https://www.logisticsmgmt.com/article/tms\\_update\\_adoption\\_accelerates/tms](https://www.logisticsmgmt.com/article/tms_update_adoption_accelerates/tms)>.

MEHDI, N.; ABDELMOUTALIB, M.; IMAD, H. A modified alns algorithm for vehicle routing problems with time windows. **Journal of Physics: Conference Series**, IOP Publishing, v. 1743, n. 1, p. 012029, jan 2021. Disponível em: <<https://dx.doi.org/10.1088/1742-6596/1743/1/012029>>.

MOLINA, J.; SALMERON, J.; EGUIA, I. An acs-based memetic algorithm for the heterogeneous vehicle routing problem with time windows. **Expert Systems with Applications**, v. 157, p. 113379, 04 2020.

MOLINA, J. C. et al. The heterogeneous vehicle routing problem with time windows and a limited number of resources. **Engineering Applications of Artificial Intelligence**, v. 94, p. 103745, 2020. ISSN 0952-1976. Disponível em: <<https://www.sciencedirect.com/science/article/pii/S095219762030155X>>.

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, p. 4316–4324, 2013. ISSN 0307-904X. Disponível em: <<https://www.sciencedirect.com/science/article/pii/S0307904X12005240>>.

PARASKEVOPOULOS, D. C. et al. A reactive variable neighborhood tabu search for the heterogeneous fleet vehicle routing problem with time windows. **Journal of Heuristics**, v. 14, p. 425–455, 2008.

PISINGER, D.; ROPKE, S. A general heuristic for vehicle routing problems. **Computers & Operations Research**, Elsevier BV, v. 34, n. 8, p. 2403–2435, ago. 2007. Disponível em: <<https://doi.org/10.1016/j.cor.2005.09.012>>.

PRINS, C. Efficient heuristics for the heterogeneous fleet multtrip vrp with application to a large-scale real case. **Journal of Mathematical Modelling and Algorithms**, v. 1, p. 135–150, 2002.

PRINS, C. A simple and effective evolutionary algorithm for the vehicle routing problem. **Computers & Operations Research**, v. 31, n. 12, p. 1985–2002, 2004. ISSN 0305-0548. Disponível em: <<https://www.sciencedirect.com/science/article/pii/S0305054803001588>>.

RESENDE, M. G.; RIBEIRO, C. C. Greedy randomized adaptive search procedures: Advances, hybridizations, and applications. In: \_\_\_\_\_. **Handbook of Metaheuristics**. Boston, MA: Springer US, 2010. p. 283–319. ISBN 978-1-4419-1665-5. Disponível em: <[https://doi.org/10.1007/978-1-4419-1665-5\\_10](https://doi.org/10.1007/978-1-4419-1665-5_10)>.

RØPKE, S. Heuristic and exact algorithms for vehicle routing problems. In: . [S.l.: s.n.], 2006.

ROPKE, S.; PISINGER, D. An adaptive large neighborhood search heuristic for the pickup and delivery problem with time windows. **Transportation Science**, v. 40, p. 455–472, 11 2006.

SACRAMENTO, D.; PISINGER, D.; ROPKE, S. An adaptive large neighborhood search metaheuristic for the vehicle routing problem with drones. **Transportation Research Part C: Emerging Technologies**, v. 102, p. 289–315, 2019. ISSN 0968-090X. Disponível em: <<https://www.sciencedirect.com/science/article/pii/S0968090X18303218>>.

SAVELSBERGH, M. An efficient implementation of local search algorithms for constrained routing problems. **European Journal of Operational Research**, v. 47, n. 1, p. 75–85, 1990. ISSN 0377-2217. Disponível em: <<https://www.sciencedirect.com/science/article/pii/037722179090091O>>.

SAVELSBERGH, M. W. P. The vehicle routing problem with time windows: Minimizing route duration. **ORSA Journal on Computing**, Institute for Operations Research and the Management Sciences (INFORMS), v. 4, n. 2, p. 146–154, maio 1992. Disponível em: <<https://doi.org/10.1287/ijoc.4.2.146>>.

SCHMITT, J. P. **Proposta de Algoritmo de Busca Larga Adaptativa em Vizinhança para o Problema de Roteamento de Veículos com Coleta e Entrega, Janela de Tempo e Requisições Dinâmicas**. 157 p. Dissertação (Mestrado) — Universidade do Estado de Santa Catarina - UDESC, 2020.

SHAW, P. Using constraint programming and local search methods to solve vehicle routing problems. In: **Principles and Practice of Constraint Programming — CP98**. Springer Berlin Heidelberg, 1998. p. 417–431. Disponível em: <[https://doi.org/10.1007/3-540-49481-2\\_30](https://doi.org/10.1007/3-540-49481-2_30)>.

SOLOMON, M. M. Algorithms for the vehicle routing and scheduling problems with time window constraints. **Operations Research**, INFORMS, v. 35, n. 2, p. 254–265, 1987. ISSN 0030364X, 15265463. Disponível em: <<http://www.jstor.org/stable/170697>>.

