



**CENTRO FEDERAL DE EDUCAÇÃO
TECNOLÓGICA DE MINAS GERAIS**

Diretoria de Pesquisa e Pós-Graduação

**Programa de Pós-Graduação em Modelagem
Matemática e Computacional**

Um Estudo com Abordagem Heurística para Problemas de Roteamento de Veículos com Janelas de Tempo e Múltiplos Depósitos

Tese de Doutorado apresentado ao Programa de Pós-Graduação em Modelagem Matemática e Computacional do CEFET-MG, como parte dos requisitos necessários do Exame de Qualificação de Doutorado em Modelagem Matemática e Computacional.

Aluno : Sinaide Nunes Bezerra

Orientador : Prof. Dr. Marcone Jamilson Freitas Souza (UFOP)

Coorientador : Prof. Dr. Sérgio Ricardo de Souza (CEFET-MG)

Belo Horizonte - MG
6 de junho de 2023

B574e Bezerra, Sinaide Nunes
Um estudo com abordagem heurística para problemas de roteamento de veículos com janelas de tempo e múltiplos depósitos / Sinaide Nunes Bezerra. – 2023. 105 f.

Tese de doutorado apresentada ao Programa de Pós-Graduação em Modelagem Matemática e Computacional.
Orientador: Marcone Jamilson Freitas Souza.
Coorientador: Sérgio Ricardo de Souza.
Tese (doutorado) – Centro Federal de Educação Tecnológica de Minas Gerais.

1. Programação heurística – Teses. 2. Otimização combinatória – Teses. 3. Frotas de veículos – Teses. 4. Software de aplicação – Teses. 5. Algoritmos – Teses. I. Souza, Marcone Jamilson Freitas. II. Souza, Sérgio Ricardo de. III. Centro Federal de Educação Tecnológica de Minas Gerais. IV. Título.

CDD 519.7



SERVIÇO PÚBLICO FEDERAL
MINISTÉRIO DA EDUCAÇÃO
CENTRO FEDERAL DE EDUCAÇÃO TECNOLÓGICA DE MINAS GERAIS
COORDENAÇÃO DO PROGRAMA DE PÓS-GRADUAÇÃO EM MODELAGEM MATEMÁTICA E COMPUTACIONAL

**“UM ESTUDO COM ABORDAGEM HEURÍSTICA PARA PROBLEMAS
DE ROTEAMENTO DE VEÍCULOS COM JANELAS DE TEMPO E
MÚLTIPLOS DEPÓSITOS”.**

Tese de Doutorado apresentada por **Sinaide Nunes Bezerra**, em 27 de fevereiro de 2023, ao Programa de Pós-Graduação em Modelagem Matemática e Computacional do CEFET-MG, e aprovada pela banca examinadora constituída pelos professores:

Prof. Dr. Marcone Jamilson Freitas Souza (Orientador)
Centro Federal de Educação Tecnológica de Minas Gerais

Prof. Dr. Sérgio Ricardo de Souza (Coorientador)
Centro Federal de Educação Tecnológica de Minas Gerais

Prof. Dr. Luiz Satoru Ochi
Universidade Federal Fluminense

Prof. Dr. Puka Huachi Vaz Penna
Universidade Federal de Ouro Preto

Profª. Drª. Elisângela Martins de Sá
Centro Federal de Educação Tecnológica de Minas Gerais

Profª. Drª. Elizabeth Fialho Wanner
Centro Federal de Educação Tecnológica de Minas Gerais

Visto e permitida a impressão,

Prof. Dr. Thiago de Souza Rodrigues
Coordenador do Programa de Pós-Graduação Stricto Sensu em
Modelagem Matemática e Computacional

Dedico a minha família.

Agradecimentos

Agradeço ao meu orientador Prof. Dr. Marcone Jamilson Freitas Souza e ao meu coorientador Prof. Dr. Sérgio Ricardo de Souza por me ajudarem nesta pesquisa de forma além da orientação. À minha esposa Laudiene e ao nosso filho Arthur pela compreensão e apoio durante toda a caminhada. Agradeço aos meus pais e irmãs por acreditarem no meu potencial. Aos amigos do Cefet-MG, em especial à Simone, pela parceria nos estudos e incentivo.

Agradeço ao Conselho Brasileiro de Desenvolvimento Científico e Tecnológico (CNPq), Fundação de Amparo à Pesquisa do Estado de Minas Gerais (FAPEMIG) e ao Departamento de Computação da Universidade Federal de Ouro Preto (UFOP) pelo uso de seus equipamentos na execução de testes. Agradeço especialmente ao Centro Federal de Educação Tecnológica de Minas Gerais (CEFET-MG) pelo apoio financeiro.

O que sabemos é uma gota; o que ignoramos é um oceano. Mas o que seria o oceano
se não infinitas gotas?
Isaac Newton

Resumo

Este trabalho trata de alguns problemas de roteamento de veículos com janelas de tempo e múltiplos depósitos. Nesse conjunto estão problemas com roteamento fechado e aberto que objetivam a redução na distância ou na quantidade de veículos. Esse conjunto de problemas é $\mathcal{NP}$ -difícil, pois eles têm como subproblema, o Problema de Roteamento de Veículos (PRV), onde os clientes são atendidos por uma frota de veículos de mesma capacidade homogênea, em que os custos são dados pela distância total percorrida ou quantidade total de veículos utilizados no atendimento aos clientes. Para resolver esses problemas, o *Smart General Variable Neighborhood Search* com Busca Local Adaptativa (SGVNSBLA) é apresentado. O algoritmo alterna o mecanismo de busca local entre duas estratégias diferentes. Na primeira estratégia, o método *Randomized Variable Neighborhood Descent* (RVND) realiza a busca local e, ao aplicar essa estratégia, a vizinhanças com os melhores vizinhos recebem uma pontuação maior. Na segunda estratégia, o método de busca local é aplicado apenas em uma única vizinhança, escolhida pelo método da roleta. Assim, a aplicação da primeira estratégia de busca local serve como método de aprendizagem para aplicação da segunda estratégia. Para testar esses algoritmos, usamos instâncias de *benchmark* do MDVRPTW envolvendo até 960 clientes, 12 depósitos e 120 veículos, num total de 48 instâncias teste. Para fins de comparação, outros algoritmos baseados em vizinhança também foram implementados. Os resultados encontrados permitiram avaliar a contribuição das estruturas de vizinhanças utilizadas nas soluções que buscam a redução na quantidade de veículos. Os resultados mostram que o desempenho do SGVNSBLA superou o dos outros algoritmos implementados visando a redução da frota. Houve uma redução da frota de veículos em 91,18% das instâncias avaliadas e o tamanho da frota alcançou uma redução média de até 23,32% em problemas com janelas de tempo e múltiplos depósitos. Ao se avaliar essas instâncias com roteamento aberto, houve redução na quantidade de veículos em 100% das instâncias, com baixa variabilidade.

Palavras-chaves: Problema de Roteamento de Veículos com Janela de Tempo e Múltiplos Depósitos. Busca Local Adaptativa. *Variable Neighborhood Search*.

Abstract

This work deals with vehicle routing problems with time windows and multiple depots. In this set, closed and open routing problems aim to reduce the distance or the number of vehicles. This set of problems is $\mathcal{NP}$ -hard because they have as sub-problem, the Vehicle Routing Problem (VRP), where customers are served by a fleet of vehicles with the same homogeneous capacity, where the total distance gives the cost traveled or the total number of vehicles used to serve customers. To solve these problems, *Smart General Variable Neighborhood Search* with adaptive local search (SGVNSBLA) is introduced. The algorithm switches the local search engine between two different strategies. In the first strategy, the Randomized Variable Neighborhood Descent (RVND) method performs the local search. When this strategy is used, the neighborhoods with the best neighbors receive a higher score. In the second strategy, the local search method is applied only in a single neighborhood, which is selected using the roulette wheel method. Thus, applying the first local search strategy serves as a learning method for applying the second strategy. We used MDVRPTW benchmark instances with up to 960 customers, 12 depots, and 120 vehicles in 48 test instances to test these algorithms. Other neighborhood-based algorithms were also implemented for comparison purposes. Based on the results found, it was possible to evaluate the contribution of neighborhood structures in solutions to reduce the number of vehicles. The results show that the performance of SGVNSBLA outperformed other fleet reduction algorithms. Vehicle fleet reduction occurred in 91.18% of the evaluated instances, and fleet size achieved an average reduction of up to 23.32% for problems with time windows and multiple deposits. In assessing these instances with open routing, a reduction in the number of vehicles occurred in 100% of the cases, with low variability.

Keywords: Multi-Depot Vehicle Routing Problem with Time Window. Adaptive Local Search. Variable Neighborhood Search.

Conteúdo

Lista de Tabelas	xiii
Lista de Figuras	xv
Lista de Algoritmos	xvi
1 Introdução	1
1.1 Apresentação Geral	1
1.2 Definição do Problema	1
1.3 Motivação	3
1.4 Objetivos e Contribuições	5
1.4.1 Objetivos Específicos	5
1.5 Principais Contribuições	5
1.6 Organização da Tese	6
2 Problemas de Roteamento de Veículos com Janelas de Tempo e Múltiplos Depósitos	7
2.1 Características Gerais dos problemas da classe PRVJTMD	8
2.2 Problemas da Classe PRVJTMD: descrição, revisão e modelos	9
2.2.1 Descrição do PRVJTMD e do PRVJTMD*	10
2.2.2 Revisão Bibliográfica sobre os problemas da classe PRVJTMD de roteamento fechado	14
2.2.3 Modelos Matemáticos Propostos para PRVJTMD com roteamento fechado	17
2.3 Problemas de Roteamento Aberto: PRAVJTMD e PRAJTMD*	21
2.3.1 Descrição do PRAVJTMD e do PRAVJTMD*	21
2.3.2 Revisão Bibliográfica sobre o PRAVJTMD e o PRAVJTMD*	23
2.3.3 Modelo Matemático Proposto para o PRVAJTMD e PRVAJTMD*	25
2.4 Resumo do Capítulo	25
3 Referencial Teórico sobre <i>Variable Neighborhood Search</i>	27
3.1 <i>Variable Neighborhood Search</i>	27
3.2 Procedimento de Perturbação	29
3.3 Mudança de vizinhanças	30
3.4 Procedimento de Busca Local	32
3.4.1 <i>Variable Neighborhood Descent</i>	32
3.4.2 <i>Randomized Variable Neighborhood Descent</i>	33

3.5	Resumo do Capítulo	34
4	Adaptações da Meta-heurística VNS Aplicadas aos PRVJTMDs Apresentados	35
4.1	Representação da Solução	35
4.2	Estruturas de Vizinhaça Propostas	36
4.2.1	Operadores Intrarrotas	37
4.2.2	Operadores Inter-Rotas	39
4.2.3	Operadores Inter-Depósitos	44
4.2.4	Operador de Destruição e Reconstrução	47
4.3	Determinação da Solução Inicial	47
4.3.1	Procedimento para Geração de <i>Clusters</i>	48
4.3.2	Procedimento para Geração de Rotas	49
4.4	Procedimento de Perturbação Proposto	52
4.5	Procedimentos de Busca Local Propostos	53
4.5.1	Procedimento Proposto de Busca Local Adaptativa	53
4.6	Funções de Avaliação	56
4.6.1	Função de Avaliação para Redução de Frota	56
4.6.2	Função de Avaliação para Redução da Distância	57
4.7	Algoritmos e Métodos Propostos	57
4.7.1	<i>Basic General VNS</i>	58
4.7.2	<i>Smart</i> GVNS Base Proposto	59
4.7.3	<i>Smart General Variable Neighborhood Search</i>	61
4.7.4	<i>Smart</i> GVNS com busca local RVND	61
4.7.5	Aceitação de um vizinho	62
4.8	Resumo do Capítulo	63
5	Experimentos Computacionais	64
5.1	Instâncias de Teste	64
5.2	Calibração de Parâmetros dos Algoritmos	65
5.3	Vizinhanças e Operadores Utilizados	67
5.4	Experimentos Computacionais para o PRVJTMD*	69
5.4.1	Resultados do Grupo I para o PRVJTMD*	69
5.4.2	Resultados do Grupo II para o PRVJTMD*	74
5.4.3	Influência das Estruturas de Vizinhanças na Busca Local	78
5.4.4	Influência das Vizinhanças na Perturbação	82
5.4.5	Influência da Eliminação Simultânea dos Operadores com Baixa Contribuição	83
5.5	Experimentos Computacionais para o PRVAJTMD e o PRVAJTMD*	84
5.5.1	Resultados do Grupo I para os problemas PRVAJTMD e PRVAJTMD*	85
5.5.2	Resultado do Grupo II para o PRVAJTMD e o PRVAJTMD*	88
5.5.3	Análise Estatística - PRVAJTMD e PRVAJTMD*	91

5.6	Resumo do Capítulo	95
6	Conclusões Gerais	97
6.1	Trabalhos Publicados	98
6.2	Propostas de Trabalhos Futuros	99
	Referências Bibliográficas	100

Lista de Tabelas

2.1	Notação utilizada para descrever os PRVJTMDs em estudo.	9
2.2	Dados dos exemplos mostrados nas Figuras 2.2(a) e 2.2(b)	14
2.3	Relação de autores e trabalhos expostos na revisão bibliográfica da classe de problemas PRVJTMD com roteamento fechado.	18
2.4	Dados para os exemplos do PRAVJTMD e do PRAVJTMD* mostrados na Figura 2.3.	22
2.5	Problemas de Roteamento Aberto: revisão bibliográfica	23
5.1	Instâncias selecionadas para o ajuste de parâmetros dos algoritmos usando IRACE.	66
5.2	Grupo de instâncias.	66
5.3	Faixa dos valores dos parâmetros testados e os resultados retornados pelo IRACE	67
5.4	Operadores utilizados pelos problemas tratados.	68
5.5	Resultados para o PRVJTMD* para SGVNSBLA, SGVNS e BGVNS referentes às instâncias do Grupo I.	70
5.6	p -valores para testes de hipóteses relativos a aleatoriedade, normalidade e homocedasticidade para o Grupo I com SGVNSBLA, SGVNS e BGVNS.	71
5.7	p -valores para testes de hipóteses de aleatoriedade, normalidade e homocedasticidade para o Grupo I, considerando apenas as situações em que houve empate no número de veículos utilizados.	72
5.8	Resultados do algoritmo SGVNSBLA para o PRVJTMD* sobre as instâncias do Grupo I em comparação com os resultados do algoritmo HGSADC para o PRVJTMD.	73
5.9	Resultados para PRVJTMD* do algoritmo SGVNSBLA versus algoritmos SGVNS e BGVNS referentes as instâncias do Grupo II.	75
5.10	p -valores para testes de hipóteses em relação a aleatoriedade, normalidade e homocedasticidade para o Grupo II com SGVNSBLA, SGVNS e BGVNS.	76
5.11	p -valores para testes de hipóteses de aleatoriedade, normalidade e homocedasticidade para o Grupo II, considerando apenas as situações em que houve empate no número de veículos utilizados.	76
5.12	Resultados do algoritmo SGVNSBLA para PRVJTMD* para instâncias do Grupo II comparados com os resultados do algoritmo HGSADC para PRVJTMD.	77
5.13	Subconjuntos definidos para analisar a influência dos operadores na busca local.	79

5.14	Resultados do algoritmo SGVNSBLA usando os conjuntos $\mathcal{LSO}_9$ e $\mathcal{LSO}_{11}$ de operadores de busca local em instâncias do Grupo I.	80
5.15	p -valores para testes de aleatoriedade, normalidade e homocedasticidade.	81
5.16	Subconjuntos de operadores usados para analisar a sua influência na perturbação.	82
5.17	Resultados do SGVNSBLA com o subconjunto $\mathcal{PO}_{2a}$	83
5.18	Subconjuntos com operadores usados para analisar a influência dos operadores nas fases de busca local e de perturbação.	84
5.19	Resultados para o PRVAJTMD e o PRVAJTMD* encontrados pelo SGVNSBLA aplicado ao Grupo I.	86
5.20	Resultados para PRVAJTMD e PRVAJTMD* encontrados pelo SGVNS-RVND aplicado ao Grupo I.	87
5.21	Grupo I: comparativo entre os resultados dos algoritmos SGVNSBLA e SGVNS-RVND para a solução do PRVAJTMD.	88
5.22	Comparativo entre SGVNSBLA e SGVNS-RVND para o PRVAJTMD* sobre resultados do Grupo I.	89
5.23	Resultados para o PRAJTMD e o PRVAJTMD* encontrados pelo SGVNSBLA aplicado ao Grupo II.	90
5.24	Resultados para o PRVAJTMD e o PRVAJTMD* encontrados pelo SGVNS-RVND aplicado ao Grupo II.	91
5.25	Comparativo entre SGVNSBLA e SGVNS-RVND para PRVAJTMD com resultados do Grupo II.	92
5.26	Comparativo entre SGVNSBLA e SGVNS-RVND para PRVAJTMD*: resultados do Grupo II.	93
5.27	Distribuição por quartis para o SGVNSBLA.	93
5.28	Distribuição por quartis para o algoritmo SGVNS-RVND.	95

Lista de Figuras

1.1	Exemplo do PRV clássico.	2
2.1	Ilustração das condições de chegada do veículo.	13
2.2	Exemplos de soluções para o PRVJTMD e PRVJTMD*.	13
2.3	Exemplo de soluções para o PRVAJTMD e PRVAJTMD*.	22
3.1	Exploração do espaço de busca pelo VNS. Fonte: Hansen et al. (2017)	28
4.1	Exemplo de uma solução com dois depósitos.	36
4.2	Exemplo de vizinhança <i>Reinsertion</i>	38
4.3	Exemplo de vizinhança <i>Swap</i>	38
4.4	Exemplo de vizinhança <i>Or-opt2</i>	38
4.5	Exemplo de vizinhança <i>2-Opt</i>	39
4.6	Exemplo de vizinhança <i>3-Opt</i>	39
4.7	Exemplos de reversão.	40
4.8	Exemplo de <i>Swap(1,1)</i>	40
4.9	Exemplo de vizinhança <i>Swap(2,1)</i>	41
4.10	Exemplo de vizinhança <i>Swap(2,2)</i>	41
4.11	Exemplo de vizinhança <i>Shift(1,0)</i>	42
4.12	Exemplo de vizinhança <i>Shift(2,0)</i>	42
4.13	Exemplos de vizinhança <i>2-Opt*</i> Alternativa 1	43
4.14	Exemplos de vizinhança <i>2-Opt*</i> Alternativa 2.	43
4.15	Exemplo de <i>Cross</i>	44
4.16	Exemplo de <i>Swap(1,1)-InterDepot</i>	45
4.17	Exemplo de vizinhança <i>Swap(2,1)-InterDepot</i>	45
4.18	Exemplo de vizinhança <i>Swap(2,2)-InterDepot</i>	46
4.19	Exemplo de <i>Shift(1,0)-InterDepot</i>	46
4.20	Exemplo de <i>ShiftDepot</i>	46
4.21	Exemplo de <i>SwapDepot</i>	47
4.22	Exemplo para geração da solução inicial.	48
4.23	Exemplo das estruturas de decisão para o método da roleta.	55
5.1	Percentual de contribuição dos operadores na busca local.	78
5.2	<i>Boxplots</i> com resultados do SGVNSBLA para Grupo I com operadores definidos em $\mathcal{LSO}_9$ e $\mathcal{LSO}_{11}$	81
5.3	Gráficos mostrando o número de inviabilidades encontradas nos subconjuntos $\mathcal{PO}_a - \mathcal{PO}_{2c}$	82

5.4	Gráfico de bolhas referente ao número de soluções viáveis e inviáveis encontradas pelo SGVNSBLA com os operadores definidos em $\mathcal{LSP}\mathcal{O}_{10a}$ e $\mathcal{LSP}\mathcal{O}_{10b}$	84
5.5	<i>Boxplots</i> dos resultados do algoritmo SGVNSBLA.	94
5.6	<i>Boxplots</i> dos resultados do algoritmo SGVNS-RVND.	94

Lista de Algoritmos

1	VNS Básico (s, v_{max}, t_{max})	29
2	perturbacaoBasica ($s, v, \mathcal{N}$)	30
3	vizinhancaSequencial(s, s', v)	30
4	vizinhancaCiclica (s, s', v)	31
5	vizinhancaPipe (s, s', v)	31
6	vizinhancaAssimetrica (s, s', v, θ)	32
7	VND($s, \mathcal{N}$)	33
8	RVND($s, \mathcal{N}$)	34
9	GerarClusters($\mathcal{C}, \mathcal{D}$)	50
10	CriarRotas($\mathcal{HD}, \mathcal{C}, \mathcal{D}$)	51
11	Perturbacao ($s, v, p, \mathcal{PO}$)	52
12	RVND($s, s', \mathcal{N}, sucesso$)	53
13	AtualizarSucesso($s, s', s'', v, sucesso$)	53
14	SingleLocalSearch($s, s', \mathcal{N}, sucesso$)	54
15	BLA($s, s', \mathcal{N}, sucesso, it, maxTreinamento$)	55
16	BGVNS($\mathcal{C}, \mathcal{D}, \mathcal{LSO}, \mathcal{PO}, iterMax, maxTempo$)	58
17	AtualizarVizinhanca ($s, s'', v, maxVizinhanca$)	58
18	SGVNSBLA($\mathcal{C}, \mathcal{D}, \mathcal{LSO}, \mathcal{PO}, iterMax, maxTempo, maxNivel,$ $maxTreinamento, maxItAdaptativo, maxVizPerturbacao, sucesso$)	59
19	AtualizarNivelPerturbacao ($s, s'', v, p, maxNivel, maxVizPerturbacao$)	60
20	AtualizarSolucao ($s, s'', iter$)	61
21	ControladorBLA($itAdaptativo, maxItAdaptativo, sucesso$)	61
22	SGVNS($\mathcal{C}, \mathcal{D}, \mathcal{LSO}, \mathcal{PO}, iterMax, maxTempo, maxNivel,$ $maxVizPerturbacao$)	62
23	SGVNS-RVND($\mathcal{C}, \mathcal{D}, \mathcal{LSO}, \mathcal{PO}, iterMax, maxTempo, maxNivel,$ $maxVizPerturbacao$)	62

Nomenclaturas

Conjuntos

$\mathcal{A}'$	Conjunto dos pares (i, d) de clientes i designados ao depósito d mais próximo à i
$\mathcal{B}'$	Conjunto dos pares (i, d) de clientes i designados ao segundo depósito d mais próximo à i
$\mathcal{HD}$	Conjunto dos pares (i, d) de clientes i designados aos depósitos d
$\mathcal{HR}$	Conjunto dos pares $(i, rd1d2)$
$\mathcal{K}$	Conjunto de veículos
$\mathcal{LSO}$	Conjunto de operadores utilizados na busca local
$\mathcal{N}$	Conjunto de vizinhanças
$\mathcal{PO}$	Estruturas de vizinhanças geradas na fase de perturbação
$\mathcal{R}$	Conjunto de listas que representam as rotas dos depósitos
$\mathcal{U}$	Conjunto de veículos utilizados

Outros Símbolos

$F(s)$	Função de Avaliação para Redução de Frota
L	Distância total percorrida pelos veículos em uma solução s
γ	Valor aleatório $\in [0, 1]$
$\phi(k)$	Violações na rota atendida pelo veículo k
φ	Violações nas rotas geradas
a_{ki}	Momento de chegada do veículo k ao cliente i
b_{ki}	Início de atendimento pelo veículo k ao cliente i
$f^{SGVNSBLA}$	Valor da função objetiva dado pelo SGVNSBLA
h_i	Tempo de serviço do cliente i
JTF	Limite superior da janela de tempo

JTI	Limite inferior da janela de tempo
K	Quantidade de veículos do conjunto $\mathcal{K}$
Q	Capacidade de um veículo
q_i	Demanda do cliente i
T	Duração máxima na rota
TS	Tempo de início de serviço

Ponderadores

α	Peso relativo à quantidade de veículos $ \mathcal{U} $
β	Peso relativo à distância L
ω^Q	Peso relativo à sobrecarga do veículo
ω^T	Peso relativo ao tempo total excedido na duração da rota do veículo k
ω^{TW}	Peso relativo aos atrasos na entrega de produtos pelo veículo k

Siglas

ALNS	Adaptive Large Neighborhood Search
BGVNS	Basic General Variable Neighborhood Search
BLA	Busca Local Adaptativa
BT	Busca Tabu
CEFET-MG	Centro Federal de Educação Tecnológica de Minas Gerais
GRASP	Greedy Randomized Adaptive Search
GVNS	General Variable Neighborhood Search
ILS	Iterated Local Search
LNS	Large Neighborhood Search
MRC	Melhor Resultado Conhecido
PCV	Problema do Caixeiro Viajante
PRV	Problema de Roteamento de Veículos
PRVC	PRV Capacitado
PRVJT	Problema de Roteamento de Veículos com Janela de Tempo
PRVJTDF	Problema de Roteamento de Veículos com Janelas de Tempo e Dimensionamento de Frota

PRVJTMD Problema de Roteamento de Veículos com Janela de Tempo e Múltiplos Depósitos

PRVJTMD Problema de Roteamento de Veículos com Janelas de Tempo e Múltiplos Depósitos

PRVJTMD* Problema de Roteamento de Veículos com Janelas de Tempo e Múltiplos Depósitos com Redução da Frota

RPDPTW Rich Pickup and Delivery Problem with Time Windows

RVND Randomized Variable Neighborhood Descent

SA Simulated Annealing

SGVNS-RVND Smart General Variable Neighborhood Search com RVND

SGVNSBLA Smart General Variable Neighborhood Search com Busca Local Adaptativa

UFOP Universidade Federal de Ouro Preto

VND Variable Neighborhood Descent

VNS Variable Neighborhood Search

Capítulo 1

Introdução

1.1 Apresentação Geral

Problemas de Roteamento de Veículos (PRV) são amplamente estudados na área de otimização combinatória e estão presentes em várias situações cotidianas do mundo moderno, como ao utilizar o GPS no trânsito, solicitar veículos de aluguel por aplicativos e mapear a distribuição de cargas e encomendas. A necessidade de descentralizar os pontos de distribuição de produtos, entre outros, tem contribuído para o aumento da complexidade das redes de distribuição de bens e serviços. Esses ajustes podem estar relacionados ao número de centros de distribuição, o tamanho da frota, a capacidade dos veículos alocados e a programação de suas atividades. A distribuição e os processos logísticos empresariais são ferramentas fundamentais para enfrentar mercados cada vez mais competitivos. Este contexto foi agravado a partir do ano de 2020 pelos enormes impactos da pandemia SARS-Cov-2 em todo o processo econômico global. Todo este contexto impulsionou o aumento das pesquisas sobre questões na gestão de processos, com a intenção de reduzir custos através do emprego de rotas mais curtas e menor quantidade de veículos envolvidos.

1.2 Definição do Problema

O Problema de Roteamento de Veículos em sua formulação clássica trata do atendimento de um determinado conjunto de clientes para a entrega dos produtos solicitados. Esse problema foi introduzido formalmente pela primeira vez por [Dantzig e Ramser \(1959\)](#). Na formulação proposta, os veículos partem de um depósito e devem completar uma determinada rota com o menor custo possível, dado pela distância percorrida para atender a todos os clientes. Na Figura [1.1](#) é apresentado um exemplo para o problema descrito.

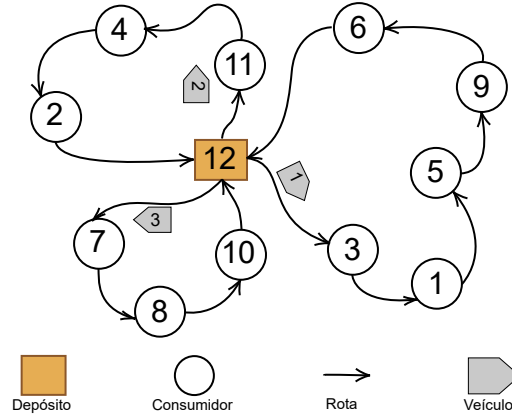


Figura 1.1: Exemplo do PRV clássico.

No exemplo apresentado na Figura 1.1, os consumidores são representados de 1 a 11 e o depósito pelo número 12. Existem três rotas, cada uma atendida por um dos três veículos disponíveis.

Numa formulação mais abrangente do PRV, cada veículo sai do depósito e executa uma rota por um determinado tempo máximo de duração, atendendo a certo número de clientes e retorna ou não ao depósito, garantindo que a demanda total na rota não exceda a capacidade do veículo. Essa formulação é denominada Problema de Roteamento de Veículos Capacitados (PRVC). Outras características podem ser adicionadas, tais como o intervalo de tempo para atendimento ao cliente (janelas de tempo), a redução na quantidade de veículos empregados na distribuição dos produtos e a quantidade de depósitos envolvidos. A adição de cada uma dessas características colabora para o aumento da complexidade do problema. Essas e outras características contribuem para gerar diferentes situações associadas ao PRVC, que podem envolver uma frota homogênea ou heterogênea, como apresentado em [Cordeau et al. \(1997\)](#); [Vidal et al. \(2012\)](#); [Subramanian et al. \(2013b\)](#); [Oliveira et al. \(2016\)](#). Segundo [Lenstra e Kan \(1981\)](#), os problemas de roteamento de veículos pertencem à classe de problemas $\mathcal{NP}$ -difícil, e a incorporação de restrições colabora para o aumento da complexidade computacional.

Dentre as características relatadas para o PRV, destacam-se nesta pesquisa a quantidade de depósitos envolvidos, as janelas de tempo, o roteamento aberto e a redução da frota de veículos utilizados. No PRV com Múltiplos Depósitos – PRVMD (*Multi-depot Vehicle Routing Problem* – MDVRP), a diferença para o PRV clássico está no fato de haver mais de um depósito envolvido na distribuição. No Problema de Roteamento de Veículos com Janela de Tempo – PRVJT (*Vehicle Routing Problem with Time Window* – VRPTW), as janelas são associadas a um cliente e indicam o intervalo em que esse cliente deve ser atendido. Definimos nesta tese que o PRV com

redução da frota consiste em determinar o menor conjunto de veículos que atenda a todos os clientes.

A integração entre as janelas para atendimento aos clientes e os vários depósitos estão caracterizadas no PRV com Janela de Tempo e Múltiplos Depósitos – PRVJTMD (*Multi-Depot Vehicle Routing Problem with Time Windows* – MDVRPTW). Nesse problema, as rotas são representadas por um circuito hamiltoniano, pois cada vértice do grafo é visitado uma única vez e termina no vértice de início (depósito). Para um problema com as mesmas características em que a rota é representada por um caminho Hamiltoniano, isto é, todos os vértices são visitados apenas uma única vez e a rota termina num vértice diferente da origem, tem-se o PRV Aberto com Janelas de Tempo e Múltiplos Depósitos – PRVAJTMD (*Multi-Depot Open Vehicle Routing Problem with Time Windows* – MDOVRPTW). Nessa tese, conjuntamente aos problemas PRVJTMD e PRVAJTMD, são incorporados a esses problemas o objetivo de minimização do número total de veículos usados. Os problemas com a adição desse objetivo são aqui denominados como PRVJTMD* e PRVAJTMD*, respectivamente.

Nesta tese, estudamos os problemas PRVJTMD*, PRVAJTMD e PRVJTMD* com frota homogênea. Apesar da existência de métodos exatos para a resolução do PRV e suas variantes, observa-se que instâncias com elevadas dimensões, na maioria dos casos, são resolvidas por meio de métodos heurísticos ([Laporte, 2009](#)). Neste contexto, para a resolução dos problemas elencados, são utilizados métodos meta-heurísticos baseados na exploração de vizinhanças.

1.3 Motivação

O crescimento populacional, o crescente aumento da informatização, a necessidade de descentralização dos pontos de vendas e, o aumento da variedade de produtos constituem alguns dos fatores que têm contribuído para o crescimento e para o aumento da complexidade das redes de distribuição de bens e serviços. Neste contexto, companhias que trabalham com o tráfego de mercadorias e pessoas têm estado cotidianamente sujeitas a readequações para reelaboração de seus trabalhos em atendimento às diferentes restrições relacionadas, por exemplo, à capacidade dos veículos e ao horário de suas atividades. Assim, a distribuição e os processos logísticos empresariais revelam-se importantes instrumentos para a conquista de mercados cada vez mais concorridos, desencadeando o aumento do interesse em pacotes de otimização. Este cenário contribui para o interesse em pesquisas que almejam melhorar o gerenciamento dos processos relacionados ao roteamento de veículos. [Toth e Vigo \(2002\)](#) afirmam que o uso de métodos computacionais em processos de distribuição resulta numa economia entre 5% à 20% nos custos de transporte. A redução dos custos pelo

emprego de rotas mais curtas e menor quantidade de veículos envolvidos são fatores importantes dessas melhorias.

Como o PRV é um problema da classe $\mathcal{NP}$ -difícil, por extensão, suas diversas variantes também o são, em especial aquelas aqui abordadas. Neste caso, a maioria dos algoritmos exatos desenvolvidos para resolver estes problemas não retornam respostas em um tempo aceitável e instâncias de dimensões elevadas, pois o tempo requerido no processo pode não ser factível para a tomada de decisão. Para amortizar esse impacto, opta-se pelo uso de meta-heurísticas. Elas são consideradas uma abordagem adequada para a obtenção de soluções de boa qualidade em um tempo de computação aceitável, o que torna o seu uso prático e essencial. Dentre as várias meta-heurísticas existentes, *Variable Neighborhood Search* (VNS), proposta por [Mladenović e Hansen \(1997\)](#), foi escolhida como método base para resolver os PRVMDs estudados neste trabalho. Ela é uma técnica já aplicada em vários outros problemas de otimização combinatória e apresentou bom desempenho em problemas de roteamento de veículos, como apresentados em [Polacek et al. \(2004, 2008\)](#); [Salhi et al. \(2014\)](#); [Rego e Souza \(2019\)](#); [Bezerra et al. \(2018\)](#) e [Bezerra et al. \(2023\)](#).

A principal característica do algoritmo VNS encontra-se no fato de esse método promover modificações na solução através da exploração do espaço de soluções pelas estruturas de vizinhanças geradas, em um processo denominado perturbação. A geração de diferentes estruturas de vizinhanças pode gerar novas trajetórias na busca da solução do problema, proporcionando, assim, mecanismos para não ficar preso em ótimos locais. Após a perturbação, inicia-se o processo de refinamento da solução, usando a fase de busca local. Nesta fase, a exploração da vizinhança ocorre no sentido de encontrar melhores vizinhos (soluções). A solução corrente é comparada com outras soluções adjacentes; assim que uma melhor é encontrada, a solução corrente é atualizada.

Neste trabalho é desenvolvida uma abordagem para a classe de problemas envolvendo múltiplos depósitos. A abordagem proposta é apresentada em três versões do algoritmo VNS, que são utilizadas na solução dos problemas relatados. Nessas versões, a fase de perturbação segue conforme proposta em [Rego e Souza \(2019\)](#), em que o nível da perturbação cresce gradativamente e, especificamente para o MDVRP, conforme proposto em [Bezerra et al. \(2018\)](#), em que considera-se o custo relativo a distância dos clientes aos depósitos. Na busca local, foram implementados mecanismos de seleção dos operadores, com a incorporação de memória para selecioná-los. O algoritmo *Randomized Variable Neighborhood Descent* – RVND, proposto em [Souza et al. \(2010\)](#) também é utilizado como um dos métodos de busca local. Outros algoritmos foram implementados para efeito de comparação com os métodos propostos e são apresentados no decorrer desta tese.

1.4 Objetivos e Contribuições

O objetivo geral dessa tese é propor algoritmos capazes de resolver os Problemas de Roteamento de Veículos com múltiplos depósitos descritos na Seção 1.2 com uso de meta-heurísticas. Estes problemas possuem restrições complexas, que estão presentes no cotidiano das empresas e pessoas que trabalham com transporte de carga ou passageiros. Para alcançar o objetivo exposto, extensões da meta-heurística VNS foram desenvolvidas e aplicadas na solução das instâncias de cada problema apresentado.

1.4.1 Objetivos Específicos

Nessa tese, temos o seguintes objetivos específicos:

- Fazer uma revisão de literatura sobre os métodos utilizados para resolver os problemas apresentados relacionados à classe de problemas com múltiplos depósitos descritos na Seção 1.2;
- Desenvolver um algoritmo baseado no VNS que incorpora memória na seleção das estruturas de vizinhanças na busca local para solucionar o PRVJTMD*, PRVAJTMD e PRVAJTMD*;
- Analisar o comportamento dos operadores utilizados na geração das estruturas de vizinhanças e exploração das soluções encontradas;
- Propor modelos matemáticos que ponderem a utilização/aceitação dos veículos.
- Desenvolver modelos computacionais que considerem a redução da frota de veículos como fator de maior ponderação aplicados ao PRVJTMD* e PRVAJTMD*.

1.5 Principais Contribuições

As principais contribuições desta pesquisa são:

1. Apresentação de modelos que promovem a redução na frota dos veículos utilizados;
2. Desenvolvimento de métodos heurísticos baseados na meta-heurística VNS;
3. Desenvolvimento de operadores de busca local eficientes para os PRVJTMDs com redução na frota de veículos;
4. Análise do uso de um mecanismo de memória na busca local utilizada no VNS;

5. Apresentação de modelos matemáticos para os problemas com redução da frota e com roteamento aberto.

Com o estudo, foi possível encontrar resultados ainda não explorados na literatura para os PRVJTMD com redução na quantidade de veículos utilizados, tanto para o problema com rotas fechadas (PRVJTMD*), quanto para o problema com rotas abertas (PRVAJTMD e PRVAJTMD*). Esses resultados foram comparados com as versões que priorizam a redução nas distâncias. Com isto foi possível mostrar o comportamento dos métodos empregados nas duas situações. Ainda foi possível ampliar o conhecimento relativos às seguintes situações:

- Métodos baseados em vizinhança alcançam bons resultados para a classe dos PRVJTMDs;
- Mudanças no processo de perturbação promovem melhorias nos resultados;
- Métodos baseados na exploração de vizinhanças mostraram-se eficientes na redução da frota de veículos.

Essas situações podem ser observadas nos resultados encontrados a partir dos testes computacionais realizados sobre os conjuntos de instâncias utilizadas.

1.6 Organização da Tese

O restante desta pesquisa está estruturado em seis capítulos distribuídos da seguinte maneira:

- No Capítulo 2 são apresentados os problemas de roteamento de veículos com múltiplos depósitos, assim como uma revisão dos trabalhos correlatos;
- No Capítulo 3 o método *Variable Neighborhood Search* é descrito;
- No Capítulo 4, as alterações propostas tendo como base o VNS, são propostas e são apresentados os algoritmos propostos na resolução dos problemas de roteamento de veículos estudados neste trabalho;
- No Capítulo 5 são apresentados os resultados obtidos;
- No Capítulo 6 são apresentadas as conclusões envolvendo a análise geral do trabalho desenvolvido, as considerações finais e as sugestões para trabalhos futuros.

Capítulo 2

Problemas de Roteamento de Veículos com Janelas de Tempo e Múltiplos Depósitos

Este capítulo apresenta os Problemas de Roteamento de Veículos com Janelas de Tempo e Múltiplos Depósitos (PRVJTMD) com roteamento fechado e com roteamento aberto, envolvendo, em ambos os casos, (i) somente a minimização da distância; (ii) a minimização do número de veículos usados, como objetivo primário, e a minimização da distância total percorrida, como objetivo secundário. Este capítulo está organizado como segue: Seção 2.1 descreve as características gerais dos problemas abordados. A Seção 2.2 apresenta o PRVJTMD, incluindo a respectiva revisão bibliográfica e modelo matemático, e introduz o Problema de Roteamento de Veículos com Janelas de Tempo e Múltiplos Depósitos visando a minimização do número de veículos usados, aqui denominado PRVJTMD*. incluindo, do mesmo modo, a revisão bibliográfica associada e a proposição de modelo matemático. A Seção 2.3 apresenta o Problema de Roteamento Aberto de Veículos com Janelas de Tempo e Múltiplos Depósitos (PRAVJTMD), incluindo, também, a revisão bibliográfica e o modelo matemático, e, além disso, introduz o Problema de Roteamento Aberto de Veículos com Janelas de Tempo e Múltiplos Depósitos com minimização do número de veículos usados (PRAVJTMD*), com a respectiva revisão bibliográfica e o modelo matemático proposto.

2.1 Características Gerais dos problemas da classe PRVJTMD

A classe de Problemas de Roteamento de Veículos com Janelas de Tempo e Múltiplos Depósitos (PRVJTMD) é uma generalização do Problema de Roteamento de Veículos com Janelas de Tempo (PRVJT). Nesta classe de problemas, considera-se projetar um conjunto de rotas que possibilite a minimização do custo total do transporte, atendendo a todos os clientes, a capacidade do veículo e o tempo máximo de duração da rota. A frota de veículos disponíveis pode ter capacidade de carga homogênea ou heterogênea.

Neste capítulo serão discutidas as variantes do PRVJTMD com roteamento fechado e com roteamento aberto. Nos problemas com roteamento fechado, o veículo inicia e finaliza a rota em um depósito, enquanto no problema com roteamento aberto o veículo inicia em um depósito e finaliza a rota no último cliente. São abordados os seguintes problemas:

- (i) Problema de Roteamento de Veículos com Janelas de Tempo e Múltiplos Depósitos – PRVJTMD;
- (ii) Problema de Roteamento de Veículos com Janelas de Tempo e Múltiplos Depósitos com Redução da Frota – PRVJTMD*;
- (iii) Problema de Roteamento de Veículos Aberto com Janelas de Tempo e Múltiplos Depósitos – PRVAJTMD;
- (iv) Problema de Roteamento de Veículos Aberto com Janelas de Tempo e Múltiplos Depósitos com Redução da Frota – PRVAJTMD*;

Nos problemas discutidos neste trabalho, as seguintes características são comuns.

- (i) há sempre mais que um depósito;
- (ii) cada consumidor é atendido por apenas um veículo;
- (iii) as janelas de tempo para atendimento aos clientes devem ser respeitadas;
- (iv) a frota de veículos é homogênea;
- (v) a capacidade dos veículos é conhecida;
- (vi) o total de demanda para cada rota não pode exceder a capacidade do veículo;
- (vii) cada rota possui um tempo máximo para ser percorrida;

(viii) o custo total da distribuição deve ser mínimo.

Segundo Brandão (2020), a resolução dos problemas de roteamento aberto com múltiplos depósitos compreendem os três subproblemas a seguir:

- (i) Atribuição dos clientes aos depósitos;
- (ii) Definição dos *clusters* de clientes atribuídos a cada depósito;
- (iii) Definição da sequência de roteamento dos clientes incluídos em cada *cluster*.

Esses subproblemas podem ser estendidos a qualquer problema de roteamento de veículos com múltiplos depósitos. O último subproblema corresponde ao problema de encontrar o caminho hamiltoniano ótimo, que é $\mathcal{NP}$ -difícil e, conseqüentemente, todos os outros problemas também o são. Por isso, algoritmos com soluções aproximadas ainda são necessários para a grande maioria das aplicações práticas.

Devido a complexidade dos modelos matemáticos, algoritmos e funções de avaliação descritos nesta tese, uma grande quantidade de acrônimos foram utilizados. Com o intuito de melhor elucidação nas descrições dos problemas em estudo, a Tabela 2.1, resume esses acrônimos.

Tabela 2.1: Notação utilizada para descrever os PRVJTMDs em estudo.

Acrônimo	Descrição
α	Peso relativo à quantidade de veículos utilizados
β	Peso relativo à quantidade de veículos utilizados
$\mathcal{C}$	Conjunto de clientes, $\mathcal{C} = \{1, 2, \dots, N\}$
$\mathcal{D}$	Conjunto de depósitos, $\mathcal{D} = \{N + 1, N + 2, \dots, N + M\}$
$\mathcal{V}$	Conjunto de vértices que representam depósitos e clientes, $\mathcal{V} = \mathcal{D} \cup \mathcal{C}$
$\mathcal{A}$	Conjunto de arcos que representam a conexão entre um depósito e um cliente ou entre dois clientes
$\mathcal{K}$	Conjunto de veículos
$\mathcal{K}_d$	Conjunto de veículos do depósito d
d	Depósito $\in \mathcal{D}$
q_i	Demanda do cliente i
h_i	Tempo de serviço (descarga) no cliente i
JTI_i	Início da janela de atendimento no cliente i
JTF_i	Fim da janela de atendimento no cliente i
MT	Tempo máximo permitido na rota
VC	Capacidade do veículo
T_k	Tempo gasto na rota pelo veículo k
WU_k	Tempo total de espera para descarga do veículo k
a_{ki}	Momento de chegada do veículo k ao cliente i
b_{ki}	Início de atendimento pelo veículo k ao cliente i

2.2 Problemas da Classe PRVJTMD: descrição, revisão e modelos

Esta seção descreve o Problema de Roteamento de Veículos com Janelas de Tempo e Múltiplos Depósitos em suas versões PRVJTMD e PRVJTMD*, com roteamento

fechado. A Subseção 2.2.1 apresenta estes dois problemas, incluindo exemplos ilustrativos para melhor compreensão das diferenças entre os problemas. A Subseção 2.2.2 apresenta a revisão bibliográfica associada. Por fim, a Subseção 2.2.3 introduz os modelos matemáticos desses dois problemas.

2.2.1 Descrição do PRVJTMD e do PRVJTMD*

O Problema de Roteamento de Veículos com Janela de Tempo e Múltiplos Depósitos é descrito em [Cordeau et al. \(2001\)](#). Desde de então, diversos trabalhos, abordam esse problema, nos termos da formulação inicialmente proposta, em que todos o objetivo é reduzir a distância total. Desde sua formulação inicial, vários fatores, como o crescimento populacional, a globalização, a crescente informatização, a necessidade de descentralização dos os pontos de venda, a grande variedade de produtos, entre outros, têm contribuído para o aumento da complexidade das redes de distribuição de bens e serviços. Neste contexto, empresas que lidam com a logística de mercadorias estão sujeitas a adequar continuamente seu trabalho em resposta a diferentes cenários. Esses ajustes podem estar relacionados ao número de centros de distribuição, tamanho da frota, a capacidade dos veículos alocados e o cronograma de suas atividades. Este contexto, agravado a partir de 2020 pelos enormes impactos da pandemia de SARS-Cov-2 em todo o processo econômico global, levou ao surgimento de pesquisas sobre questões da distribuição gestão de processos, visando reduzir custos empregando rotas mais curtas e menos veículos envolvido.

Por entendermos que o custo da utilização de um veículo é elevado, apresentaremos neste trabalho uma formulação que penaliza o uso dos veículos como o fator predominante na busca da solução dos problemas de roteamento de veículos com janelas de tempo e múltiplos depósitos. Para distingui-lo do PRVJTMD clássico, foi denominado por PRVJTMD*. Neste problema, objetiva reduzir o custo da distribuição, tendo como fator mais caro a quantidade de veículos utilizados. Esta formulação se aproxima do proposto em [Bräysy e Gendreau \(2005a\)](#), [Ombuki et al. \(2006\)](#) e [Nalepa e Blocho \(2016\)](#). Nestes trabalhos os autores ressaltam a importância na redução do número de veículos.

Nesta tese é proposto um modelo que atenda as restrições impostas pelo PRVJTMD com as características na redução da frota. Na formulação proposta, assim como em [Ombuki et al. \(2006\)](#), pondera-se como fator de maior custo a quantidade de veículos utilizados. Por não existir na literatura está imposição de forma explícita, acrescentamos nas formulações em que se objetiva a redução da frota de veículos. Neste trabalho, está imposição está descrita na formulação e modelos matemáticos apresentados para o PRVJTMD* e PRVAJTMD*, em que:

O Problema de Roteamento de Veículos com Janelas de Tempo e Múltiplos Depósitos (PRVJTMD) foi descrito inicialmente em [Cordeau et al. \(2001\)](#). A formulação proposta por estes autores tem, como objetivo, a redução da distância total percorrida. Desde então, diversos trabalhos abordam esse problema nos mesmos termos da formulação inicialmente proposta. Ao longo desse texto, para simplificar, essa formulação será denominada como PRVJTMD clássico. Por entender que o custo da utilização de um veículo é elevado, apresenta-se nesta tese uma formulação que penaliza o uso dos veículos como o fator predominante na busca da solução dos problemas de roteamento de veículos com janelas de tempo e múltiplos depósitos. Para distingui-lo do PRVJTMD clássico, denomina-se como PRVJTMD* ao Problema de Roteamento de Veículos com Janelas de Tempo e Múltiplos Depósitos visando a minimização do número de veículos usados. Assim, neste problema, o objetivo de redução do custo da distribuição vê como fator mais caro a quantidade de veículos utilizados. Esta formulação desse problema, portanto, segue a formulação do Problema de Roteamento de Veículos com Janela de Tempo (PRVJT), exposta por autores como [Solomon \(1987\)](#), [Bräysy e Gendreau \(2005a,b\)](#), [Ombuki et al. \(2006\)](#) e [Nalepa e Blocho \(2016\)](#), dentre outros. Nestes trabalhos, os autores ressaltam a importância na redução do número de veículos. Nesta tese é proposto um modelo que atende às restrições impostas pelo PRVJTMD com as características na redução da frota. Na formulação proposta, assim como em [Ombuki et al. \(2006\)](#), pondera-se, como fator de maior custo, a quantidade de veículos utilizados. Por não existir na literatura está imposição de forma explícita, conforme apontado na revisão bibliográfica apresentada na Subseção 2.2.2, acrescentamos nas formulações em que se objetiva a redução da frota de veículos. Neste trabalho, esta imposição está descrita na formulação e no modelo matemático apresentado para o PRVJTMD*:

- O número total de veículos utilizados deve ser minimizado, seguido da redução na distância total percorrida.