STATISTA. **Transportation & Logistics**. 2022. Disponível em: <<https://www.statista.com/markets/419/transportation-logistics/>>.

SUBRAMANIAN, A. et al. A hybrid algorithm for the heterogeneous fleet vehicle routing problem. **European Journal of Operational Research**, v. 221, n. 2, p. 285–295, 2012. ISSN 0377-2217. Disponível em: <<https://www.sciencedirect.com/science/article/pii/S0377221712002093>>.

TAILLARD, E. D. A heuristic column generation method for the heterogeneous fleet Vehicle Routing Problem. **RAIRO Operations Research**, v. 33, p. 1–14, 1999.

TANG, J. et al. Beam search combined with max-min ant systems and benchmarking data tests for weighted vehicle routing problem. **IEEE Transactions on Automation Science and Engineering**, v. 11, n. 4, p. 1097–1109, 2014.

TANG, J. et al. A max–min ant system for the split delivery weighted vehicle routing problem. **Expert Systems with Applications**, v. 40, n. 18, p. 7468–7477, 2013. ISSN 0957-4174. Disponível em: <<https://www.sciencedirect.com/science/article/pii/S0957417413004673>>.

TANG, J.; ZHANG, J.; PAN, Z. A scatter search algorithm for solving vehicle routing problem with loading cost. **Expert Systems with Applications**, v. 37, n. 6, p. 4073–4083, 2010. ISSN 0957-4174. Disponível em: <<https://www.sciencedirect.com/science/article/pii/S0957417409009695>>.

TAO, Z. et al. A mixed pso algorithm for the vrpspd. In: **2008 Chinese Control and Decision Conference**. [S.I.: s.n.], 2008. p. 4017–4021.

TARANTILIS, C.; KIRANOUDIS, C.; VASSILIADIS, V. A threshold accepting metaheuristic for the heterogeneous fixed fleet vehicle routing problem. **European Journal of Operational Research**, v. 152, p. 148–158, 01 2004.

- TARANTILIS, C. D.; KIRANOUDIS, C. T.; VASSILIADIS, V. S. A list based threshold accepting metaheuristic for the heterogeneous fixed fleet vehicle routing problem. **Journal of the Operational Research Society**, Taylor & Francis, v. 54, n. 1, p. 65–71, 2003. Disponível em: <<https://doi.org/10.1057/palgrave.jors.2601443>>.
- TAVARES, L.; LOPES, H.; LIMA, C. E. Construction and improvement heuristics applied to the capacitated vehicle routing problem. In: . [S.I.: s.n.], 2009. p. 690–695.
- TOTH, P.; VIGO, D. 1. an overview of vehicle routing problems. In: \_\_\_\_\_. **The Vehicle Routing Problem**. [s.n.], 2002. p. 1–26. Disponível em: <<https://pubs.siam.org/doi/abs/10.1137/1.9780898718515.ch1>>.
- TOTH, P.; VIGO, D. **Vehicle Routing Problems, Methods, and Applications**. Second edition. [S.I.]: SIAM - Society for Industrial and Applied Mathematics Philadelphia, 2014.
- VIEIRA, H. P. Metaheuristica para a Solução de Problemas de Roteamento de Veículos com Janela de Tempo. **Universidade Estadual de Campinas - UNICAMP**, p. 6–8, 2013.
- WANG, X.; SHAO, S.; TANG, J. Iterative local-search heuristic for weighted vehicle routing problem. **IEEE Transactions on Intelligent Transportation Systems**, v. 22, n. 6, p. 3444–3454, 2021.
- WAZLAWICK, R. **Metodologia de pesquisa para ciência da computação**. [S.I.]: Elsevier Brasil, 2017. v. 2.
- WEI, L.; ZHUO, F. A variable neighborhood tabu search algorithm for the heterogeneous fleet vehicle routing problem with time windows. **2010 LEITS**, 2010.
- WEN, M. et al. An adaptive large neighborhood search for the larger-scale multi depot green vehicle routing problem with time windows. **Journal of Cleaner Production**, v. 374, p. 133916, 2022. ISSN 0959-6526. Disponível em: <<https://www.sciencedirect.com/science/article/pii/S0959652622034886>>.
- WILCOXON, F. Individual comparisons by ranking methods. **Biometrics Bulletin**, [International Biometric Society, Wiley], v. 1, n. 6, p. 80–83, 1945. ISSN 00994987. Disponível em: <<http://www.jstor.org/stable/3001968>>.
- WU, Y. et al. An improved adaptive large neighborhood search algorithm for the heterogeneous fixed fleet vehicle routing problem. p. 657–663, 2017.
- ZHANG, J.; TANG, J.; FUNG, R. Y. K. A scatter search for multi-depot vehicle routing problem with weight-related cost. **Asia-Pacific Journal of Operational Research**, v. 28, n. 03, p. 323–348, 2011.
- ZHENG, K.; LU, Z.; SUN, X. An effective parallel improving tabu search algorithm for heterogeneous fixed fleet vehicle routing problem. v. 1, p. 42–46, 2010.