Segundo [Christiaens e Vanden Berghe \(2020\)](#), a redução da frota reflete a necessidade das empresas de transporte do mundo real para acomodar suas solicitações. Neste sentido, a proposta busca melhor adequação econômica aos custos de transporte com a redução da quantidade de veículos envolvidos na entrega de mercadorias.

A seguir descreve-se as formulações para o PRVJTMD e para o PRVJTMD*. Este modelo é uma adaptação das definições propostas por ([Cordeau et al., 2001](#)), ([Polacek et al., 2004](#)) e ([Montoya-Torres et al., 2015](#)). Nesta tese ele é definido por um grafo completo $\mathcal{G} = (\mathcal{V}, \mathcal{A})$, sendo $\mathcal{V}$ um conjunto de $(N + M)$ vértices e $\mathcal{A}$ um conjunto de arcos, em que N clientes são atendidos por M depósitos. Particionando-se $\mathcal{V}$ em dois subconjuntos, quais sejam, o conjunto de clientes a serem atendidos, representados

por $\mathcal{C} = \{1, 2, \dots, N\}$, e o conjunto de depósitos $\mathcal{D} = \{N + 1, N + 2, \dots, N + M\}$, tem-se $\mathcal{V} = \mathcal{D} \cup \mathcal{C}$ e $\mathcal{D} \cap \mathcal{C} = \emptyset$. Cada depósito possui uma frota de veículos homogênea, tendo cada veículo capacidade VC e tempo máximo de duração da rota MT . O conjunto de todos os veículos é representado por $\mathcal{K}$ e $|\mathcal{K}| = K$. O atendimento ao cliente deve ocorrer respeitando-se um intervalo dado por $[JTI_i, JTF_i]$. Para cada consumidor i , existe um valor não negativo de demanda q_i e um tempo de serviço h_i , que deve iniciar em $b_{ki} \mid JTI_i \leq b_{ki} \leq JTF_i$. Para o depósito d , tanto a demanda q_d quanto o tempo de serviço h_d têm valor zero (0). Existe um custo não negativo c_{ij} associado a cada arco $(i, j) \in \mathcal{A}$, que representa a distância, o tempo gasto ou uma taxa entre os vértices i e j . A ele é adicionado o custo operacional de atendimento à cada cliente. Para compatibilidade das unidades de custo e tempo, as dimensões são convertidas com a variável auxiliar $w_{i,j}$, $\forall (i, j) \in \mathcal{A}$.

Estes problemas envolvem aspectos espaciais e temporais das movimentações dos veículos. Esses aspectos aumentam consideravelmente sua complexidade, pois, além de incorporar a restrição de capacidade, considera-se o intervalo de tempo em que o cliente deve ser atendido. Cada consumidor i deve ser atendido em uma janela com intervalo $[JTI_i, JTF_i]$, em que JTI_i corresponde ao limite inferior autorizado para o início do trabalho e JTF_i o limite superior. Quando um veículo k sai de um vértice i , que pode ser um cliente ou depósito, e chega em um vértice j , o início do serviço de descarga inicia-se imediatamente, ou após, à chegada, que é dado por a_{kj} . Neste atendimento não há vínculo entre o tempo de serviço h_i , associado à descarga do produto, e a janela de tempo. A Figura 2.1 ilustra as possíveis situações encontradas na entrega dos produtos em relação a janela do cliente.

Nesta representação, as três primeiras ocorrências resultam na entrega da encomenda. Na primeira situação, indicada por (a), o veículo chega ao cliente i antes do limite inferior da janela, o que acarreta em um tempo de espera para o início do atendimento, $a_{kj} \leq JTI_j$, indicando que o atendimento b_{kj} iniciará assim que a janela de atendimento JTI_j for aberta. Nas situações (b) e (c), o veículo chega ao cliente dentro da janela permitida para o atendimento, em que $JTI_j \leq a_{kj} \leq JTF_j$, iniciando imediatamente a entrega do produto. No cenário representado em (b), a descarga do produto ocorre integralmente no intervalo da janela de tempo; em (c), $(a_{kj} + h_i) > JTF_i$. Na última ocorrência, representada na letra (d), o veículo chega ao cliente após o limite superior da janela e a entrega não poderá ser efetivada. Assim, um veículo inicia imediatamente sua descarga somente quando chega ao cliente durante o intervalo definido.

Na Figura 2.2 são mostrados dois exemplos de problemas de roteamento de veículos com múltiplos depósitos e janelas de atendimento. Nos dois casos, existem dez clientes e dois depósitos. A Figura 2.2(a) apresenta uma solução para o PRVJTMD,

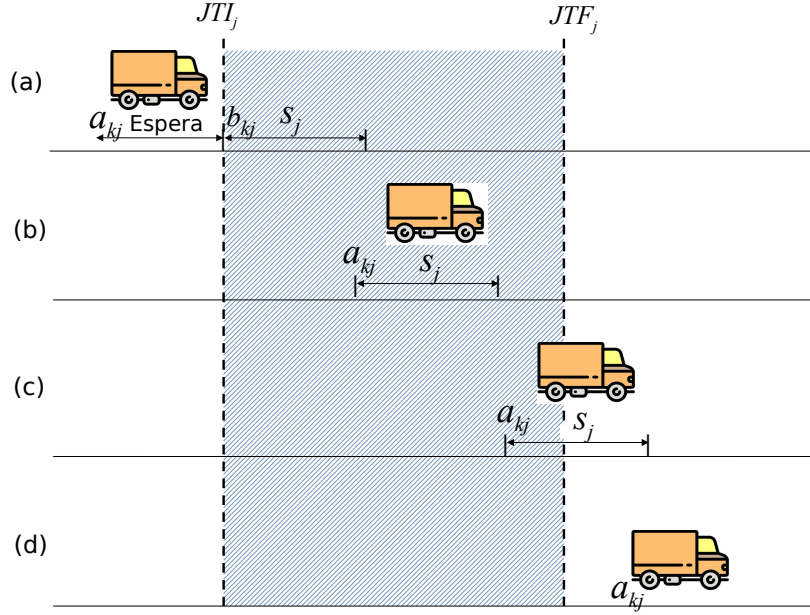
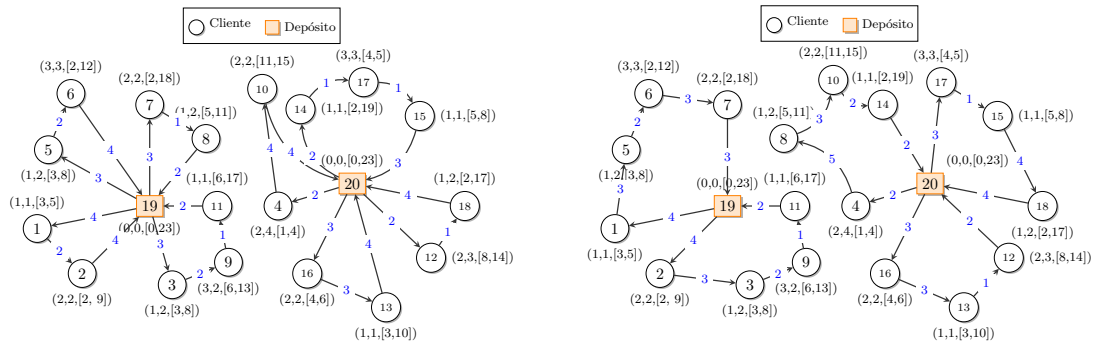


Figura 2.1: Ilustração das condições de chegada do veículo.

enquanto a Figura 2.2(b) mostra, para o mesmo conjunto de dados, uma solução para o PRVJTMD*. Nesta representação, cada veículo tem capacidade de carga $VC = 7$, com tempo máximo disponível para atendimento de uma rota $MT = 23$. Cada arco tem um custo c_{ij} , dado pelo tempo de viagem entre o nó i ao nó j . Para cada nó i , é associada uma tupla $(q_i, h_i, [JTI_i, JTF_i])$, com a demanda q_i , o tempo de serviço gasto no atendimento h_i e a janela de tempo $[JTI_i, JTF_i]$.



(a) Exemplo de uma solução para o PRVJTMD. Note que o número de rotas é igual à quantidade de veículos por depósito, em que, $|\mathcal{K}_{19}| = |\mathcal{K}_{20}| = 4$.

(b) Exemplo de uma solução para o PRVJTMD*. Note que existem cinco rotas, em que, $|\mathcal{K}_{19}| = 2$ and $|\mathcal{K}_{20}| = 3$.

Figura 2.2: Exemplos de soluções para o PRVJTMD e PRVJTMD*.

Na solução mostrada na Figura 2.2(a) para o PRVJTMD, o número de veículos é fixo e igual ao tamanho da frota disponível. No caso da solução mostrada na Figura 2.2(b) para o PRVJTMD*, uma quantidade menor de veículos é utilizada,

garantindo as restrições desse problema e mostrando o efeito da minimização do número de veículos usados. Neste caso, cinco veículos são utilizados em contraposição aos oito veículos disponíveis. É importante mostrar que, nesta variante, para o mesmo conjunto de depósitos e clientes, é possível encontrar soluções compatíveis em termos de custos de viagem com um menor número de veículos usados.

A Tabela 2.2 mostra os dados para as duas soluções apresentadas na Figura 2.2. Nessa tabela, para cada veículo são apresentadas as rotas na coluna “Rotas”; o tempo de viagem c_{ij} entre os vértices i e j ; o tempo de descarga h_i para cada vértice na rota; a demanda q_i para cada vértice; o tempo total de espera para descarga na rota WU_k pelo veículo k ; o tempo total gasto na rota $T_k = \sum_{i,j \in V} (c_{ij} + h_i)$ pelo veículo k . O custo final de cada problema são informados na última linha da tabela.

Tabela 2.2: Dados dos exemplos mostrados nas Figuras 2.2(a) e 2.2(b)

Dados da figura Figure 2.2(a)							Dados da Figure 2.2(b)							
k	Routes	c_{ij}	h_i	WU_k	T_k	q_i	Q_k	Routes	c_{ij}	h_i	WU_k	T_k	q_i	Q_k
1	(19, 5, 6, 19)	(3, 2, 4)	(0,2, 3, 0)	0	14	(0,1, 3, 0)	4	(19, 1, 5, 6, 7, 19)	(4, 3, 2, 3, 3)	(0,1, 2, 3, 2, 0)	0	23	(0,1, 1, 3, 2, 0)	7
2	(19, 7, 8, 19)	(3, 1, 2)	(0,2, 2, 0)	0	10	(0,2, 1, 0)	3	(19, 2, 3, 9, 11, 19)	(4, 3, 2, 1, 2)	(0,2, 2, 2, 1, 0)	0	19	(0,2, 1, 3, 1, 0)	7
3	(19, 3, 9, 11, 19)	(3, 2, 1, 2)	(0,2, 2, 1, 0)	0	13	(0,1, 3, 1, 0)	5	(20, 4, 8, 10, 14, 20)	(2, 5, 3, 2, 2)	(0,4, 2, 2, 1, 0)	0	23	(0,2, 1, 2, 1, 0)	6
4	(19, 1, 2, 19)	(4, 2, 4)	(0,1, 2, 0)	0	13	(0,1, 2, 0)	3	(20, 17, 15, 18, 20)	(3, 1, 4, 4)	(0,3, 1, 2, 0)	1	19	(0,3, 1, 1, 0)	5
5	(20, 4, 10, 20)	(2, 4, 4)	(0,4, 2, 0)	0	16	(0,2, 2, 0)	4	(20, 16, 13, 12, 20)	(3, 3, 1, 2)	(0,2, 1, 3, 0)	1	16	(0,2, 1, 2, 0)	5
6	(20, 14, 17, 15, 20)	(2, 1, 1, 3)	(0,1, 3, 1, 0)	0	12	(0,1, 3, 1, 0)	5							
7	(20, 12, 18, 20)	(2, 1, 4)	(0,3, 2, 0)	0	12	(0,2, 1, 0)	3							
8	(20, 16, 13, 20)	(3, 3, 4)	(0,2, 1, 0)	0	13	(0,2, 1, 0)	3							
Total					103							100		

2.2.2 Revisão Bibliográfica sobre os problemas da classe PRVJTMD de roteamento fechado

O PRVJTMD foi originalmente introduzido por Cordeau et al. (2001). Nesse trabalho, os autores apresentaram uma solução utilizando a meta-heurística Busca Tabu – BT (*Tabu Search* – TS). Os autores, além disso, introduziram o primeiro conjunto de instâncias apropriado para experimentos com este problema, que continua sendo usado, inclusive nessa tese, como referência para experimentos atuais. Este conjunto de instâncias está apresentado na Seção 5.1. Em Cordeau et al. (2004), os autores aprimoraram a proposta original de solução que haviam apresentado, produzindo novos resultados e, dentre eles, cinco ainda permanecem como os melhores obtidos para as respectivas instâncias.

Polacek et al. (2004) propuseram um algoritmo baseado em VNS, com critérios de aceitação para as melhores soluções, juntamente com a escolha de vizinhanças na busca local, utilizando, para os experimentos computacionais, o mesmo grupo de instâncias de Cordeau et al. (2001). As avaliações realizadas mostraram que o algoritmo apresentou resultados competitivos em relação ao algoritmo BT de Cordeau et al. (2001) no que diz respeito à qualidade da solução obtida e ao esforço computacional necessário. Polacek et al. (2008) apresentaram uma versão paralela do

algoritmo VNS introduzido em [Polacek et al. \(2004\)](#), com o uso de novas estratégias de intensificação e diversificação por meio de esquemas de adaptação e cooperação. Os resultados mostraram que todos os melhores resultados conhecidos na literatura para as instâncias de [Cordeau et al. \(2001\)](#) foram encontrados. Além disso, esse algoritmo encontrou 11 novos melhores resultados nas 20 instâncias testadas.

[Tansini e Viera \(2006\)](#) apresentaram heurísticas para agrupamento de clientes aos depósitos, utilizando, como referência, janelas de tempo e distâncias na avaliação da proximidade entre clientes e depósitos. O método utilizou atribuições gulosas e permitiu obter uma redução entre 5% e 6% da distância percorrida nas instâncias testadas.

[Ting e Chen \(2008\)](#), por sua vez, introduziram uma meta-heurística híbrida envolvendo *Multiple Ant Colony System* – MACS e *Simulated Annealing* – SA. Esta meta-heurística híbrida obteve três novos resultados para as instâncias de [Cordeau et al. \(2001\)](#). [Cordeau e Maischberger \(2012\)](#) apresentaram uma meta-heurística híbrida paralela, combinando *Iterated Local Search* – ILS e Busca Tabu para promover uma melhor exploração do espaço de soluções do problema. Esse algoritmo híbrido alcançou seis novos resultados com pequenas melhorias em relação aos valores encontrados em [Polacek et al. \(2008\)](#).

[Tamashiro et al. \(2010\)](#) apresentaram uma abordagem da meta-heurística Busca Tabu combinada com um método de economia estendida para resolver o PRVJTMD. O algoritmo híbrido proposto consiste em duas fases. A primeira fase é uma implementação do método de Busca Tabu aplicada para resolver o problema de atribuição de clientes e, ao mesmo tempo, levar à redução da vizinhança. Na segunda fase, busca-se minimizar a distância das rotas associadas aos veículos.

[Noori e Ghannadpour \(2012\)](#) propuseram uma meta-heurística híbrida de três níveis para resolver o PRVJTMD. No primeiro nível aplica-se o método “*Cluster first, Route second*”, associando os clientes aos depósitos; no segundo nível, para cada *cluster*, aplica-se uma implementação de Algoritmo Genético – AG (*Genetic Algorithm* – AG) para otimizar as rotas; e, na terceira fase, as soluções são refinadas, com uma meta-heurística de Busca Local. Os resultados obtidos por esta meta-heurística híbrida alcançaram os resultados mostrados em [Polacek et al. \(2004\)](#).

[Vidal et al. \(2013a,b\)](#) propuseram, nesta série de artigos, um algoritmo genético híbrido com gerenciamento adaptativo de diversidade, denominando *Hybrid Genetic Search with Advanced Diversity Control* – HGSADC, para resolver uma ampla gama de problemas derivados do PRVJT. Para o PRVJTMD, o algoritmo encontrou nove das melhores soluções conhecidas e alcançou todas as outras melhores soluções conhecidas para o conjunto de instâncias de [Cordeau et al. \(2001\)](#). Adicionalmente, os autores propuseram um novo conjunto de instâncias para o PRVJTMD, no qual a

frota possui maior número de veículos e há maior número de depósitos. Este conjunto é apresentado na Seção 5.1.

Montoya-Torres et al. (2015) apresentaram uma revisão de literatura sobre o PRVMD, avaliando artigos publicados até 2014. Segundo os autores, 88,44% dos trabalhos avaliados trataram este problema considerando um único objetivo e 25% destes foram relacionados ao PRVJTMD. Os autores listaram diversas variações do PRVJTMD, sem, no entanto, mostrar autores que considerassem a variante PRVJTMD* aqui tratada. Dentre essas variantes, destacamos a proposta de Li et al. (2016), em que os autores trataram o PRVJTMD com depósito compartilhado, gerando a variante denominada *Multi-Depot Vehicle Routing Problem with Time Windows under Shared Depot Resources* – (MDVRPTWSDR). Nessa variante, os veículos saem de um determinado depósito, mas podem retornar para um depósito diferente. Um algoritmo genético híbrido com busca local adaptativa, denominado *Hybrid Genetic Algorithm with Adaptive Local Search* – (HGA_ALS), é proposto para resolver o problema. Os autores utilizaram, para a realização dos experimentos computacionais, dez das vinte instâncias descritas em Cordeau et al. (2001). Wang et al. (2019b) representam uma importante continuidade da abordagem de Li et al. (2016), ao incluir a questão ambiental como fundamental para futuras pesquisas nessa classe de problemas.

Na literatura, soluções para os problemas de roteamento de veículos com o uso de métodos baseados na exploração de vizinhanças são amplamente utilizados. Belloso et al. (2019) propõem um algoritmo baseado na meta-heurística *Greedy Randomized Adaptive Search with Memory Programming* – GRASP para solucionar o Problema de Roteamento de Veículos com Coleta e Entrega (PRVCE). Os resultados mostraram uma abordagem competitiva ao encontrar 20 novas soluções para as 36 instâncias avaliadas. Rezgui et al. (2019) propõem uma versão do procedimento *Smart General Variable Neighborhood Search* para solucionar o PRVJT adaptado ao uso de veículos elétricos. Nesse trabalho, as instâncias de Solomon (1987) são utilizadas para avaliação do método proposto.

Em Karakostas et al. (2020a) foi introduzido o problema denominado *Fleet-size and Mix Pollution Location-Inventory-Routing Problem with Just-inTime replenishment policy and Capacity Planning*. Nesse artigo, os autores estendem as decisões de nível estratégico de um Problema de Roteamento de Veículos e Alocação de Estoque (*Location-Inventory-Routing Problem* – LIRP) clássico considerando decisões de seleção de capacidade e composição heterogênea da frota. Os autores propuseram um algoritmo baseado no procedimento *General Variable Neighborhood Search* – (GVNS) (Hansen et al., 2010), com seis diferentes estratégias de busca local, que utilizam o método *Variable Neighborhood Descent* – (VND) (Mladenović e Hansen, 1997). Em

[Karakostas et al. \(2020b\)](#), três versões do procedimento SGVNS são propostas para a solução de uma estrutura de distribuição de suprimentos no atendimento a pacientes de imunoterapia.

[Ren et al. \(2020\)](#) investigam o Problema de Roteamento Sustentável de Veículos com Janelas de tempo (*Green Vehicle Routing Problem with Time Window – GVRPTW*) em uma formulação bi-objetivo. Os autores buscam minimizar a poluição causada por veículos que utilizam gasolina e óleo diesel como combustíveis com a inserção de veículos menos poluentes (elétricos). O outro objetivo é minimizar o tempo de atraso total, por estar ligado diretamente à satisfação dos clientes. Para solucionar o problema bi-objetivo, um algoritmo baseado na meta-heurística VNS é proposto para encontrar a fronteira de Pareto do modelo proposto. Para os testes, são utilizadas como referência as instâncias de Solomon ([Solomon, 1987](#)).

Um procedimento de Busca Local Adaptativa (BLA) é utilizada por [Masmoudi et al. \(2021\)](#) para melhorar os mecanismos de intensificação e diversificação em um problema que envolve uma frota com vários tipos de veículos com capacidades diferentes.

Outra variante de impacto a ser destacada é a que envolve uma frota heterogênea, como [Rabbouch et al. \(2018\)](#) e [Abdallah e Ennigrou \(2020\)](#). Para concluir esta revisão literária, vale destacar que os artigos que tratam da formulação multiobjetivo do PRVJTMD, como [Wang et al. \(2019a\)](#) e [Li et al. \(2019\)](#), têm alcançado grande audiência recentemente, principalmente por abordarem questões ambientais.

A Tabela 2.3 resume as técnicas utilizadas para a solução, os anos de publicação e a lista de autores associados a cada um dos artigos relatados nesta seção.

A principal conclusão advinda da análise desta série de artigos é que, embora diversas e significativas variações de PRVJTMD tenham sido abordadas na literatura, a formulação desse problema envolvendo a minimização do número de veículos usados não é tratada, pelo menos em trabalhos do conhecimento do autor dessa tese. Assim, a introdução do PRVJTMD* é uma das principais contribuições da tese.

2.2.3 Modelos Matemáticos Propostos para PRVJTMD com roteamento fechado

Esta subseção apresenta os modelos matemáticos para a classe PRVJTMD com roteamento fechado. A Subseção 2.2.3.1 trata do modelo matemático do PRVJTMD. A Subseção 2.2.3.2 introduz o modelo matemático para o PRVJTMD*.

Tabela 2.3: Relação de autores e trabalhos expostos na revisão bibliográfica da classe de problemas PRVJTMD com roteamento fechado.

Problema	Técnica	Ano	Autor
PRVJTMD	BT	2001	Cordeau et al. (2001)
	BT	2004	Cordeau et al. (2004)
	VNS	2004	Polacek et al. (2004)
	Greedy	2006	Tansini e Viera (2006)
	VNS paralelo	2008	Polacek et al. (2008)
	MACS e SA	2008	Ting e Chen (2008)
	ILS-BT	2012	Cordeau e Maischberger (2012)
	BT	2010	Tamashiro et al. (2010)
	AG	2012	Noori e Ghannadpour (2012)
	HGSADC	2013a; 2013b	Vidal et al. (2013a,b)
	Review	2015	Montoya-Torres et al. (2015)
	HGA_ALS	2016	Li et al. (2016)
	HGA_ALS	2019b	Wang et al. (2019b)
	GRASP	2019	Belloso et al. (2019)
	SGVNS	2019	Rezgui et al. (2019)
	LIRP	2020a	Karakostas et al. (2020a)
	SGVNS	2020b	Karakostas et al. (2020b)
	VNS	2020	Ren et al. (2020)
	BLA	2021	Masmoudi et al. (2021)

2.2.3.1 Modelo Matemático para o PRVJTMD

A formulação matemática para o PRVJTMD foi construída a partir do modelo apresentado em Li et al. (2016). As seguintes variáveis de decisão são, então, definidas:

$$x_{ijk} = \begin{cases} 1, & \text{se veículo } k \text{ visita o cliente } j \text{ após o cliente } i \\ 0, & \text{caso contrário.} \end{cases}$$

A partir dessa definição, a formulação para o PRVJTMD é dada por:

$$\min \sum_{k \in \mathcal{K}} \sum_{i \in \mathcal{V}} \sum_{j \in \mathcal{V}} c_{ij} x_{kij}, \quad (2.1)$$

Sujeito a:

$$\sum_{k \in \mathcal{K}} \sum_{j \in \mathcal{V}} x_{kji} = 1, \quad \forall i \in \mathcal{C}, \quad (2.2)$$

$$\sum_{j \in \mathcal{V}} x_{kji} = \sum_{j \in \mathcal{V}} x_{kij}, \quad \forall i \in \mathcal{C}, \forall k \in \mathcal{K}, \quad (2.3)$$

$$\sum_{j \in \mathcal{C}} x_{kdj} = \sum_{j \in \mathcal{C}} x_{kjd}, \quad \forall k \in \mathcal{K}, d \in \mathcal{D}, \quad (2.4)$$

$$\sum_{d \in \mathcal{D}} \sum_{j \in \mathcal{C}} x_{kdj} \leq 1, \quad \forall k \in \mathcal{K}, \quad (2.5)$$

$$\sum_{i \in \mathcal{L}} \sum_{j \in \mathcal{L}} x_{kij} \leq |\mathcal{L}| - 1, \quad \forall \mathcal{L} \subseteq \mathcal{C}, k \in \mathcal{K}, \quad (2.6)$$

$$\sum_{i \in \mathcal{D}} \sum_{j \in \mathcal{D}} x_{kij} = 0, \quad \forall k \in \mathcal{K}, \quad (2.7)$$

$$x_{kij}(b_{ki} + h_i + c_{ij} - b_{kj}) \leq 0, \quad \forall i, j \in \mathcal{V}, k \in \mathcal{K}, \quad (2.8)$$

$$b_{ki} \geq a_{ki} \geq 0, \quad \forall i \in \mathcal{C}, k \in \mathcal{K}, \quad (2.9)$$

$$JTI_i \leq b_{ki} \leq JTF_i, \quad \forall i \in \mathcal{C}, k \in \mathcal{K}, \quad (2.10)$$

$$\sum_{i \in \mathcal{V}} \sum_{j \in \mathcal{C}} q_j x_{kij} \leq VC, \quad \forall k \in \mathcal{K}, \quad (2.11)$$

$$\sum_{i \in \mathcal{V}} \sum_{j \in \mathcal{V}} (c_{ij} + h_j) x_{kij} \leq MT, \quad \forall k \in \mathcal{K}, \quad (2.12)$$

$$x_{kij} \in \{0, 1\}, \quad \forall i, j \in \mathcal{V}, k \in \mathcal{K}. \quad (2.13)$$

A expressão (2.1) representa a função objetivo do problema, a ser minimizada. A função objetivo é calculada pela soma dos custos das viagens. As restrições (2.2) garantem que cada consumidor é visitado por exatamente um veículo. As restrições

(2.3) asseguram que se um veículo chega ao cliente i , ele deve sair deste cliente. As restrições (2.4) indicam que cada veículo inicia e finaliza a rota no mesmo depósito. As restrições (2.5) garantem que cada veículo pode partir apenas de um único depósito, ou seja, cada veículo será alocado a no máximo uma rota. As restrições (2.6) obrigam que o grafo seja conectado, como também asseguram a eliminação de subrotas. Essas restrições proíbem soluções que consistem em vários passeios desconectados. As restrições (2.7) impõem que os veículos não viajem do depósito i para o depósito j . As restrições (2.8) implicam na ordem das visitas aos nós, pois, se o veículo k viaja diretamente do nó i para o nó j , então o momento da chegada a_{kj} no nó j deve ser igual a $(b_{ki} + h_i + c_{ij})$. As restrições (2.9) garantem que o início do serviço no nó i atendido pelo veículo k seja atrasado ou igual ao instante de chegada. As restrições (2.10) garantem a ocorrência do atendimento ao consumidor i dentro da janela de tempo $[JTI_i, JTF_i]$. As restrições (2.11) garantem que a carga no veículo k não ultrapasse sua capacidade. As restrições (2.12) impõem que o tempo total na rota seja no máximo igual a MT . As restrições (2.13) definem as variáveis de decisão x como binárias.

2.2.3.2 Modelo Matemático Proposto para o PRVJTMD*

O modelo matemático para o PRVJTMD* proposto neste trabalho se faz com as proposições a seguir:

Nele, são introduzidas as variáveis:

$$y_k = \begin{cases} 1, & \text{se veículo } k \text{ é utilizado} \\ 0, & \text{caso contrário.} \end{cases}$$

Na função de minimização (2.1) é adicionada a parcela relativa à ponderação dos veículos, como se segue:

$$\min \beta \sum_{k \in \mathcal{K}} \sum_{i \in \mathcal{V}} \sum_{j \in \mathcal{V}} c_{ij} x_{kij} + \alpha \sum_{k \in \mathcal{K}} y_k \quad (2.14)$$

As restrições definidas na formulação do PRVJTMD, estabelecidas pelas equações (2.2)-(2.13), são mantidas para o PRVJTMD* e adicionadas as seguintes:

$$y_k \geq \frac{\sum_{i \in \mathcal{V}} \sum_{j \in \mathcal{V}} x_{kij}}{|\mathcal{N}|^2}, \quad \forall k \in \mathcal{K}, \quad (2.15)$$

$$y_{kij} \in \{0, 1\}, \quad \forall i, j \in \mathcal{V}, k \in \mathcal{K}. \quad (2.16)$$

A expressão (2.14) representa a função objetivo do problema, a ser minimizada. A função objetivo consiste na soma ponderada do custo de viagem e do número total de veículos envolvidos. Os fatores de ponderação α e β têm grande importância nas duas parcelas da função. As restrições (2.15) indicam quais veículos serão utilizados. As restrições (2.16) definem as variáveis de decisão y como binárias. É importante ressaltar que este modelo matemático do PRVJTMD* é uma das contribuições dessa tese.

2.3 Problemas de Roteamento Aberto: PRAVJTMD e PRAJTMD*

Esta seção apresenta o Problema de Roteamento Aberto de Veículos com Janelas de Tempo e Múltiplos Depósitos, em suas duas versões, denominadas PRAVJTMD e PRAVJTMD*. A Subseção 2.3.1 introduz estes dois problemas, apresentando exemplos ilustrativos que permitem o melhor entendimento a respeito de seus significados. A Subseção 2.3.2 apresenta a revisão bibliográfica a respeito dos problemas. A Subseção 2.3.3 introduz os modelos matemáticos dos dois problemas.

2.3.1 Descrição do PRAVJTMD e do PRAVJTMD*

O Problema de Roteamento de Veículos Aberto com Janelas de Tempo e Múltiplos Depósitos – PRAVJTMD (*Multi-Depot Open Vehicle Routing Problem with Time-Windows* – MDOVRPTW) é uma variante do PRVJTMD, em que a restrição que obriga o veículo a retornar para o depósito é retirada. Portanto, a descrição desses problemas inclui, no lugar dessa restrição, a seguinte cláusula:

- Os veículos não retornam para nenhum dos depósitos.

Nesta tese, aborda-se o PRAVJTMD e, além disso, introduz-se o Problema de Roteamento Aberto de Veículos com Janelas de Tempo e Múltiplos Depósitos visando a minimização do número total de veículos usados, aqui denominado como PRAVJTMD*. Assim como no PRVJTMD*, este problema consiste em construir uma solução que penaliza fortemente a utilização de um veículo.

As Figuras 2.3(a) e 2.3(b) ilustram exemplos de solução de uma instância para o PRAVJTMD e para o PRAVJTMD*, respectivamente. Nestes exemplos, a instância

utilizada possui 2 depósitos, que devem atender a 18 clientes. A frota total é composta por 8 veículos, cada um com uma capacidade $VC = 8$. O tempo em rota de cada veículo não pode exceder $MT = 20$. Cada arco tem um custo c_{ij} , dado pelo tempo de viagem entre os nós i e j . Uma tupla $(q_i, h_i, [e_i, l_i])$ está associado ao nó i , sendo, respectivamente, q_i a demanda, h_i o tempo de serviço, e $[e_i, l_i]$ a janela de tempo para o nó i .

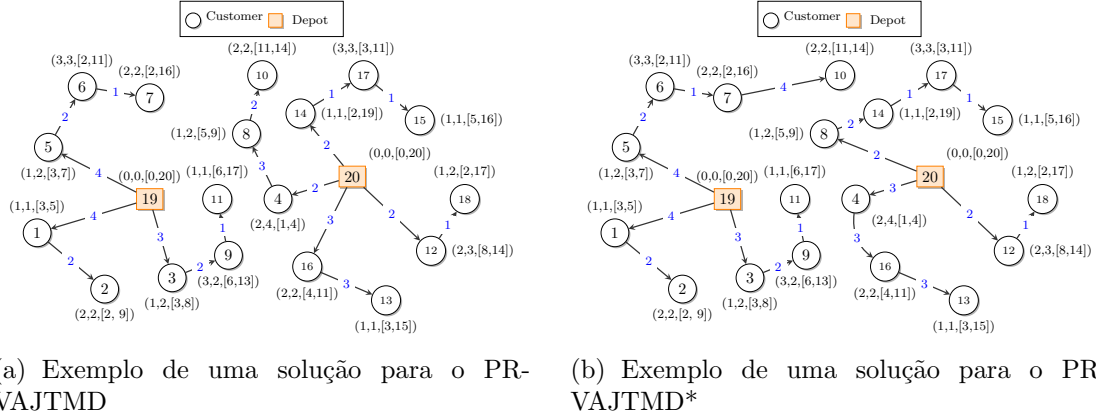


Figura 2.3: Exemplo de soluções para o PRVAJTMD e PRVAJTMD*.

A Tabela 2.4 mostra os dados para os exemplos apresentados. Para cada veículo k , a coluna “Rotas” mostra as rotas atribuídas à cada veículo; a coluna “Distância” mostra a distância percorrida pelo veículo k ; c_{ij} é o custo relativo a cada arco (i, j) ; h_i representa o tempo de serviço para cada cliente i ; a coluna WU_{ki} , o tempo de espera para descarregar o veículo k no cliente i ; a coluna T_k representa o tempo total gasto pelo veículo k na rota; Q_k representa a carga total transportada na rota k . Para compatibilizar a distância com o tempo gasto em cada arco, considerou-se uma unidade de tempo igual a uma unidade de distância.

Tabela 2.4: Dados para os exemplos do PRAVJTMD e do PRAVJTMD* mostrados na Figura 2.3.

Dados da Figura 2.3(a)								Dados da Figure 2.3(b)								
k	Rotas	Distância	c_{ij}	h_i	WU_k	T_k	Q_k	Rotas	Distância	c_{ij}	h_i	WU_k	T_k	Q_k		
1	(19, 1, 2)	6	(4, 2)	(0, 1, 2)	0	9	(0, 1, 2)	3	(19, 1, 2)	6	(4, 2)	(0, 1, 2)	0	9	(0, 1, 2)	3
2	(19, 5, 6, 7)	7	(4, 2, 1)	(0, 2, 3, 2)	0	14	(0, 1, 3, 2)	6	(19, 5, 6, 7, 10)	11	(4, 2, 1, 4)	(0, 2, 3, 2, 2)	0	20	(0, 1, 3, 2, 2)	8
3	(19, 3, 9, 11)	6	(3, 2, 1)	(0, 2, 2, 1)	0	11	(0, 1, 3, 1)	5	(19, 3, 9, 11)	6	(3, 2, 1)	(0, 2, 2, 1)	0	11	(0, 1, 3, 1)	5
4	(20, 4, 8, 10)	7	(2, 3, 2)	(0, 4, 2, 2)	0	15	(0, 2, 1, 2)	5	(20, 8, 14, 17, 15)	6	(2, 2, 1, 1)	(0, 2, 1, 3, 1)	3	16	(0, 1, 1, 3, 1)	6
5	(20, 14, 17, 15)	4	(2, 1, 1)	(0, 1, 3, 1)	0	9	(0, 1, 3, 1)	5	(20, 12, 18)	3	(2, 1)	(0, 3, 2)	0	8	(0, 2, 1)	3
6	(20, 12, 18)	3	(2, 1)	(0, 3, 2)	0	8	(0, 2, 1)	3	(20, 4, 16, 13)	9	(3, 3, 3)	(0, 4, 2, 1)	0	16	(0, 2, 1)	3
7	(20, 16, 13)	6	(3, 3)	(0, 2, 1)	0	9	(0, 2, 1)	3								
Total		39			0	75			41			3	75			

Na Figura 2.3(a), que mostra a solução do PRAVJTMD, 7 veículos foram utilizados, com distância total percorrida de 39. Na Figura 2.3(b), por outro lado, que mostra a solução via PRAVJTMD*, um veículo a menos foi utilizado, totalizando 6 veículos, com uma distância total de 41.

2.3.2 Revisão Bibliográfica sobre o PRAVJTMD e o PRAVJTMD*

O problema de roteamento aberto de veículos (PRVA) foi tratado pela primeira vez por [Schrage \(1981\)](#). Desde então, surgiram vários trabalhos com investigações sobre esses problemas. Na última década, os estudos relacionados ao tema têm despertado grande interesse de pesquisa. Esse interesse encontra-se no fato do surgimento de novas modalidades de entrega e prestação de serviço, em que há um local de origem onde os veículos partem, executam o trabalho e não retornam para o depósito. Esta é uma modalidade comum em empresas que subcontratam os serviços de entrega, por exemplo. Na Tabela 2.5 são mostrados alguns autores e a relação de métodos utilizados na resolução do PRV aberto.

Tabela 2.5: Problemas de Roteamento Aberto: revisão bibliográfica

Problema	Técnica	Ano	Autor
PRVA	Busca Tabu	2004	Brandão (2004)
	<i>Gravitational emulation local search algorithm</i>	2018	Rahmani Hosseinabadi et al. (2018)
PRVAJT	<i>Greedy look-ahead</i>	2007	Repoussis et al. (2007)
	VNS	2013	Redi et al. (2013)
	VNS com Aprendizado por Reforço	2019	Chen et al. (2019)
	<i>Multiple Neighborhood Search hybridized with a Tabu Search</i>	2017	Soto et al. (2017)
PRVAMD	<i>Hybrid adaptive large neighbourhood search algorithm</i>	2019	Lahyani et al. (2019)
	<i>Memory-based iterated local search algorithm</i>	2020	Brandão (2020)
	Programação Matemática	2021	Lalla-Ruiz e Mes (2021)
	<i>Hybrid genetic algorithm with adaptive local search</i>	2016	Li et al. (2016)
PRVAJTMD (variantes)	<i>Gravitational emulation local search algorithm</i>	2018	Shen et al. (2018)

Na tabela estão relacionados os problemas, os autores e os algoritmos propostos. Nessa relação de trabalhos, [Brandão \(2004\)](#) desenvolveu um algoritmo de Busca Tabu (BT) para resolver o PRVA. [Rahmani Hosseinabadi et al. \(2018\)](#) propuseram o algoritmo de pesquisa local de emulação gravitacional (*Gravitational Emulation Local Search* – GELS) para resolver o PRVA. Em ambos os estudos, o objectivo era reduzir o tempo de viagem, a distância e o número de veículos utilizados.

O Problema de Roteamento Aberto de Veículos com Janelas de Tempo (PRAVJT) foi proposto inicialmente em [Repoussis et al. \(2007\)](#). Os autores desenvolveram um método heurístico guloso para resolver o PRAVJT. [Redi et al. \(2013\)](#) apresentou um algoritmo de melhoria na seleção dos operadores que atuam nas vizinhanças pelo algoritmo VNS. No algoritmo VNS proposto, as rotas são criadas de maneira a garantir que os clientes com as primeiras janelas de atendimento sejam atendidos primeiramente. O algoritmo VNS proposto foi testado em 12 conjuntos de instâncias R1 e 8 conjuntos de instâncias RC1 das instâncias clássicas para o PRVJT introduzidas em [Solomon \(1987\)](#). [Chen et al. \(2019\)](#) resolveu um problema real de transporte de contêineres. Os autores desenvolveram uma versão do algoritmo VNS com reforço de aprendizagem (*Variable Neighbourhood Search algorithm with Reinforcement Learning* – VNS-RLS). O algoritmo foi aplicado em 15 instâncias referentes ao Porto

Ningbo na República Popular da China.

Tarantilis e Kiranoudis (2002) introduziu o Problema de Roteamento Aberto de Veículos com Múltiplos Depósitos (PRAVMD), ao estudar um problema real de distribuição de alimentos. Para a solução desse problema, os autores propuseram uma meta-heurística de busca local denominada *List-based threshold accepting* (LBTA). BTA). Soto et al. (2017) estudou o PRAVMD introduzindo o algoritmo *Multiple Neighborhood Search hybridized with a Tabu Search* (MNS-TS). Os autores introduzem técnicas de *Ejection Chain* para reduzir a dimensão e a quantidade de operadores utilizados na exploração das vizinhanças do problema. Lahyani et al. (2019) desenvolveram um algoritmo híbrido baseado na meta-heurística híbrida adaptativa, denominado *Hybrid Adaptive Large Neighborhood Search* (HALNS). Segundo os autores, o método combina o poder de um *Adaptive Large Neighborhood Search* (ALNS) para realizar uma pesquisa diversificada e completa do espaço de solução, com procedimentos de buscas locais que proporcionam melhorias que intensificam a pesquisa em torno de boas e promissoras soluções. Brandão (2020) apresentou o algoritmo *Memory-Based Iterated Local Search Algorithm* (MBILSA) para a solução de instâncias do PRAVMD. Os movimentos executados durante a busca local são validados e, a partir dos históricos de pesquisa, os movimentos são selecionados para serem aplicados nos procedimentos de perturbação. Lalla-Ruiz e Mes (2021), por sua vez, propuseram uma formulação matemática baseada em dois índices para o PRAVMD. Os autores avaliaram as contribuições de restrições alternativas para a conservação do fluxo e sub-rotas.

Uma variante do PRAVJTMD com compartilhamento de recursos entre depósitos foi abordada em Li et al. (2016). Os autores tratam o problema como uma combinação do PRVJTMD e PRVA. No problema apresentado, os veículos iniciam a entrega em um depósito A, mas ao atender o último cliente da rota, podem deslocar-se para o depósito mais próximo. Um algoritmo genético híbrido com fase de busca local adaptativa foi apresentado pelos autores e aplicado no problema discutido.

Shen et al. (2018) abordaram o PRAVJTMD considerando a restrição de redução de emissão de carbono na atmosfera. Os autores consideraram políticas comerciais de baixo teor de carbono. No modelo apresentado, os autores consideraram a redução na emissão de carbono e redução nos custos econômicos, que incluem o salário do condutor, custos de penalização, custos de combustível e custos do comércio de emissões de carbono. Para este problema, a solução foi obtida com um método híbrido envolvendo *Particle Swarm Optimization* (PSO) e técnica de Busca Tabu (BT)

2.3.3 Modelo Matemático Proposto para o PRVAJTMD e PRVAJTMD*

A formulação matemática para o PRVAJTMD segue conforme a expressão (2.1) com as restrições (2.2), (2.5) à (2.13), apresentadas na Seção 2.2.3 para o PRVJTMD, com a adição das restrições a seguir:

$$\sum_{j \in \mathcal{V}} x_{kji} = \sum_{j \in \mathcal{V}} x_{kij} + z_{ik} \quad \forall i \in \mathcal{C}, \forall k \in \mathcal{K}, \quad (2.17)$$

$$\sum_{j \in \mathcal{C}} x_{kdj} = \sum_{j \in \mathcal{C}} z_{jk}, \quad \forall k \in \mathcal{K} \quad (2.18)$$

$$\sum_{i \in \mathcal{C}} \sum_{d \in \mathcal{D}} x_{kid} = 0, \quad \forall k \in \mathcal{K}, \quad (2.19)$$

As restrições (2.17) garantem que se um veículo chega ao cliente i , este veículo deve sair deste cliente ou este cliente será o último da rota. As restrições (2.18) garantem que se um veículo começa a rota em um determinado depósito, exatamente um cliente será o último cliente da rota. As restrições (2.19) impõem que os veículos não viajem de nenhum cliente para um depósito.

A formulação matemática proposta para o PRVAJTMD* segue conforme proposta estabelecida para o PRVJTMD*, com a expressão (2.14) adicionada das restrições (2.2), (2.5) à (2.13), e (2.15) à (2.16), descritas na Seção 2.2.3, com as restrições (2.3) à (2.19).

2.4 Resumo do Capítulo

Este capítulo apresentou uma descrição da classe de problemas de roteamentos de veículos com janelas de tempo e múltiplos depósitos, nas versões com roteamento fechado e roteamento aberto. As características comuns entre os problemas foram relatadas e cada problema foi tratado separadamente. O problema de roteamento fechado foi abortado com a proposição de um modelo matemático envolvendo a incorporação da redução de veículos. No modelo proposto, foi adicionada uma parcela que penaliza a utilização dos veículos. Na sequência, dois problemas de roteamento aberto são caracterizados. No primeiro caso, foi abordada a redução da distância total das rotas; no segundo caso, a redução na quantidade de veículos. No decorrer do

capítulo, foi apresentada uma revisão bibliográfica, em que situou-se os problemas de roteamento de veículos abertos e fechados.

Capítulo 3

Referencial Teórico sobre *Variable Neighborhood Search*

Este Capítulo descreve a meta-heurística *Variable Neighborhood Search* (VNS) e sua estrutura básica. Inicialmente, a Seção 3.1 apresenta esta meta-heurística, destacando suas características mais relevantes e, dentre essas, os mecanismos para escapar de mínimos locais e as operações aplicadas na solução. Após a sua apresentação, a Seção 3.2 discute o critério de perturbação; a Seção 3.3 apresenta métodos para a seleção e mudança de vizinhanças; a Seção 3.4 mostra o procedimento de busca local adotado e, finalmente, as considerações finais do Capítulo são apresentadas na Seção 3.5.

3.1 *Variable Neighborhood Search*

Na solução de problemas de otimização, uma técnica de uso geral é a utilização de heurísticas que alterem uma determinada solução através do uso de suas vizinhanças.

Nessa abordagem, seja $\mathcal{S}$ o conjunto de soluções viáveis, com $s \in \mathcal{S}$, $\mathcal{N}$ o conjunto de vizinhanças que geram um solução factível e $f(s)$ o valor da função objetivo a ser minimizada. Uma solução s' é denominada como um vizinho de s . Esses vizinhos são obtidos a partir das modificações sofridas ao se aplicar a estrutura de vizinhança v , denotado por $\mathcal{N}_v(s)$, em que $\mathcal{N}_v \subseteq \mathcal{N}$ e $v = \{1, 2, \dots, v_{max}\} \mid v_{max} = |\mathcal{N}|$.

O algoritmo *Variable Neighborhood Search* (VNS), proposto por [Mladenović e Hansen \(1997\)](#), é uma meta-heurística que consiste em explorar de forma sistemática a combinação de vizinhanças no processo de solução de um problema de otimização. Este método apresenta uma estrutura flexível para a construção de heurísticas capazes de fornecer aproximações eficientes em problemas combinatórios e não-lineares, baseado em três propriedades:

- (i) um ótimo local para uma determinada vizinhança não é necessariamente um ótimo local para outra vizinhança;
- (ii) o ótimo global é também ótimo local para qualquer estrutura de vizinhança;
- (iii) evidências empíricas mostram que ótimos locais para diferentes vizinhanças encontram-se relativamente próximos (Hansen et al., 2017).

A primeira propriedade indica que o uso de movimentos mais complexos permite encontrar diferentes ótimos locais, de acordo com a estrutura de vizinhança utilizada. A segunda propriedade indica que o uso de várias vizinhanças se faz necessário quando os ótimos locais encontrados são de baixa qualidade. Finalmente, a terceira propriedade sugere maior exploração da solução incumbente pelas vizinhanças que atuam no processo. Através do distanciamento proporcionado pelas vizinhanças, é possível encontrar melhores soluções que a solução corrente e, assim, escapar dos ótimos locais. Essas propriedades podem ser observadas na Figura 3.1, em que representa-se geometricamente a exploração do espaço de soluções pelo VNS.

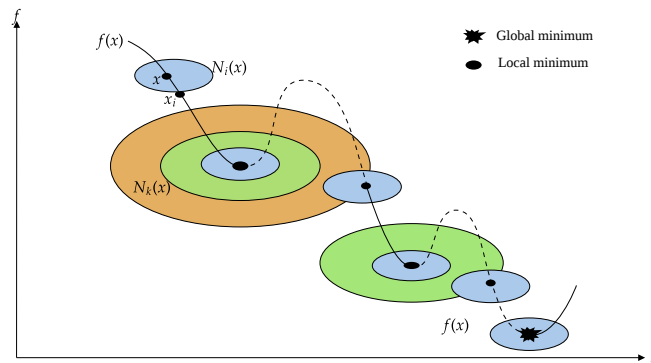


Figura 3.1: Exploração do espaço de busca pelo VNS. Fonte: Hansen et al. (2017)

Diferentemente de outras meta-heurísticas, o VNS não segue uma trajetória no espaço de soluções. Por este método, as vizinhanças gradativamente mais “distantes” da solução corrente são exploradas, focando-se no refinamento em torno de uma nova solução gerada pela vizinhança aplicada, através de um procedimento de busca local. Outro recurso fundamental no VNS é o processo de perturbação. Nessa fase, a solução sofre mudanças através do conjunto de vizinhanças definidas, visando proporcionar diversificação suficiente para não ficar preso em ótimos locais. Desta forma, a fase de refinamento e o procedimento de perturbação, ambos associados à etapa de mudança de vizinhança, são executados alternativamente e, quando não é possível obter nenhuma outra solução de melhora, o método é interrompido e a solução geral encontrada durante todo o processo é devolvida como resposta. Assim, os três critérios a seguir devem ser atendidos de forma recorrente, até que alguma condição de parada seja alcançada:

- (i) Executar a perturbação;
- (ii) Aplicar a busca local;
- (iii) Efetuar as mudanças de vizinhanças.

A meta-heurística VNS, em sua forma básica, descrita por [Mladenović e Hansen \(1997\)](#) e [Hansen et al. \(2017\)](#), é apresentada no Algoritmo 1. Nesse modelo, $\mathcal{N} = \{\mathcal{N}_1, \mathcal{N}_2, \dots, \mathcal{N}_{v_{\max}}\}$ representa um conjunto de estruturas de vizinhanças que operam sobre a solução s . Em cada estrutura de vizinhança $\mathcal{N}_v$, $1 \leq v \leq v_{\max}$, uma solução s' é mapeada, representada por $s' \leftarrow \mathcal{N}_v(s)$. No processo de perturbação, descrito na linha 4, um vizinho s' , pertencente à vizinhança $\mathcal{N}_v$, é retornado. O processo de busca local, representado na linha 5, retorna um ótimo local s'' . A mudança de vizinhança é indicada na linha 6. Como critério de parada, pode ser utilizado o tempo de processamento máximo, o número máximo de iterações, o número máximo de iterações sem melhora, combinações destes ou algum outro critério que seja aderente às características do problema tratado.

Algoritmo 1 VNS Básico ($s, v_{\max}, t_{\max}$)

```

1: while (critério de parada não atendido) do
2:    $v \leftarrow 1$ ; {nível inicial de perturbação}
3:   while  $v \leq v_{\max}$  do
4:      $s' \leftarrow Perturbação(s, v)$ ;
5:      $s'' \leftarrow BuscaLocal(s')$ ;
6:      $MudançaVizinhança(s, s'', v)$  {mudança de vizinhança}
7:   end while
8: end while
9: return  $s$ ;

```

O conjunto de estruturas de vizinhanças adotadas durante a perturbação não são necessariamente as mesmas adotadas durante a busca local. Essa observação permite a adoção de estruturas de vizinhança geradas por operadores especializados na diversificação da solução e outros especializados no seu refinamento.

O VNS combina elementos determinísticos e estocásticos no processo de escolha das vizinhanças. Caso os processos de perturbação e busca local sejam determinísticos, após uma certa quantidade de execuções, os vizinhos s'' de s' convergem para o mesmo resultado e ciclos invariavelmente ocorrem, não promovendo a diversificação necessária para escapar dos ótimos locais.

3.2 Procedimento de Perturbação

As perturbações são movimentos de caráter aleatório, que modificam a solução atual, conduzindo-a para outra região do espaço de busca. Segundo [Rego e Souza](#)

(2019), elas devem ocorrer de maneira gradativa, dada pela mudança de vizinhança e pela quantidade de vezes que a estrutura é aplicada à solução. Devido às diferentes características das instâncias dos problemas tratados, as diversificações provocadas pela v -ésima vizinhança podem não permitir que a solução alcançada escape do mínimo local, ou podem conduzir a solução s para uma outra bacia de atração, mais ou menos distante do ótimo global.

Um procedimento simples proposto por Hansen et al. (2017) consiste em gerar uma solução de forma aleatória a partir da v -ésima estrutura de vizinhança. Este procedimento, apresentado no Algoritmo 2, permite maior liberdade de implementação em diversos problemas de otimização.

Algoritmo 2 perturbacaoBasica ($s, v, \mathcal{N}$)

```

1: Gerar aleatoriamente um vizinho  $s' \in \mathcal{N}_v(s)$ ;
2: return  $s'$ ;

```

3.3 Mudança de vizinhanças

O propósito da mudança de vizinhanças é guiar a meta-heurística VNS durante a exploração do espaço de busca. Para cada solução s' gerada por uma estrutura de vizinhança $\mathcal{N}_v(s)$, na forma $s' \leftarrow \mathcal{N}_v(s)$, uma outra estrutura de vizinhança deve ser aplicada para explorar outras regiões no espaço de busca. Segundo Hansen et al. (2017), existem quatro procedimentos amplamente utilizados para a mudança de vizinhanças, descritos a seguir:

(i) Mudança Sequencial de Vizinhanças

O Algoritmo 3 apresenta o procedimento de mudança sequencial de vizinhanças. Nesse procedimento, se uma solução é melhorada em relação à solução incumbente em alguma das estruturas de vizinhanças, a busca retorna para a primeira estrutura, de acordo com a sequência pré-estabelecida em $\mathcal{N}$; caso, no entanto, não haja melhoria, as vizinhanças são aplicadas sequencialmente até o limite $|\mathcal{N}|$.

Algoritmo 3 vizinhancaSequencial(s, s', v)

```

1: if  $f(s') < f(s)$  then
2:    $s \leftarrow s'$ ;
3:    $v \leftarrow 1$ ;
4: else
5:    $v \leftarrow v + 1$ ;
6: end if

```

(ii) **Mudança Cíclica de Vizinhanças**

Neste procedimento, independentemente de ter-se verificado a melhoria da solução, quando um vizinho é gerado, ocorre a mudança para a próxima estrutura de vizinhança, conforme definido em $\mathcal{N}$, até o limite $|\mathcal{N}|$, conforme mostra o Algoritmo 4.

Algoritmo 4 vizinhançaCiclica (s, s', v)

```

1:  $v \leftarrow v + 1$ ;
2: if  $f(s') < f(s)$  then
3:    $s \leftarrow s'$ ;
4: end if

```

(iii) **Mudança *Pipe* de Vizinhanças**

Neste procedimento, conforme o Algoritmo 5, caso haja melhoria da solução em alguma vizinhança, não há mudança de estrutura, pois a proposta é explorá-la, intensificando a procura na região.

Algoritmo 5 vizinhançaPipe (s, s', v)

```

1: if  $f(s') < f(s)$  then
2:    $s \leftarrow s'$ ;
3: else
4:    $v \leftarrow v + 1$ ;
5: end if

```

(iv) **Mudança Assimétrica de Vizinhanças**

Ao se aplicar o procedimento representado pelo Algoritmo 6, o objetivo é permitir a exploração de vales distantes da solução incumbente.

A aceitação da solução depende de outros fatores além de sua melhoria, sendo avaliada levando-se em consideração não apenas os valores da função objetivo e da solução corrente, mas, também, a distância entre elas. Em (Hansen et al., 2017), a seguinte função é apresentada, quando se deseja minimizar:

$$g(x,y) = f(x) - f(y) - \alpha d(x,y) \quad (3.1)$$

em que α representa um parâmetro positivo, enquanto $d(x,y)$ refere-se à distância entre as soluções x e y . O processo de mudança, apresentado no Algoritmo 6, segue como uma adaptação da mudança sequencial de vizinhança apresentada no Algoritmo 3, porém, com o critério de escolha conforme a Equação (3.1). Os procedimentos Mudança Cíclica de Vizinhanças e Mudança *Pipe* de Vizinhanças também podem ser incorporados, desde que se considere a avaliação imposta pelo procedimento de Mudança Assimétrica de Vizinhanças.

Algoritmo 6 vizinhancaAssimetrica (s, s', v, θ)

```

1: if  $f(s') - f(s) < \alpha d(s', s)$  then
2:    $s \leftarrow s'$ ;
3:    $v \leftarrow 1$ ;
4: else
5:    $v \leftarrow v + 1$ ;
6: end if

```

3.4 Procedimento de Busca Local

No procedimento de busca local, uma solução que minimize, ou maximize, a função objetivo $f(s)$ é gerada. Nesta fase, o refinamento da solução acontece, pois, durante a busca, procura-se encontrar melhores resultados para o problema tratado. A meta-heurística VNS depende fortemente desse procedimento para seu sucesso, que ocorre pela exploração da solução atual dada $\mathcal{N}_v(s)$ e termina quando uma solução s fornece o mínimo local em relação ao conjunto de vizinhanças exploradas. Neste processo, uma solução s' é gerada a partir da solução s . Essa solução é gerada através da estrutura de vizinhança $\mathcal{N}_v(s)$, no qual explora os vizinhos à solução s . Caso o novo vizinho s' seja melhor que s , ele passa a ser a nova solução incumbente.

Quanto aos critérios de melhora, as estratégias mais comumente utilizadas durante o procedimento de busca local em uma vizinhança $\mathcal{N}_v(s)$ ocorrem a partir das estratégias Primeira de Melhora (*First Improvement*) ou Melhor Encontrado (*Best Improvement*). No primeiro caso, a atualização da referência é efetivada assim que uma melhoria é encontrada na solução corrente; no segundo caso, a atualização ocorre somente após a comparação entre todos os vizinhos pertencentes à $\mathcal{N}_v$. Vários métodos de busca local podem ser incorporados ao VNS. Nas subseções 3.4.1, 3.4.2 e 4.7.1 são apresentadas três heurísticas de refinamento da solução que podem ser associadas ao VNS.

3.4.1 *Variable Neighborhood Descent*

A heurística *Variable Neighborhood Descent* (VND) é um método de busca local, conforme Mladenović e Hansen (1997); Hansen et al. (2010), que usa uma ordem pré-definida de vizinhanças para explorar o espaço de soluções. Segundo os autores, é mais provável que o ótimo local gerado a partir de um conjunto com várias estruturas de vizinhança seja um ótimo global que a solução gerada por uma única estrutura de vizinhança. Mais precisamente, o VND explora várias estruturas de vizinhança, de maneira sequencial ou aninhada, como estratégia de melhoria para uma determinada solução. A mesma sequência ocorre todas as vezes que a busca local é aplicada e, além disso, pode-se usar tanto *First Improvement* quanto *Best Improvement* como

estratégia de atualização.

Na condução do VND básico, várias estruturas de vizinhança são ordenadas em uma lista sequencialmente estabelecida pelo conjunto de vizinhanças $\mathcal{N} = \{\mathcal{N}_1, \mathcal{N}_2, \dots, \mathcal{N}_{v_{\max}}\}$, em que, para $\mathcal{N}_v$, $1 \leq v \leq v_{\max}$, os vizinhos da solução corrente s são avaliados retornando o vizinho $s' \leftarrow \mathcal{N}_v(s)$.

No Algoritmo 7 são apresentados os passos para execução do VND. Neste algoritmo, a estratégia *Best Improvement* é utilizada no retorno do vizinho (linha 3). Na linha 4, a solução corrente pode ser atualizada pelo procedimento “vizinhançaSequencial”. Havendo uma melhoria na solução, o procedimento retoma a busca, a partir da primeira estrutura de vizinhança definida em $\mathcal{N}$. Caso seja utilizada a estratégia *First Improvement*, a função descrita na linha 3 deve ser alterada para retornar o primeiro vizinho s' na vizinhança $\mathcal{N}_v(s)$ tal que $f(s') < f(s)$.

Outros procedimentos de mudanças de vizinhanças, como Mudança Cíclica de Vizinhanças e Mudança *Pipe* de Vizinhanças, são apresentadas em Hansen et al. (2017) como alternativas à Mudança Sequencial das estruturas de vizinhança.

Algoritmo 7 VND($s, \mathcal{N}$)

```

1:  $v \leftarrow 1$ ;
2: while  $v \leq |\mathcal{N}'|$  do
3:    $s' \leftarrow \arg \min_{z \in \mathcal{N}'_v(s)} f(z)$ ; {Melhor vizinho de  $\mathcal{N}'_v(s)$ }
4:   vizinhançaSequencial ( $s, s', v$ );
5: end while
6: return  $s$ ;

```

3.4.2 Randomized Variable Neighborhood Descent

O procedimento de busca local *Randomized Variable Neighborhood Descent* (RVND), apresentada no Algoritmo 8, foi proposta por (Souza et al., 2010) como uma variação do VND, em que se substitui a ordem pré-definida de vizinhanças para explorar o espaço de soluções por uma ordem aleatória a cada chamada. Segundo Subramanian et al. (2013a), esta abordagem produz, em média, resultados melhores que a versão com ordem determinística das vizinhanças e tem a vantagem de não necessitar de um estudo prévio da melhor ordem de vizinhanças, uma vez que a sequência das vizinhanças é gerada de forma aleatória e dificilmente uma mesma sequência será repetida. O método é visto como uma boa alternativa para o VND, uma vez que definir a melhor sequência de estruturas de vizinhanças é um problema com alto grau de dificuldade.

No Algoritmo 8 é mostrado o procedimento de busca local definido no procedimento RVND. No início do algoritmo, na linha 1, $\mathcal{N}'$ representa o conjunto de

vizinhanças gerado aleatoriamente a partir de $\mathcal{N}$. O restante do algoritmo é idêntico ao VND apresentado no Algoritmo 7 na Seção 3.4.1.

Algoritmo 8 RVND($s, \mathcal{N}$)

```

1:  $\mathcal{N}' \leftarrow \text{rand}(\mathcal{N});$ 
2:  $v \leftarrow 1;$ 
3: while  $v \leq |\mathcal{N}'|$  do
4:    $s' \leftarrow \arg \min_{z \in \mathcal{N}'_v(s)} f(z); \{\text{Melhor vizinho em } \mathcal{N}'_v(s)\}$ 
5:   VizinhancaSequencial ( $s, s', v$ );
6: end while
7: return  $s;$ 

```

3.5 Resumo do Capítulo

Neste Capítulo é apresentado o algoritmo VNS básico, com seus três critérios de funcionamento: perturbação, mudança de vizinhanças e busca local. Nas etapas de perturbação e busca local, as estruturas de vizinhança são exploradas com finalidades específicas. Na perturbação, o intuito é a diversificação, enquanto que na busca local, deseja-se refinar a solução perturbada. Esses critérios têm uma estrutura relativamente simples, quando comparada a outros métodos mais complexos, sendo aplicados nos mais diversos problemas de otimização.

O VNS é um método que possui poucos parâmetros a serem informados, o que facilita seu entendimento e faz com que o método seja muito estudado e aplicado nos mais diversos problemas relativos ao PRV, mais especificamente nos PRVJTMDs. No Capítulo 4 são apresentadas as propostas para a escolha das estruturas de vizinhança a serem geradas nas fases de perturbação e busca local.

Capítulo 4

Adaptações da Meta-heurística VNS Aplicadas aos PRVJTMDs Apresentados

Neste capítulo é introduzida uma proposta de adaptação da meta-heurística *Variable Neighborhood Search* (VNS) para a solução de instâncias dos problemas de roteamento de veículos definidos na Seção 1.2. A adaptação do VNS proposta nesta pesquisa mantém as premissas básicas do VNS clássico. As principais alterações relacionam-se ao critério de parada, ao processo de perturbação e às mudanças nas estruturas de vizinhanças. Inicialmente, a representação computacional para a geração das rotas é apresentada e discutida na Seção 4.1. Na sequência, as estruturas de vizinhança utilizadas no processo de perturbação e busca local são apresentadas, discutidas e exemplificadas na Seção 4.2. Na Seção 4.3, é descrito o método de geração da população inicial. Na Seção 4.4, os procedimentos de perturbação propostos são apresentados. As adaptações e proposições de buscas locais são apresentadas e discutidas na Seção 4.5. As funções de avaliação utilizadas pelos algoritmos nos problemas que envolvem a redução na quantidade de veículos e distância percorrida são apresentadas na Seção 4.6. Finalmente, as considerações finais sobre as adaptações apresentadas são discutidas na Seção 4.8.

4.1 Representação da Solução

A representação da solução s utilizada nesta tese é dada por um conjunto de listas representadas por:

$$\mathcal{R} = \left\{ r_1^1, r_2^1, \dots, r_m^1, r_1^2, r_2^2, \dots, r_n^2, r_1^{|\mathcal{D}|}, r_2^{|\mathcal{D}|}, \dots, r_t^{|\mathcal{D}|} \right\} \quad (4.1)$$

Neste conjunto, cada elemento r_i^j representa uma rota i associada ao depósito j , de modo que $r^1, r^2, \dots, r^j, \dots, r^{|\mathcal{D}|}$ representam o conjunto de rotas de cada depósito j , para $j = 1, \dots, |\mathcal{D}|$ e $|\mathcal{D}| \geq 2$. Além disso, cada depósito j possui uma quantidade própria m_i de rotas, para $m_i \geq 1$.

Na Figura 4.1, o depósito 14 está associado às rotas $r_1^{14} = \{14, 4, 2, 14\}$, $r_2^{14} = \{14, 12, 3, 14\}$ e $r_3^{14} = \{14, 5, 9, 14\}$; o depósito 15 está associado às rotas $r_1^{15} = \{15, 1, 6, 15\}$, $r_2^{15} = \{15, 11, 13, 15\}$ e $r_3^{15} = \{15, 7, 8, 10, 15\}$, com cada veículo v_i^j associado ao depósito j e rota i . Nessa solução, o veículo $veic_1^{14}$ parte do depósito 14, atende aos consumidores de r_1^{14} e retorna ao depósito. O mesmo ocorre para $veic_2^{14}$, que atende r_2^{14} e para $veic_3^{14}$, que atende à rota r_3^{14} . Situação similar acontece para as rotas associadas ao depósito 15, com os veículos $veic_1^{15}$, $veic_2^{15}$ e $veic_3^{15}$, que atendem às rotas r_1^{15} , r_2^{15} e r_3^{15} , respectivamente.

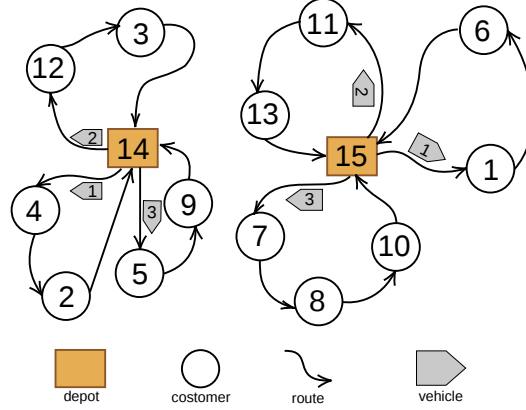


Figura 4.1: Exemplo de uma solução com dois depósitos.

No exemplo, o conjunto $\mathcal{R}$ é composto da seguinte maneira:

$$\mathcal{R} = \left\{ (14, 4, 2, 14), (14, 12, 3, 14), (14, 5, 9, 14), \right. \\ \left. (15, 1, 6, 15), (15, 11, 13, 15), (15, 7, 8, 10, 15) \right\}$$

Neste trabalho, a representação das sub-rotas pertencentes a um depósito é armazenada em uma lista. A solução completa do problema é dada pelo conjunto de vetores associados aos depósitos.

4.2 Estruturas de Vizinhaça Propostas

Existem diferentes vizinhanças amplamente discutidas na literatura para explorar o espaço de soluções dos problemas de otimização. Segundo Vidal et al. (2013a), a troca de arcos e/ou a movimentação dos vértices do grafo $\mathcal{G} = (\mathcal{V}, \mathcal{A})$, quando

utilizadas na busca local aplicada ao PRVJTMD, altera a sequência de visitas aos clientes. Essas alterações possibilitam a melhoria simultânea de várias rotas e, assim, a combinação das operações realizadas nas vizinhanças pode promover uma melhoria significativa na redução do custo final da solução.

Para a exploração de uma vizinhança, movimentos são aplicados na solução através de operadores específicos. Entre os operadores mais utilizados para a exploração das vizinhanças, encontram-se os de transferência, permutação e reversão. Movimentos de inserção, nomeados por [Osman \(1993\)](#) como *Shift*, consistem em transferir um ou mais clientes de uma rota para outra. Nos movimentos de troca ou intercâmbio, denominados em inglês como *Swap*, permutam-se um ou mais vértices da solução. O movimento *2-Opt* é um método baseado na exclusão e reinserção de dois arcos não consecutivos na mesma rota e, ao se manipular rotas diferentes, uma alternativa possível é dada pela utilização da vizinhança *2-Opt**, proposta por [Potvin e Rousseau \(1995\)](#). *A priori*, movimentos de troca e inserção não impõem grande alteração na sequência inicial apresentada, gerando vizinhos mais próximos; já movimentos com características de cruzamentos e/ou inversão alteram sobremaneira as características da solução, ocasionando um maior distanciamento da solução corrente.

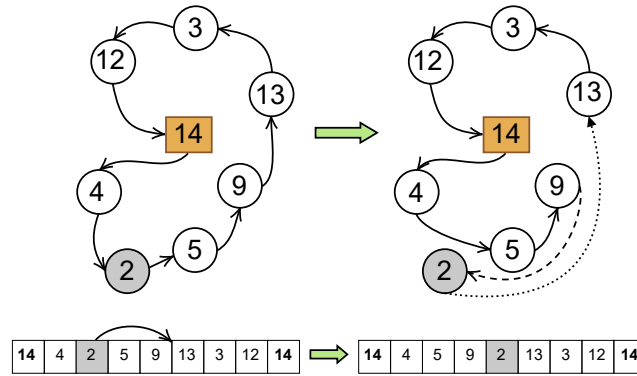
Nos problemas com vários depósitos, os movimentos acarretados por esses operadores podem ocorrer na mesma rota (intrarota), entre diferentes rotas (inter-rotas) ou entre diferentes depósitos (inter-depósitos). Tais métodos são amplamente descritos na literatura e apresentados também por [Subramanian et al. \(2010, 2013a\)](#); [Vidal et al. \(2012, 2013a, 2014\)](#) e [Bezerra et al. \(2018\)](#).

As Seções 4.2.1, 4.2.2 e 4.2.3 apresentam cada um dos operadores utilizados nesta tese.

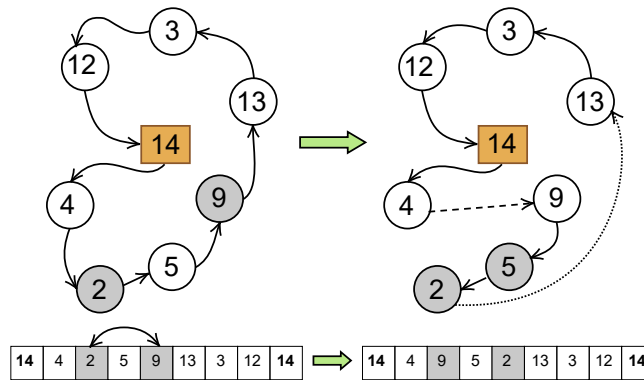
4.2.1 Operadores Intrarrotas

Esta seção apresenta os operadores utilizados na geração das estruturas de vizinhanças através de movimentos que ocorrem em uma mesma rota, denominados movimentos intrarrotas. Neste trabalho, para modificações de uma mesma rota, um conjunto com cinco diferentes operadores compõem o conjunto $\mathcal{N}$. Os operadores *Reinsertion*, *Swap*, *Or-opt*, *2-opt*, *3-opt* e *Reverse* definem os movimentos que geram as estruturas de vizinhanças intrarrotas utilizadas. Eles são descritos como se segue:

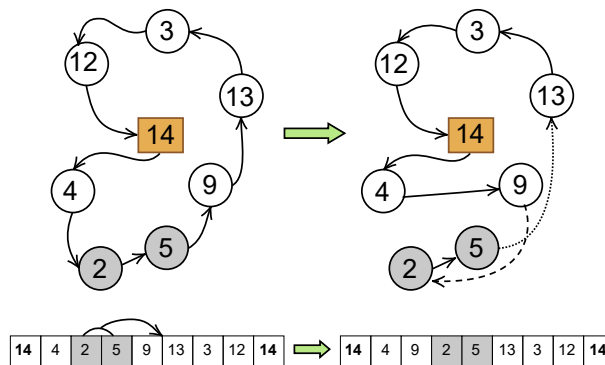
- (i) *Reinsertion*: um cliente é removido e reinserido em outra posição da rota. Na Figura 4.2, o cliente 2 é realocado para a posição entre os clientes 9 e 13.

Figura 4.2: Exemplo de vizinhança *Reinsertion*.

- (ii) *Swap*: consiste em permutar dois clientes de uma mesma rota. Na Figura 4.3, os clientes 2 e 9 são trocados, formando a nova rota.

Figura 4.3: Exemplo de vizinhança *Swap*.

- (iii) *Or-opt2*: dois clientes consecutivos são removidos e reinseridos em outra posição da rota. Na Figura 4.4, os clientes adjacentes 2 e 5 são realocados para a posição entre os clientes 9 e 13, formando uma nova rota.

Figura 4.4: Exemplo de vizinhança *Or-opt2*.

- (iv) *2-Opt*: dois arcos não adjacentes são excluídos e outros dois são adicionados para que uma nova rota seja gerada. Na configuração com apenas uma única rota apresentada na Figura 4.5, os arcos (4, 2) e (9, 12) são excluídos, enquanto os arcos (4, 9) e (2, 12) são inseridos, alterando a rota para (14, 4, 9, 5, 2, 12, 3, 14).

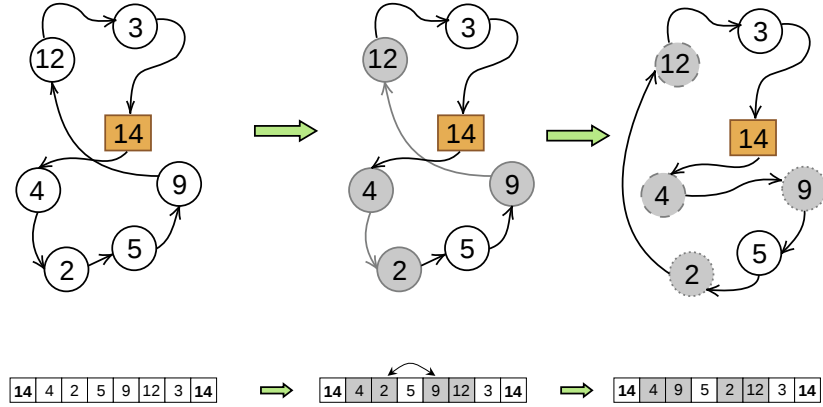


Figura 4.5: Exemplo de vizinhança *2-Opt*.

- (v) *3-Opt*: três arcos são excluídos e todas as possibilidades de troca entre eles são testadas na geração da nova rota. Na Figura 4.6, os arcos (4, 2), (5, 9) e (12, 3) são excluídos, enquanto os arcos (4, 9), (5, 12) e (3, 2) são inseridos, alterando a rota para (14, 4, 9, 5, 12, 3, 2, 14).

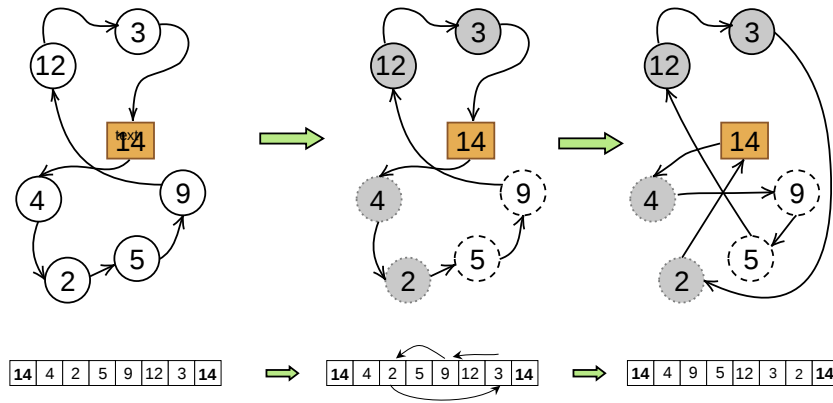


Figura 4.6: Exemplo de vizinhança *3-Opt*.

- (vi) *Reverse*: este movimento, representado na Figura 4.7, inverte a direção da rota.

4.2.2 Operadores Inter-Rotas

Nesta tese, um conjunto com oito diferentes tipos de movimentos é utilizado para fazer modificações entre clientes de rotas distintas pertencentes ao mesmo depósito.

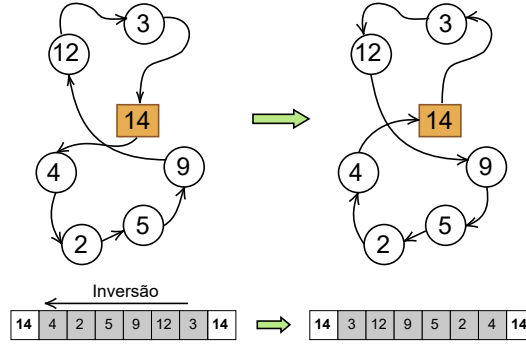


Figura 4.7: Exemplos de reversão.

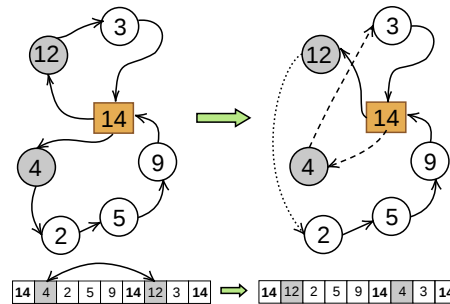
Esses movimentos definem as estruturas de vizinhanças inter-rotas.

Dentre as oito estruturas definidas, quatro delas são baseadas em movimentos λ -interchanges (Osman, 1993), que consistem na troca de até λ clientes consecutivos entre duas rotas. Para minimizar o esforço computacional, λ foi limitado a 2. De acordo com Cordeau et al. (2005), essas trocas são melhores explicadas como duplas (λ_1, λ_2) (com $\lambda_1 \leq \lambda$ e $\lambda_2 \leq \lambda$), sendo que os λ_1 clientes são transferidos da rota r_k para a rota r_l e os λ_2 clientes da rota r_l para a rota r_k .

Portanto, combinando os elementos e desconsiderando as simetrias, são possíveis as seguintes combinações de movimentos 2-interchanges: $((1, 0), (1, 1), (2, 0), (2, 1), (2, 2))$. Observe que tais combinações incluem movimentos de realocação $((1, 0), (2, 0))$ e de troca $((1, 1), (2, 1), (2, 2))$. Dois operadores que manipulam os movimentos baseiam-se em realocação (*shift*). Outros dois são adaptações para o movimento 2-Opt quando aplicado em rotas diferentes. A última estrutura de vizinhança a ser definida baseia-se na operação que consiste em trocar dois segmentos de diferentes rotas, denominada *Cross-exchange* (Taillard et al., 1997).

A descrição dos movimentos inter-rotas é apresentada a seguir:

- (i) *Swap(1,1)*: permutação entre um cliente v_j de uma rota r_k e um cliente v_t em outra rota r_l . Na Figura 4.8 é mostrado um exemplo em que os clientes 1 e 13 são permutados.

Figura 4.8: Exemplo de *Swap(1,1)*.

- (ii) $Swap(2, 1)$: permutação de dois clientes adjacentes v_j e v_{j+1} de uma rota r_k por um cliente v_t em outra rota r_l . Na Figura 4.9, os clientes adjacentes 4 e 2 foram trocados com o cliente 12.

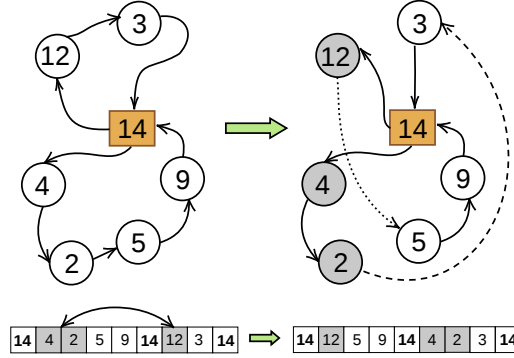


Figura 4.9: Exemplo de vizinhança $Swap(2, 1)$.

- (iii) $Swap(2, 2)$: permutação entre dois clientes adjacentes (v_j, v_{j+1}) de uma rota r_k e outros dois clientes adjacentes (v_t, v_{t+1}) pertencentes a outra rota r_l . Na Figura 4.10, os clientes adjacentes 5 e 9 foram trocados com os clientes 12 e 3.

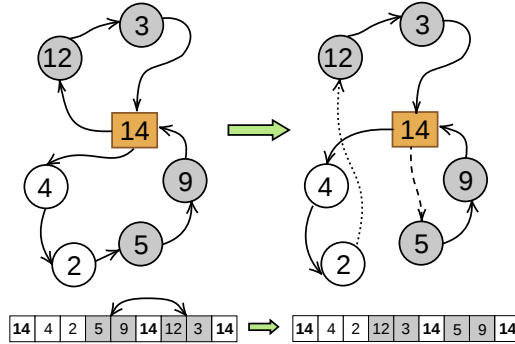
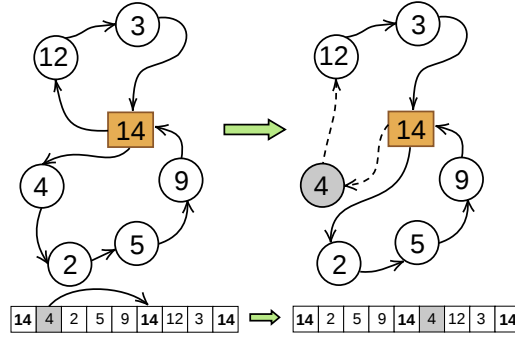
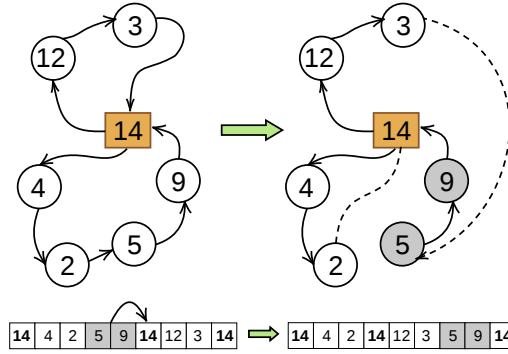


Figura 4.10: Exemplo de vizinhança $Swap(2, 2)$.

- (iv) $Shift(1, 0)$: transferência de um cliente v_j de uma rota r_k para outra posição, ou para uma outra rota r_l . Na Figura 4.11, o cliente 4 foi transferido para a outra rota.

Figura 4.11: Exemplo de vizinhança $Shift(1,0)$.

- (v) $Shift(2, 0)$: transferência de dois clientes adjacentes v_j e v_{j+1} de uma rota r_k para outra posição não adjacente em r_k ou outra rota r_l . Na Figura 4.12, os clientes adjacentes 5 e 9 foram transferidos para outra rota.

Figura 4.12: Exemplo de vizinhança $Shift(2,0)$.

A seguir serão apresentados os operadores $2-Opt^*$ Alternativa 1, $2-Opt^*$ Alternativa 2 e $Cross$. Para um melhor entendimento desses operadores, optamos por outra configuração de rotas associadas ao depósito 15.

As vizinhanças geradas pelo operador $2-Opt^*$ são baseadas na exclusão e reinserção de dois pares de arcos de duas rotas diferentes. Existem duas alternativas para este operador que são apresentadas nos itens (i) e (ii). Nos dois casos, há duas rotas distintas r_1 e r_2 , em que o arco (u, x) pertence à rota r_1 e o arco (v, y) pertence à rota r_2 .

(vi) *2-Opt* Alternativa 1*: substitui-se (u, x) e (v, y) por (u, v) e (x, y) . .

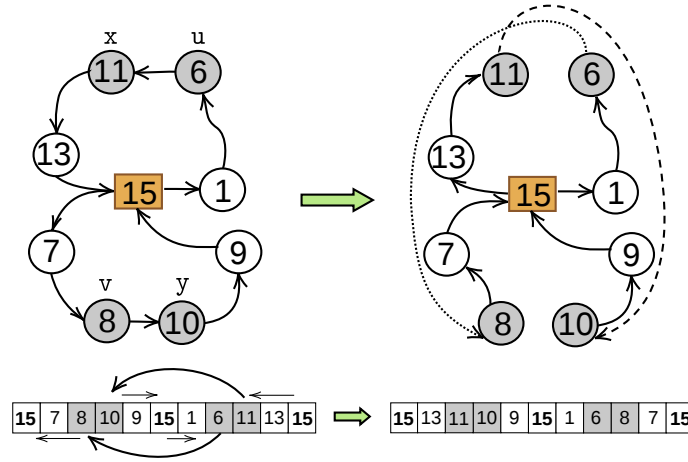


Figura 4.13: Exemplos de vizinhaça *2-Opt* Alternativa 1*

Na Figura 4.13, os arcos $(u, x) = (6, 11)$ e $(v, y) = (8, 10)$ são substituídos pelos arcos $(u, v) = (6, 8)$ e $(x, y) = (11, 10)$. Observe que há inversão parcial das duas novas rotas geradas.

(vii) *2-Opt* Alternativa 2*: substitui-se (u, x) e (v, y) por (u, y) e (x, v) . .

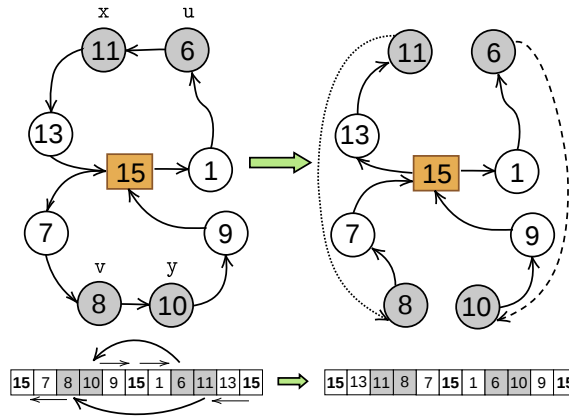


Figura 4.14: Exemplos de vizinhaça *2-Opt* Alternativa 2*.

Na Figura 4.14, os arcos $(u, x) = (6, 11)$ e $(v, y) = (8, 10)$ são substituídos pelos arcos $(u, y) = (6, 10)$ e $(x, v) = (11, 8)$. Neste caso, uma das rotas geradas é totalmente invertida.

- (viii) *Cross*: partes de duas rotas são permutadas. Assemelha-se ao operador *2-Opt** Alternativa 2, em que são substituídos os arcos (u, x) e (v, y) pelos arcos (u, y) e (v, x) .

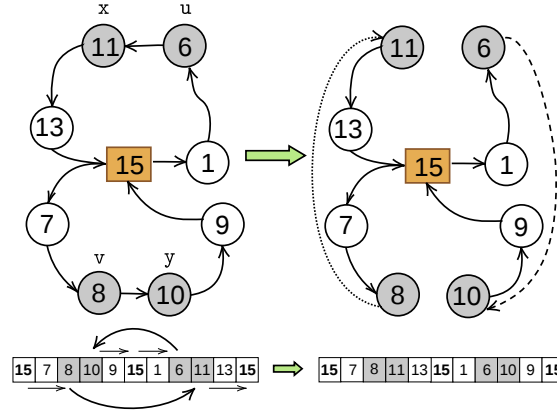


Figura 4.15: Exemplo de *Cross*.

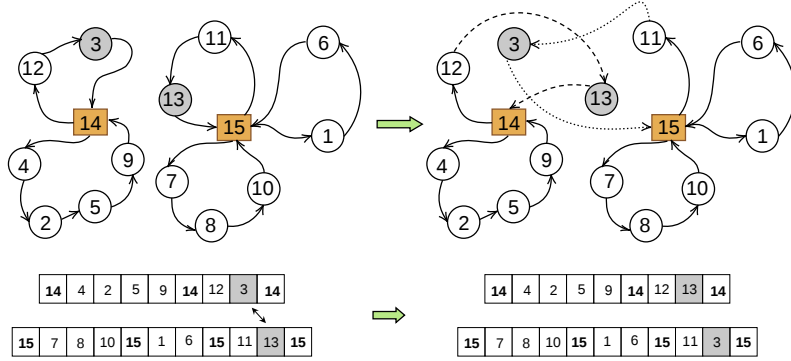
Na Figura 4.15, os arcos $(u, x) = (6, 11)$ e $(v, y) = (8, 10)$ são substituídos pelos arcos $(u, y) = (6, 10)$ e $(v, x) = (8, 11)$. Neste caso, não há inversão no sentido em nenhuma das rotas geradas.

Tanto o movimento *2-Opt** quanto o movimento *Cross* apresentam comportamentos parecidos em relação à troca dos elementos em diferentes rotas; porém, no movimento *Cross*, ocorre a troca entre partes das rotas selecionadas sem a inversão dos segmentos.

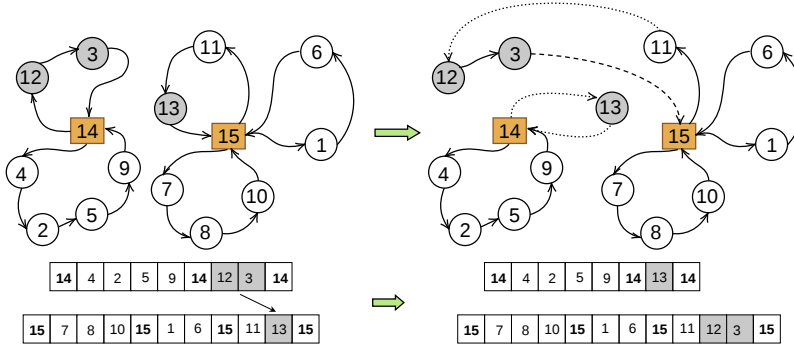
4.2.3 Operadores Inter-Depósitos

As estruturas de vizinhaça inter-depósitos propostas neste trabalho são geradas por movimentos de permutação e realocação que ocorrem em rotas pertencentes a diferentes depósitos. Neste trabalho, seis operadores são propostos para a geração destas estruturas de vizinhaça. São eles:

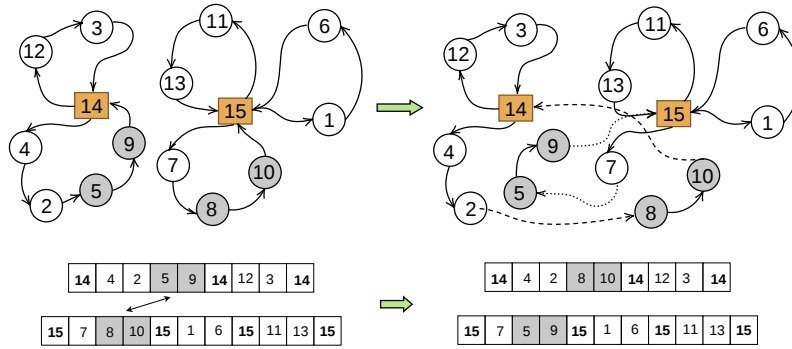
- (i) *Swap(1,1)-InterDepot*: consiste em permutar um cliente v_j de uma rota r_k com um cliente v_t de outra rota r_l pertencente a outro depósito.

Figura 4.16: Exemplo de $Swap(1,1)$ -InterDepot.

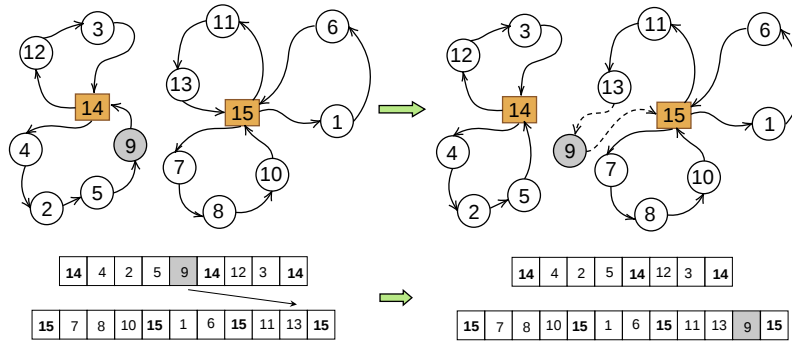
- (ii) $Swap(2,1)$ -InterDepot: permutação de dois clientes adjacentes v_j e v_{j+1} de uma rota r_k por um cliente v_t em outra rota r_l . Na Figura 4.17, os clientes 12 e 3 do depósito 14 foram trocados com o cliente 13 pertencente ao depósito 14.

Figura 4.17: Exemplo de vizinhança $Swap(2,1)$ -InterDepot.

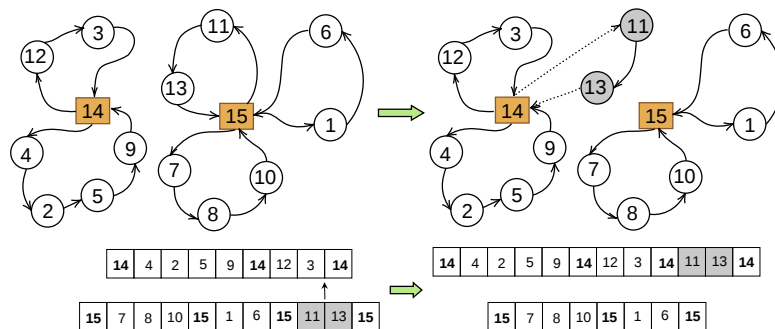
- (iii) $Swap(2, 2)$ -InterDepot: permutação entre dois clientes adjacentes (v_j, v_{j+1}) de uma rota r_k e outros dois clientes adjacentes (v_t, v_{t+1}) pertencentes a outra rota r_l . Na Figura 4.10, os clientes adjacentes 5 e 9 foram trocados com os clientes 12 e 3. Na Figura 4.18, os clientes 5 e 9 do depósito 14 foram trocados com os clientes 8 e 10 pertencentes ao depósito 15.

Figura 4.18: Exemplo de vizinhança $Swap(2,2)$ -InterDepot.

- (iv) $Shift(1,0)$ -InterDepot: consiste em transferir um cliente v_j de uma rota r_k para outra rota r_l pertencente a outro depósito.

Figura 4.19: Exemplo de $Shift(1,0)$ -InterDepot.

- (v) $ShiftDepot$: uma rota do conjunto do conjunto de rotas r^{d_1} do depósito d_1 é transferida para outro depósito d_2 . O depósito d_1 deve ter o maior número de rotas, sendo, no mínimo, duas rotas, enquanto d_2 é o depósito com a menor quantidade de rotas. Se o número de rotas do conjunto de depósitos for igual, d_1 e d_2 são escolhidos aleatoriamente.

Figura 4.20: Exemplo de $ShiftDepot$.

- (vi) *SwapDepot*: consiste em permutar uma rota do conjunto do conjunto de rotas r^{d_1} do depósito d_1 com outra rota do conjunto r^{d_2} de rotas do depósito d_2 .

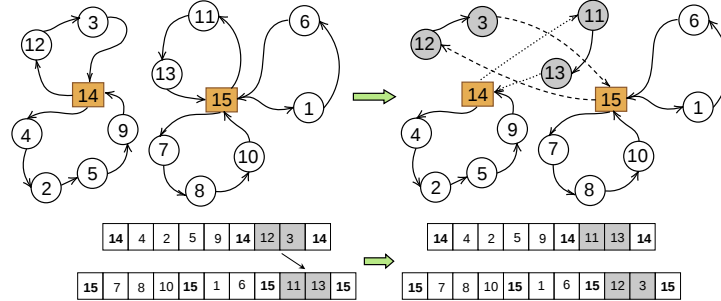


Figura 4.21: Exemplo de *SwapDepot*.

4.2.4 Operador de Destruição e Reconstrução

Os operadores de destruição e reconstrução promovem movimentos de realocação de um ou mais clientes em outras rotas pertencentes à solução do problema tratado. Dois operadores de destruição e reconstrução são utilizados neste trabalho. São eles:

- (i) *Eliminates Route*: consiste em eliminar uma rota selecionada aleatoriamente da solução. Os clientes da rota selecionada são reinseridos nas demais rotas.
- (ii) *Eliminates Smallest Route*: consiste em eliminar a rota com a menor quantidade de cliente da solução. Os clientes da rota selecionada são reinseridos nas demais rotas.

Em ambos os operadores, os clientes são realocados em outra rota somente se a realocação produzir uma solução viável.

4.3 Determinação da Solução Inicial

Como procedimento usado para a geração da solução inicial, utilizou-se a técnica de primeiro agrupar e depois rotear, inspirado no algoritmo de Gillet e Johnson (Gillett e Johnson, 1976). Dessa forma, os clientes são distribuídos ao depósito de maneira balanceada. Neste caso, o conjunto composto por N clientes é totalmente atribuído ao depósito em instâncias com um único depósito ($M = 1$); divididos em partes iguais, quando M é par; e, quando M é ímpar e $M > 2$, haverá um depósito com um cliente a mais que o restante. Inicialmente, a heurística determina a distância entre cada consumidor $i \in \mathcal{V}^{CLI}$ e o depósito j pertencente ao conjunto $\mathcal{V}^{DEP}$; em seguida, determina-se os dois depósitos mais próximos ao consumidor i e

suas respectivas distâncias, dadas por a_1 e a_2 ; após, calcula-se a taxa $vr_i = a_1/a_2$, associada a cada cliente i . Os valores desta taxa referentes a cada consumidor são classificados de forma crescente e, a partir deste ordenamento, o consumidor i será atribuído ao depósito j mais próximo. O processo se repete até que todos os clientes sejam alocados em apenas um depósito. Ao fim da alocação, as rotas são criadas de forma parcialmente gulosa, com uma probabilidade de escolha aleatória $\gamma = 60\%$ e, quando a aleatoriedade não é estabelecida, respeita-se a ordem de precedência dos clientes, dada por vr_i . Caso $\gamma = 0$, o procedimento torna-se determinístico e a composição de rotas associadas ao depósito j será sempre a mesma.

A Figura 4.22 mostra, em três passos, um exemplo de como é o processo de geração da população inicial. Na Figura 4.22(a), os clientes encontram-se distribuídos geograficamente; em seguida, na Figura 4.22(b), aplica-se o agrupamento (*clustering*), associando os clientes a seu referido depósito; na terceira etapa, geram-se as rotas.

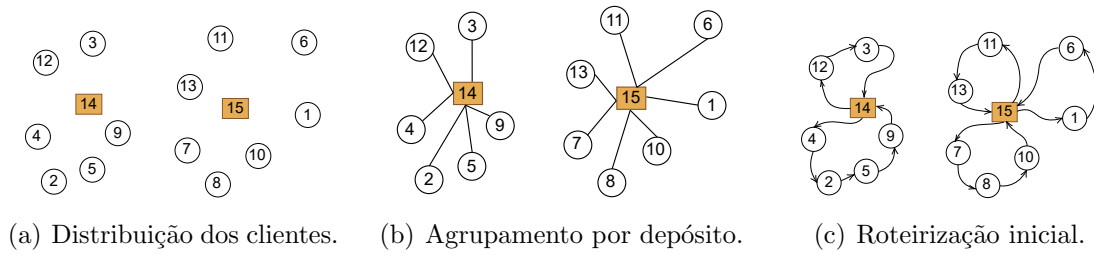


Figura 4.22: Exemplo para geração da solução inicial.

Para compor a solução, o cliente selecionado será adicionado à primeira rota. Caso seja viável, permanece; caso contrário, irá compor outra rota existente, caso exista; senão, uma nova rota será gerada, apenas com o elemento escolhido. Durante a geração da solução inicial, a restrição relativa ao número máximo de veículos é relaxada, podendo ser maior que K . Essa relaxação é necessária para garantir a geração de uma solução inicial.

4.3.1 Procedimento para Geração de *Clusters*

Enquanto os clientes não são associados a nenhuma rota, o método encontra um depósito que pode aceitá-los. Isso é mostrado no Algoritmo 9. Nas linhas 8-18, determina-se os dois depósitos mais próximos a cada cliente $i \mid i \in \mathcal{C}'$. Esses depósitos foram identificados como $d1$ e $d2$ (linhas 11 e 13). As distâncias entre o cliente i e os depósitos $d1$ e $d2$ foram nomeados $c_{d1,i}$ e $c_{d2,i}$, respectivamente.

Na linha 15 é calculada a razão de proximidade entre cada cliente i e seus depósitos $d1$ e $d2$. O conjunto de pares $\{i, c_{d1,i}/c_{d2,i}\}$, descrito na linha 16, constitui o conjunto $\mathcal{HR}$. Este conjunto é classificado em ordem ascendente a partir dessas proporções na

linha 19. Então, os clientes são designados para um desses depósitos. Este procedimento começa na linha 22 e termina na linha 36.

A distribuição entre os depósitos ocorre de forma equilibrada (linhas 28 a 34). Portanto, na ordem dada por $\mathcal{HR}$, cada par formado pelo cliente $i \in \mathcal{C}'$ e seu respectivo depósito $d1$ (or $d2$) é atribuído aos conjuntos $\mathcal{HD}$ e $\mathcal{C}''$. Na nova composição do conjunto $\mathcal{C}'$, todos os clientes atribuídos são removidos (linha 37). Os depósitos que atingem seu limite de clientes são removidos, como mostrados entre as linhas 39 a 46. Quando todos os clientes são alocados, o conjunto $\mathcal{HD}$, com os pares de clientes e depósitos $\mathcal{HD}$, é retornado (linha 48).

4.3.2 Procedimento para Geração de Rotas

O Algoritmo 10 mostra o procedimento para construir as rotas. As rotas são construídas respeitando a capacidade dos veículos e o tempo máximo de rota, além das janelas de tempo dos clientes. Essas rotas são construídas de forma parcialmente gulosa. O cliente i e o depósito d , em que $\{i, d\} \in \mathcal{HD}$, são escolhidos a partir de um valor previamente fixado $\gamma \in [0, 1]$. Se um valor aleatório real for menor ou igual a γ (linha 4), o elemento na j -ésima posição em $\mathcal{HD}$ é escolhido (linha 5).

Caso contrário, obtém-se o primeiro elemento do conjunto $\mathcal{HD}$ (linha 3). O valor de γ foi fixado em 0,6. Este valor foi definido a partir de testes empíricos. Nas linhas 7 a 8, cliente e o depósito são definidos. As rotas são representadas no subconjunto $r_d \subset \mathcal{R}$, que recebe o cliente i , como mostrado na linha 10. Na linha 11, o par $\{i, d\}$ é descartado de $\mathcal{HD}$. Todos os clientes devem ser atendidos. Pode ocorrer essa imposição de que o número de veículos utilizados pode ser maior que $|\mathcal{K}|$. O algoritmo é finalizado na linha 13, e o conjunto de rotas $\mathcal{R}$ é retornado.

Algoritmo 9 GerarClusters($\mathcal{C}, \mathcal{D}$)

```

1:  $\mathcal{C}' \leftarrow \mathcal{C}; \mathcal{D}' \leftarrow \mathcal{D}; \{\text{Cópias dos conjuntos: clientes e depósitos}\}$ 
2:  $\mathcal{HD} \leftarrow \emptyset; \{\text{Conjunto de pares } (i, d) \text{ de clientes } i \text{ atribuídos a depósitos } d\}$ 
3: while  $|\mathcal{C}'| > 0$  do
4:    $\mathcal{A}' \leftarrow \emptyset; \{\text{Conjunto de pares } (i, d), \text{ onde } d \text{ é o depósito mais próximo do cliente } i\}$ 
5:    $\mathcal{B}' \leftarrow \emptyset; \{\text{Conjunto de pares } (i, d), \text{ onde } d \text{ é o segundo depósito mais próximo do cliente } i\}$ 
6:    $\mathcal{HR} \leftarrow \emptyset; \{\text{Conjunto de pares } (i, rd1d2), \text{ onde } rd1d2 \text{ é a razão de proximidade do cliente } i$ 
     para os dois depósitos mais próximos}\}
7:    $j \leftarrow 1;$ 
8:   while  $j \leq |\mathcal{C}'|$  do
9:     Seja  $i$  o cliente da  $j$ -ésima posição de  $\mathcal{C}' \mid i \in \mathcal{C}'$ ;
10:     $c_{di} \leftarrow$  distância entre o cliente  $i$  e cada depósito  $d \in \mathcal{D}'$ ;
11:     $d1 = \arg \min_d \{c_{di}\}, \forall d \in \mathcal{D}'$ ;
12:     $\mathcal{A}' \leftarrow \mathcal{A}' \cup \{(i, d1)\};$ 
13:     $d2 = \arg \min_d \{c_{di}\}, \forall d \in \mathcal{D}', d1 \neq d2$ ;
14:     $\mathcal{B}' \leftarrow \mathcal{B}' \cup \{(i, d2)\};$ 
15:     $rd1d2 = c_{d1,i}/c_{d2,i}$ ;
16:     $\mathcal{HR} \leftarrow \mathcal{HR} \cup \{(i, rd1d2)\};$ 
17:     $j \leftarrow j + 1$ ;
18:  end while
19:  Ordenar  $\mathcal{HR}$  em ordem crescente, de acordo com o valor de  $rd1d2$ ;
20:   $j \leftarrow 1$ ;
21:   $\mathcal{C}'' \leftarrow \emptyset; \{\text{Conjunto de clientes designados a algum depósito}\}$ 
22:  while  $j \leq |\mathcal{C}'|$  do
23:    Seja  $i$  o cliente da  $j$ -th posição de  $\mathcal{HR} \mid i \in \mathcal{C}'$ ;
24:    Seja  $d1$  o depósito associado ao cliente  $i \mid (i, d1) \in \mathcal{A}'$ ;
25:    Seja  $d2$  o depósito associado ao cliente  $i \mid (i, d2) \in \mathcal{B}'$ ;
26:    Seja  $td$  o total de clientes em  $\mathcal{HD}$  atribuído ao depósito  $d1$ 
27:    Seja  $td'$  o total de clientes em  $\mathcal{HD}$  atribuído ao depósito  $d2$ 
28:    if  $td < \lceil |\mathcal{C}|/|\mathcal{D}| \rceil$  then
29:       $\mathcal{HD} \leftarrow \mathcal{HD} \cup \{(i, d1)\};$ 
30:       $\mathcal{C}'' \leftarrow \mathcal{C}'' \cup \{i\}$ 
31:    else if  $td' < \lceil |\mathcal{C}|/|\mathcal{D}| \rceil$  then
32:       $\mathcal{HD} \leftarrow \mathcal{HD} \cup \{(i, d2)\};$ 
33:       $\mathcal{C}'' \leftarrow \mathcal{C}'' \cup \{i\}$ 
34:    end if
35:     $j \leftarrow j + 1$ ;
36:  end while
37:   $\mathcal{C}' \leftarrow \mathcal{C}' \setminus \mathcal{C}''$ ;
38:   $j \leftarrow 1$ ;
39:  while  $j \leq |\mathcal{D}'|$  do
40:    Seja  $d$  o depósito da  $j$ -th posição de  $\mathcal{D}' \mid d \in \mathcal{D}$ ;
41:    Seja  $td$  o total de clientes em  $\mathcal{HD}$  atribuído ao depósito  $d$ 
42:    if  $td \geq \lceil |\mathcal{C}|/|\mathcal{D}| \rceil$  then
43:       $\mathcal{D}' \leftarrow \mathcal{D}' \setminus \{d\};$ 
44:    end if
45:     $j \leftarrow j + 1$ ;
46:  end while
47: end while
48: return  $\mathcal{HD}$ ;

```

Algoritmo 10 CriarRotas($\mathcal{HD}$, $\mathcal{C}$, $\mathcal{D}$)

```

1:  $\mathcal{R} \leftarrow \emptyset$ ; {Conjunto de soluções do MDVRPTW*}
2: while  $|\mathcal{HD}| > 0$  do
3:    $j \leftarrow 1$ ;
4:   if  $\text{rand}(0, 1) \leq \gamma$  then
5:      $j \leftarrow \text{rand}(|\mathcal{HD}|)$ ;
6:   end if
7:   Seja  $i$  o cliente da  $j$ -ésima posição em  $\mathcal{HD}$  |  $i \in \mathcal{C}$ ;
8:   Seja  $d$  o depósito associado ao cliente  $i$  na  $j$ -ésima posição em  $\mathcal{HD}$  |  $d \in \mathcal{D}$ ;
9:   Seja  $r_d$  ser a lista associada ao depósito  $d$  |  $r_d \subset \mathcal{R}$ ;
10:   $r_d \leftarrow r_d \cup \{i\}$ ; {Compor as rotas do depósito  $d$  respeitando as janelas de tempo dos clientes,
    as capacidades máximas dos veículos e a duração dessas rotas};
11:   $\mathcal{HD} \leftarrow \mathcal{HD} \setminus \{(i, d)\}$ ;
12: end while
13: return  $\mathcal{R}$ ;

```

4.4 Procedimento de Perturbação Proposto

A etapa de perturbação é aplicada para que o algoritmo não fique preso em uma mesma região do espaço de solução do problema e para explorar outras regiões, como discutido na Seção 3.1. Para fazer uma melhor exploração do espaço de soluções do problema, e evitar variações bruscas para outras regiões, é aplicado um método que perturba a solução de maneira gradual. O algoritmo aumenta progressivamente o nível de perturbação em uma solução quando ela está presa em um ótimo local. Portanto, cada estrutura de vizinhança é utilizada de 1 a $maxNível$ vezes e isso evita grandes saltos entre as regiões de busca. Com essa estratégia, pretende-se que não ocorra uma grande variação no espaço de soluções do problema e que, gradualmente, ocorra maiores diversificações deste espaço de busca.

O procedimento de perturbação proposto segue conforme introduzido em [Rego e Souza \(2019\)](#). Os autores propõem que o nível da perturbação aumente gradativamente. Esta decisão permite que a partir de um mesmo operador, os vizinhos se distanciem de maneira mais suave no espaço de busca. A perturbação ocorre através da mudança no operadores utilizados e pela quantidade de vezes que são aplicados à solução. O Algoritmo 11 mostra este procedimento.

Algoritmo 11 Perturbacao ($s, v, p, \mathcal{PO}$)

```

1: for ( $i = 1; i \leq p; i++$ ) do
2:   Gerar vizinho  $s' \in \mathcal{PO}_v(s)$ ;
3:    $s \leftarrow s'$ 
4: end for
5: return  $s'$ ;

```

O princípio dessa abordagem está em explorar inicialmente regiões mais próximas à solução corrente. À medida que a diversificação aumenta, é possível explorar regiões mais distantes no espaço de soluções. No algoritmo, o conjunto de vizinhanças utilizadas na perturbação é dado por $\mathcal{PO} = \{\mathcal{PO}_1^1, \mathcal{PO}_1^2, \dots, \mathcal{PO}_1^{maxNivel}, \mathcal{PO}_2^1, \mathcal{PO}_2^2, \dots, \mathcal{PO}_2^{maxNivel}, \dots, \mathcal{PO}_{maxVizPerturbacao}^1, \mathcal{PO}_{maxVizPerturbacao}^2, \dots, \mathcal{PO}_{maxVizPerturbacao}^{maxNivel}\} \mid \mathcal{PO}_p^v \in \mathcal{PO}; v = 1, 2, \dots, maxVizPerturbacao; p = 1, 2, \dots, maxNivel$. Nesta representação, as vizinhanças identificadas pelo mesmo índice são geradas a partir de um único operador associado à vizinhança v . Este operador é aplicado na solução s $p = \{1, 2, \dots, maxNivel\}$ vezes, em que o vizinho s' é gerado. Inicialmente um vizinho s' mais próximo à solução s é retornado, e, à medida que o valor de p aumenta, estes vizinhos podem se distanciar. A partir das perturbações em s , é possível promover uma boa diversificação nas soluções e guiá-las para regiões mais promissoras do espaço de soluções.

4.5 Procedimentos de Busca Local Propostos

Nesta seção são apresentados os procedimentos de busca local propostos neste trabalho e utilizados nos algoritmos desenvolvidos.

4.5.1 Procedimento Proposto de Busca Local Adaptativa

O método consiste na busca das melhores estruturas de vizinhança para o problema tratado. Para essa heurística, em geral, as vizinhanças são ordenadas a partir de algum mecanismo capaz de mapear aquelas mais adaptadas ao problema tratado. Um sistema de pontuação é atribuído a cada estrutura de vizinhança, permitindo identificar aquelas que produzem melhores resultados e, conseqüentemente, têm maiores chances de serem escolhidas durante o procedimento de busca local. Em nossa proposta, ocorre uma exploração em todas as vizinhanças através do procedimento *Randomized Variable Neighborhood Descent* (RVND). Durante este processo, armazenamos na variável *sucesso* uma pontuação atribuída a cada uma das estruturas de vizinhança exploradas. O algoritmo RVND proposto é mostrado no Algoritmo 12.

Algoritmo 12 RVND($s, s', \mathcal{N}, sucesso$)

```

1:  $\mathcal{N}' \leftarrow rand(\mathcal{N})$ ;
2:  $v \leftarrow 1$ ;
3:  $s' \leftarrow s$ ;
4: while  $v \leq |\mathcal{N}'|$  do
5:    $s'' \leftarrow \arg \min_{z \in \mathcal{N}'_v(s')} f(z)$ ; {Melhor vizinho em  $\mathcal{N}'_v(s')$ }
6:   VizinhancaSequencial( $s', s'', v$ );
7:   AtualizarSucesso( $s, s', s'', v, sucesso$ );
8: end while
9: return  $s'$ ;

```

Neste algoritmo, a alteração proposta está na operação de chamada ao método AtualizarSucesso (linha 7). O método para a pontuação atribuída às estruturas de vizinhanças é mostrado no Algoritmo 13.

Algoritmo 13 AtualizarSucesso($s, s', s'', v, sucesso$)

```

1: if  $f(s'') < f(s)$  then
2:    $sucesso[v] \leftarrow sucesso[v] + 15$ ;
3: else if  $f(s'') < f(s')$  then
4:    $sucesso[v] \leftarrow sucesso[v] + 5$ ;
5: end if

```

A pontuação é atribuída da seguinte maneira: se o vizinho s'' retornado na vizinhança $\mathcal{N}'_v$ melhora somente a solução perturbada s' , ou seja, se $f(s) < f(s'') < f(s')$, então a variável *sucesso*, associada ao conjunto $\mathcal{N}'_v$, é aumentada em 5 unidades. Caso a solução retornada também seja melhor que s | $f(s'') < f(s') < f(s)$, então a variável

sucesso é aumentado em 15 unidades. Estas situações são mostradas no Algoritmo 13 (linhas 2 e 4).

Feita a atribuição dos pontos às estruturas de vizinhança, é possível identificar aquelas que promoveram melhores resultados durante o processo de busca por melhores vizinhos. Diante desta possibilidade, propomos selecionar apenas uma das várias estruturas de vizinhanças possíveis, com prioridade para aquelas com a melhor pontuação.

No Algoritmo 14, é mostrado um modelo no qual apenas uma das estrutura de vizinhança será escolhida na linha 1 e, em seguida, explorada na busca de um melhor vizinho. O processo de atualização continua ocorrendo, conforme mostrado na linha 4.

Algoritmo 14 SingleLocalSearch($s, s', \mathcal{N}, sucesso$)

```

1:  $v \leftarrow \text{Roleta}(sucesso)$ ;
2:  $s' \leftarrow s$ ;
3:  $s'' \leftarrow \arg \min_{z \in \mathcal{N}_v(s')} f(z); \{\text{Melhor Vizinho em } \mathcal{N}_v(s')\}$ 
4: AtualizarSucesso( $s, s', s'', v, sucesso$ );
5: return  $s''$ ;

```

A vizinhança é selecionada através do método da roleta com base na variável *sucesso*. Neste método, todas as vizinhanças têm chance de serem escolhidas; no entanto, aquelas com os maiores valores para a variável *sucesso* têm maiores chances. Por exemplo, sejam $\mathcal{N}_1$, $\mathcal{N}_2$ e $\mathcal{N}_3$ estruturas de vizinhanças e considere que o RVND seja executada 10 vezes. No exemplo, $\mathcal{N}_1$ melhora a solução em três das dez execuções; $\mathcal{N}_2$ em cinco; e $\mathcal{N}_3$ em duas. Após a fase de ranqueamento, as probabilidades associadas a $\mathcal{N}_1$, $\mathcal{N}_2$ e $\mathcal{N}_3$ são dadas pela variável $sucesso = \{\frac{15}{50}, \frac{25}{50}, \frac{10}{50}\} = \{0.3, 0.5, 0.2\}$, respectivamente.

A Figura 4.23 ilustra o processo de roleta proposto. Nela, v indica a estrutura de vizinhança; *sucesso* indica o valor atribuído a cada vizinhança como apresentado no Algoritmo 13. Para a aplicação do método da roleta, os valores de *sucesso* são normalizados e, em seguida, são somados de forma acumulativa. Para cada estrutura de vizinhança, um intervalo é gerado.

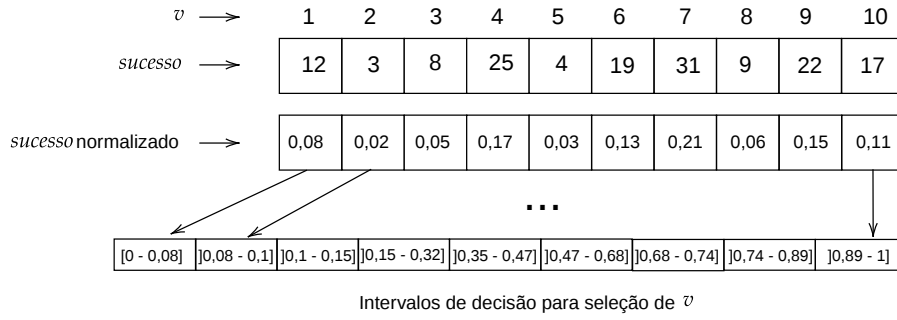


Figura 4.23: Exemplo das estruturas de decisão para o método da roleta.

A proposta de escolha por apenas uma estrutura de vizinhança durante o processo de busca local se justifica porque a utilização de estruturas de vizinhança mais adaptadas pode alcançar os mesmos resultados que a utilização de um conjunto de vizinhanças geradas por operadores não especializados no problema tratado. Outra característica importante está no fato de o método prover menor esforço computacional, já que a exploração de várias estruturas de vizinhança ocasiona o aumento no custo computacional.

A seguir será apresentado o procedimento de busca local adaptativa proposto. Ao se iniciar o algoritmo VNS, não é possível determinar quais vizinhanças são mais promissoras para serem exploradas. Por isso, torna-se necessário descobrir a quais devem se destinar maior esforço computacional. O procedimento proposto foi denominado como BLA (Busca Local Adaptativa) e é mostrado no Algoritmo 15.

Algoritmo 15 $BLA(s, s', \mathcal{N}, sucesso, it, maxTreinamento)$

```

1: if  $it < maxTreinamento$  then
2:    $s'' \leftarrow RVND(s, s', \mathcal{N}, sucesso)$ ;
3: else
4:    $s'' \leftarrow SingleLocalSearch(s, s', \mathcal{N}, sucesso)$ ;
5: end if
6: return  $s''$ ;

```

O procedimento RVND mostrado no Algoritmo 12 é utilizado para efetuar o ordenamento das estruturas de vizinhança. Esta primeira parte do algoritmo denominamos como fase de treinamento, pois, a partir dos valores armazenados na variável *sucesso*, é possível acessar as estruturas de vizinhança mais adaptadas ao problema tratado. Para que ocorra uma boa identificação dessas vizinhanças, o procedimento RVND é executado $maxTreinamento$ vezes (linha 1). Após esta fase, é possível aplicar o método de busca local por apenas uma das vizinhanças através do método SingleLocalSearch (Algoritmo 14). Outros exemplos sobre métodos adaptativos podem ser encontrados em Pisinger e Ropke (2007), Li et al. (2016), Aziz et al. (2013) e Aziz et al. (2017).

4.6 Funções de Avaliação

Nesta seção são apresentadas as funções de avaliação propostas. A primeira função impõe um maior peso no uso dos veículos, enquanto a segunda faz a ponderação na distância percorrida.

4.6.1 Função de Avaliação para Redução de Frota

Avaliamos uma solução s para o problema PRVJTMD visando minimização do número de veículos usados utilizando a função $F(s)$ descrita como:

$$F(s) = \beta \cdot L + \alpha \cdot |\mathcal{U}| + \varphi \quad (4.2)$$

Esta função de avaliação é baseada na proposta apresentada em [Ombuki et al. \(2006\)](#) para o PRVJT. Ela é composta por três parcelas, com objetivos distintos a serem minimizados. O primeiro objetivo é reduzir a distância total percorrida por todos os veículos (L). O segundo objetivo refere-se à quantidade de veículos utilizados $|\mathcal{U}| \leq |\mathcal{K}|$. O último objetivo refere-se às violações que podem ocorrer devido à aplicação dos movimentos para explorar o espaço de solução do problema. Essas violações podem ocorrer devido ao excesso de veículos disponíveis em um depósito, sobrecarga de veículos, atrasos no atendimento ao cliente e extrapolação da duração máxima permitida na rota.

Os dois primeiros objetivos presentes na Equação (4.2) são ponderados pelos pesos β e α , que refletem a importância relativa de cada um deles. Como proposto em [Ombuki et al. \(2006\)](#), utilizamos $\beta = 0,001$. O parâmetro α é calculado por:

$$\alpha = \frac{1}{N} \sum_{i \in \mathcal{C}} \sum_{j \in \mathcal{C}} c_{ij} \quad (4.3)$$

De acordo com a Equação (4.3), α representa a distância média entre cada par de clientes.

O peso φ é calculado conforme abaixo:

$$\varphi = \sum_{k \in \mathcal{U}} \phi(k) \quad (4.4)$$

Nesta expressão, o parâmetro $\phi(k)$ representa a penalidade por não respeitar as restrições impostas na rota de cada veículo k , sendo calculado como:

$$\phi(k) = \omega^Q \max\{0, Q_k - VC\} + \omega^T \max\{0, T_k - MT\} + \omega^{TW} TW_k \quad (4.5)$$

Na Equação (4.5), o parâmetro ω^Q penaliza a sobrecarga do veículo; o parâmetro ω^T penaliza o tempo excedido na duração da rota do veículo k ; e o parâmetro ω^{TW} penaliza atrasos na entrega de produtos pelo veículo k . O atraso TW_k é a soma dos tempos que excedem a janela de tempo final de cada cliente pertencente à rota do veículo k .

Por sua vez, os valores das ponderações são calculados a partir do parâmetro α , da média das demandas $\bar{q}$, e dos tempos médios de serviço para os consumidores, conforme definido abaixo:

$$\omega^Q = \frac{\alpha}{\bar{q}} \quad (4.6)$$

$$\omega^T = \omega^{TW} = \frac{\alpha}{h} \quad (4.7)$$

Essas violações são aceitas devido ao elevado grau de dificuldade em solucionar este problema, além de ser uma estratégia comumente utilizada na literatura, como, por exemplo, pode ser visto nos procedimentos descritos em [Polacek et al. \(2008\)](#) e [Vidal et al. \(2013a\)](#).

4.6.2 Função de Avaliação para Redução da Distância

A função de avaliação descrita na Equação 4.2 pode ser adaptada para avaliar as soluções geradas para o PRVJTMD e PRVAJTMD. Neles, o objetivo é apenas minimizar a distância total percorrida. Para a adaptação necessária, considera-se que $\beta = 1$ e $\alpha = 0$, resultando em:

$$F'(s) = L + \varphi \quad (4.8)$$

A parcela φ da função objetivo (4.8) permanece conforme descrito na Equação (4.4).

4.7 Algoritmos e Métodos Propostos

Nesta seção são apresentados os algoritmos propostos para a solução dos problemas PRVJTMD*, PRVAJTMD e PRVAJTMD*. Na Seção 4.7.2, o algoritmo SGVNSBLA proposto nesta tese é apresentado e discutido. Em seguida, na Seção 4.7.3, o algoritmo SGVNS é apresentado, e na Seção 4.7.4 uma outra versão do algoritmo com busca local RVND é mostrada. O último algoritmo a ser mostrado na Seção 4.7.1 é o VNS com busca local VND, denominado aqui por BGVNS. Os resultados obtidos por esses métodos são mostrados e comparados no Capítulo 5. Nesses quatro métodos, as estruturas de vizinhanças utilizadas na perturbação e busca local se mantiveram

as mesmas.

4.7.1 Basic General VNS

O procedimento textitGeneral Variable Neighborhood Search (GVNS) é definido em Hansen et al. (2010, 2017) como uma variação da meta-heurística VNS quando a busca local utiliza o algoritmo VND. Neste trabalho, o método é denominado por *Basic* – BGVNS Hansen et al. (2017). Isso se faz necessário para haja uma melhor distinção entre os outros métodos propostos. O Algoritmo 16 mostra o procedimento BGVNS adaptado para os problemas apresentados nesta tese.

Algoritmo 16 BGVNS($\mathcal{C}, \mathcal{D}, \mathcal{LSO}, \mathcal{PO}, iterMax, maxTempo$)

```

1:  $\mathcal{HD} \leftarrow \text{GerarClusters}(\mathcal{C}, \mathcal{D}); \{\text{Algoritmo 9}\}$ 
2:  $s \leftarrow \text{CriarRotas}(\mathcal{HD}, \mathcal{C}, \mathcal{D}); \{\text{Algoritmo 10}\}$ 
3:  $v \leftarrow 1; \{\text{Vizinhança inicial}\}$ 
4:  $iter \leftarrow 0; \{\text{Contador de iterações}\}$ 
5:  $\{\text{Critério de parada}\}$ 
6: while  $iter < iterMax$  ou  $t < maxTempo$  do
7:    $s' \leftarrow \text{perturbacaoBasica}(s, v, \mathcal{PO}); \{\text{Algoritmo 2}\}$ 
8:    $s'' \leftarrow \text{VND}(s', \mathcal{LSO}); \{\text{Algoritmo 7}\}$ 
9:   AtualizarVizinhanca( $s, s'', v, |\mathcal{PO}|$ );  $\{\text{Algoritmo 17}\}$ 
10:  AtualizarSolucao( $s, s'', iter$ );  $\{\text{Algoritmo 20}\}$ 
11: end while
12: return  $s$ ;
```

Na fase de perturbação, os operadores utilizados são aplicados de forma sequencial. Não há uma intensificação com o mesmo operador, diferentemente do que ocorre com os procedimentos SGVNSBLA, SGVNS e SGVNS-RVND, conforme visto a seguir. A atualização desses operadores é apresentada no Algoritmo 17. Esse procedimento segue conforme definido no Algoritmo 3, apresentado na Seção 3.3. Nele, a alteração ocorre de maneira sequencial até o máximo de elementos pertencentes ao conjunto $\mathcal{PO}$.

Algoritmo 17 AtualizarVizinhanca ($s, s'', v, maxVizinhanca$)

```

1: if  $f(s'') < f(s)$  then
2:    $v \leftarrow 1; \{\text{Retornar à primeira vizinhança}\}$ 
3: else
4:    $v \leftarrow v + 1;$ 
5: end if
6: if  $v > maxVizinhanca$  then
7:    $v \leftarrow 1; \{\text{Retornar à primeira vizinhança}\}$ 
8: end if
```

4.7.2 Smart GVNS Base Proposto

O algoritmo base proposto neste trabalho foi inspirado no algoritmo VNS proposto por (Mladenović e Hansen, 1997). Neste algoritmo, a fase de perturbação segue os critérios propostos em Rego e Souza (2019), como descrito na Seção 4.4. Na fase de busca local, são utilizados os algoritmos descritos na Seção 4.5.

O Algoritmo 18 mostra o pseudocódigo da adaptação do método VNS proposta para a resolução dos problemas de roteamento de veículos com janelas de tempo e múltiplos depósitos definidos neste trabalho. Como ele utiliza um procedimento de busca local adaptativa, o método foi nomeado como *Smart General Variable Neighborhood Search* (SGVNS) com Busca Local Adaptativa – SGVNSBLA.

Algoritmo 18 SGVNSBLA($\mathcal{C}$, $\mathcal{D}$, $\mathcal{LSO}$, $\mathcal{PO}$, $iterMax$, $maxTempo$, $maxNivel$, $maxTreinamento$, $maxItAdaptativo$, $maxVizPerturbacao$, $sucesso$)

```

1:  $\mathcal{HD} \leftarrow \text{GerarClusters}(\mathcal{C}, \mathcal{D})$ ; {Algoritmo 9}
2:  $s \leftarrow \text{CriarRotas}(\mathcal{HD}, \mathcal{C}, \mathcal{D})$ ; {Algoritmo 10}
3:  $p \leftarrow 1$ ; {Nível inicial de perturbação}
4:  $v \leftarrow 1$ ; {Vizinhança inicial}
5:  $iter \leftarrow 0$ ; {Contador de iterações}
6:  $itAdaptativo \leftarrow 0$ ;
7: {Critério de parada}
8: while  $iter < iterMax$  or  $t < maxTempo$  do
9:    $s' \leftarrow \text{Perturbacao}(s, v, p, \mathcal{PO})$ ; {Algoritmo 11}
10:   $s'' \leftarrow \text{BLA}(s, s', \mathcal{N}, sucesso, itAdaptativo, maxTreinamento)$ ; {Algoritmo 15}
11:  AtualizarNivelPerturbacao( $s, s'', v, p, maxNivel, maxVizPerturbacao$ ); {Algoritmo 19}
12:  AtualizarSolucao( $s, s'', iter$ ); {Algoritmo 20}
13:  ControladorBLA( $itAdaptativo, maxItAdaptativo, sucesso$ ); {Algoritmo 21}
14: end while
15: return  $s$ ;

```

O algoritmo recebe como parâmetros o conjunto de clientes $\mathcal{C}$; o conjunto de depósitos $\mathcal{D}$; o conjunto de operadores geradores das estruturas de vizinhanças a serem exploradas na busca local $\mathcal{LSO}$ (*Local Search Operators*); o conjunto dos operadores geradores das estruturas de vizinhanças a serem exploradas na perturbação $\mathcal{PO}$ (*Perturbation Operators*); a quantidade máxima de iterações sem melhora $iterMax$; o tempo máximo de execução $maxTempo$; o nível máximo de perturbação $maxNivel$; e a quantidade de execuções do RVND $maxTreinamento$.

Nas linhas 1 e 2, uma solução inicial é construída. Entre as linhas 3 e 6 são definidos o nível inicial de perturbação, o índice para a primeira vizinhança, o contador de iteração sem melhora e o contador de iteração para a aplicação do procedimento de busca local, respectivamente. Na linha 8, o algoritmo inicia seu laço principal. Esse laço termina quando o número de iterações sem melhora atinge seu valor máximo ou quando o tempo de execução atinge sua duração máxima. Na linha 9, a solução s é perturbada. As vizinhanças a serem exploradas durante o processo de perturbação

são definidas no conjunto $\mathcal{PO}$. A fase de busca local proposta é aplicada na linha 10. O controle do nível de perturbação é mostrado na linha 11. A atualização da solução é mostrada na linha 12. Na linha 13 as variáveis de controle da BLA podem ser atualizadas.

A seguir, os Algoritmos 19, 20 e 21 são descritos.

O processo de atualização do operador associado à vizinhança v e ao nível de perturbação p é mostrado no Algoritmo 19. Nesse processo, quando a solução s'' é melhor que a solução corrente s , a vizinhança a ser explorada durante a perturbação será a primeira do conjunto $\mathcal{PO}$, e o nível de perturbação $p = 1$ (linhas 2 e 3). Caso não ocorra melhoria da solução corrente, o nível de perturbação aumenta gradativamente (linha 5). A alteração para um próximo operador ocorre somente se o nível de perturbação ultrapassar seu valor máximo $maxNivel$ (linhas 7). Ao mudar de operador (linha 8), o nível de perturbação retorna para o valor $p = 1$ (linha 9). Quando todas as vizinhanças em $\mathcal{PO}$ forem exploradas, retorna-se para a primeira, como mostrado nas linhas 12 e 13.

Algoritmo 19 AtualizarNivelPerturbacao ($s, s'', v, p, maxNivel, maxVizPerturbacao$)

```

1: if  $f(s'') < f(s)$  then
2:    $v \leftarrow 1$ ; {Retornar à primeira vizinhança}
3:    $p \leftarrow 1$ ; {Retornar ao primeiro nível de perturbação}
4: else
5:    $p \leftarrow p + 1$ ; {Alterar o nível de perturbação}
6: end if
7: if  $p > maxNivel$  then
8:    $v \leftarrow v + 1$ ; {Mudar de vizinhança}
9:    $p \leftarrow 1$ ; {Retornar ao primeiro nível de perturbação}
10: end if
11: if  $v > maxVizPerturbacao$  then
12:    $v \leftarrow 1$ ; {Retornar à primeira vizinhança}
13:    $p \leftarrow 1$ ; {Retornar ao primeiro nível de perturbação}
14: end if

```

A atualização da solução s é apresentada pelo Algoritmo 20. Caso s'' seja melhor que a solução corrente s (linha 1), então a nova solução é atualizada, conforme mostrado na linha 2. Em seguida, a quantidade de iterações sem melhora é reiniciada (3). Caso a condição de atualização não seja atendida, a quantidade de iterações sem melhora é incrementada.

A atuação de determinadas estruturas de vizinhança pode alterar seu desempenho durante o processo de otimização, tornando-se necessário que as reavaliações ocorram periodicamente. Por este motivo, a pontuação de cada estrutura de vizinhança não é mantida até fim do processamento do algoritmo VNS. Após uma certa quantidade de

Algoritmo 20 AtualizarSolucao ($s, s'', iter$)

```

1: if  $f(s'') < f(s)$  then
2:    $s \leftarrow s''$ ; {Atualizar solução}
3:    $iter \leftarrow 0$ ; {Reseta iteração sem melhora}
4: else
5:    $iter \leftarrow iter + 1$ ; {Incrementa iteração sem melhora}
6: end if

```

iterações, a variável vetor *succeso* é reiniciada e o processo de ordenamento recomeça. No Algoritmo 21, a variável de decisão *itAdaptativo* a ser utilizada no algoritmo BLA (Algoritmo 15) é atualizada. Caso seu valor seja maior ou igual ao máximo permitido *maxItAdaptativo* (linha 1), então é reiniciada (linha 2), juntamente com a variável *succeso* (linha 3).

Algoritmo 21 ControladorBLA(*itAdaptativo, maxItAdaptativo, succeso*)

```

1: if  $itAdaptativo \geq maxItAdaptativo$  then
2:    $itAdaptativo \leftarrow 0$ ;
3:   Reiniciar succeso;
4: end if

```

4.7.3 Smart General Variable Neighborhood Search

Rego e Souza (2019) apresentam uma variação da meta-heurística VNS denominada *Smart General Variable Neighborhood Search* (SGVNS), baseada no *General Variable Neighborhood Search* (GVNS) descrito na Seção 4.7.1 (Hansen et al., 2010). Segundo os autores, a perturbação ocorre de maneira gradativa, dada pelas mudanças de vizinhanças e pela quantidade de vezes que um determinado operador utilizado na perturbação é aplicada à solução corrente. A decisão de aumentar o nível de perturbação ocorre quando não há melhoria na qualidade da solução. Essa procura por melhorias ocorre através da busca local VND. O procedimento SGVNS adaptado nesta tese é mostrado no Algoritmo 22.

A diferença entre os procedimentos SGVNSBLA e SGVNS está na fase de busca local. O primeiro algoritmo aplica o procedimento de Busca Local Adaptativa (BLA), enquanto o segundo utiliza o procedimento VND.

4.7.4 Smart GVNS com busca local RVND

Uma variação para o método proposto no Algoritmo 18 é mostrada no Algoritmo 23. Nesta proposta, a busca local utilizada é alterada para uma estrutura

Algoritmo 22 $\text{SGVNS}(\mathcal{C}, \mathcal{D}, \mathcal{LSO}, \mathcal{PO}, \text{iterMax}, \text{maxTempo}, \text{maxNivel}, \text{maxVizPerturbacao})$

```

1:  $\mathcal{HD} \leftarrow \text{GerarClusters}(\mathcal{C}, \mathcal{D})$ ; {Algoritmo 9}
2:  $s \leftarrow \text{CriarRotas}(\mathcal{HD}, \mathcal{C}, \mathcal{D})$ ; {Algoritmo 10}
3:  $p \leftarrow 1$ ; {Nível inicial de perturbação}
4:  $v \leftarrow 1$ ; {Vizinhança inicial}
5:  $\text{iter} \leftarrow 0$ ; {Contador de iterações}
6: {Critério de parada}
7: while  $\text{iter} < \text{iterMax}$  ou  $t < \text{maxTempo}$  do
8:    $s' \leftarrow \text{Perturbacao}(s, v, p, \mathcal{PO})$ ; {Algoritmo 11}
9:    $s'' \leftarrow \text{VND}(s', \mathcal{LSO})$ ; {Algoritmo 7}
10:  AtualizarNivelPerturbacao( $s, s'', v, p, \text{maxNivel}, \text{maxVizPerturbacao}$ ); {Algoritmo 19}
11:  AtualizarSolucao( $s, s'', \text{iter}$ ); {Algoritmo 20}
12: end while
13: return  $s$ ;

```

mais conhecida do algoritmo VNS com a fase de busca local RVND. O algoritmo foi nomeado nesta tese como SGVNS-RVND.

Algoritmo 23 $\text{SGVNS-RVND}(\mathcal{C}, \mathcal{D}, \mathcal{LSO}, \mathcal{PO}, \text{iterMax}, \text{maxTempo}, \text{maxNivel}, \text{maxVizPerturbacao})$

```

1:  $\mathcal{HD} \leftarrow \text{GerarClusters}(\mathcal{C}, \mathcal{D})$ ; {Algoritmo 9}
2:  $s \leftarrow \text{CriarRotas}(\mathcal{HD}, \mathcal{C}, \mathcal{D})$ ; {Algoritmo 10}
3:  $p \leftarrow 1$ ; {Nível inicial de perturbação}
4:  $v \leftarrow 1$ ; {Vizinhança inicial}
5:  $\text{iter} \leftarrow 0$ ; {Contador de iterações}
6: {Critério de parada}
7: while  $\text{iter} < \text{iterMax}$  ou  $t < \text{maxTempo}$  do
8:    $s' \leftarrow \text{Perturbacao}(s, v, p, \mathcal{PO})$ ; {Algoritmo 11}
9:    $s'' \leftarrow \text{RVND}(s', \mathcal{LSO})$ ; {Algoritmo 8}
10:  AtualizarNivelPerturbacao( $s, s'', v, p, \text{maxNivel}, \text{maxVizPerturbacao}$ ); {Algoritmo 19}
11:  AtualizarSolucao( $s, s'', \text{iter}$ ); {Algoritmo 20}
12: end while
13: return  $s$ ;

```

4.7.5 Aceitação de um vizinho

Como descrito na Seção 4.3, a solução inicial geralmente contém mais rotas do que veículos disponíveis. Este fato acarreta a relaxação do limite imposto pela restrição referente à quantidade de veículos disponíveis no início do procedimento. Mesmo com a relaxação, um vizinho é aceito na perturbação e/ou na busca local somente quando as rotas geradas são viáveis. Após uma solução com a quantidade de rotas produzidas estar dentro do limitante dado pela quantidade de veículos disponíveis ser encontrada, a restrição não é mais relaxada.

Para se avaliar um vizinho nos problemas em que o fator principal é a redução na quantidade de veículos utilizados, utiliza-se a função de avaliação definida na

Equação 4.2 durante todo o processo. Para problemas em que objetiva-se a redução na distância percorrida, a avaliação do vizinho é executada através da Equação 4.8.

4.8 Resumo do Capítulo

Neste capítulo foram discutidas as adaptações propostas para o VNS para a solução do PRVJTMD e suas variantes. Além disso, foram tratados os algoritmos com métodos propostos para atuarem na fase de perturbação e na fase de busca local. Foram apresentadas e discutidas a representação computacional dos problemas de roteamento apresentados, as estruturas de vizinhanças propostas, os métodos utilizados para geração da solução inicial. Foram estabelecidos os operadores geradores das estruturas de vizinhança exploradas nas duas fases. Alguns exemplos relativos aos tipos de movimentos foram apresentados. O método adotado para geração da população inicial foi apresentado.

Capítulo 5

Experimentos Computacionais

Este capítulo apresenta os resultados computacionais relativos à aplicação dos algoritmos desenvolvidos no Capítulo 4 para a solução de instâncias das diversas variantes do Problema de Roteamento de Veículos com Janelas de Tempo e Múltiplos Depósitos (PRVJTMD) descritas no Capítulo 2, tanto com roteamento fechado quanto com roteamento aberto. Para solucionar instâncias das variantes de roteamento fechado, PRVJTMD e o PRVJTMD*, foram utilizados os algoritmos BGVNS (Algoritmo 16), SGVNSBLA (Algoritmo 18), SGVNS (Algoritmo 22), descritos nas 4.7.1, Seções 4.7.2 e 4.7.3, respectivamente. Para as instâncias das variantes de roteamento aberto, PRVAJTMD e PRVAJTMD*, foram usados os algoritmos SGVNSBLA e SGVNS-RVND (Algoritmo 23). Esses algoritmos foram codificados em C++ e executados em um computador Intel Xeon E5620 2.40 GHz $\times$ 16 com 112 GB de memória RAM sob o sistema operacional Linux 64 bits.

O capítulo está organizado como segue. A Seção 5.1 apresenta os grupos de instâncias utilizados para os testes computacionais realizados. Para uma melhor determinação dos valores dos parâmetros utilizados nos algoritmos, foram executados testes de ajuste com instâncias dos grupos descritas. Esses ajustes são detalhados na Seção 5.2. Em seguida, os resultados computacionais para os dois problemas envolvendo roteamento fechado são apresentados na Seção 5.4. A Subseção 5.4.3 descreve um estudo sobre a influência dos operadores na busca local e no processo de perturbação, para o caso dos problemas de roteamento fechado. A Seção 5.5 apresenta os resultados para os problemas de roteamento aberto.

5.1 Instâncias de Teste

Dois conjuntos de instâncias foram utilizados como referência nesta tese para a aplicação dos algoritmos desenvolvidos, considerando-se os problemas descritos.

Esses conjuntos foram denominados como Grupo I e Grupo II. O Grupo I foi proposto em [Cordeau et al. \(2001\)](#) e está disponível em <http://neo.lcc.uma.es/vrp/vrp-instances/>. As instâncias do Grupo II foram propostas por [Vidal et al. \(2013a\)](#), e podem ser acessadas em <https://w1.cirrelt.ca/~vidalt/en/VRP-resources.html>.

O Grupo I contém 20 instâncias, envolvendo de 48 a 288 clientes; 4 a 6 depósitos; veículos com capacidade variando de 150 a 200; e duração da rota variando de 400 a 500. As dez primeiras instâncias deste grupo têm janelas de tempo estreitas, enquanto as demais dez têm janelas largas. As 28 instâncias apresentadas no Grupo II têm de 360 a 960 clientes; 4 a 12 depósitos; e veículos disponíveis por depósito entre 4 e 26. A distância entre os clientes nos dois grupos é a euclidiana, e será considerada como custo da viagem. A frota é homogênea em ambos os grupos.

5.2 Calibração de Parâmetros dos Algoritmos

Para a calibração dos valores dos parâmetros dos algoritmos desenvolvidos, utilizamos o pacote computacional IRACE (*Iterated Racing for Automatic Algorithm Configuration*) ([López-Ibáñez et al., 2016](#)). Este algoritmo foi projetado para fornecer os parâmetros mais apropriados para um algoritmo de otimização e um conjunto de instâncias. O *software* IRACE é executado como um pacote do *software* R, que é um ambiente gratuito para tratamento de dados estatísticos.

Foram selecionadas seis instâncias do Grupo I para serem utilizadas como referência para a calibração. Para determinar o grau de complexidade das instâncias, foi calculado o valor:

$$\rho = N \times M \times |\mathcal{D}|.$$

Foram definidas três instâncias com janelas estreitas e três com janelas largas. Foram selecionadas instâncias com valores próximos aos menores valores de ρ ; instâncias com valores próximos às suas médias; e instâncias com os maiores valores encontrados. Foram selecionadas 30% das instâncias do Grupo I para calibração dos parâmetros. Estas instâncias foram escolhidas porque contém diferentes características entre as instâncias do grupo. A Tabela 5.1 mostra as seis instâncias escolhidas e agrupadas pelas janelas de tempo estreitas e largas, respectivamente.

A Tabela 5.2 mostra a relação dos parâmetros a serem calibrados com os respectivos algoritmos. A relação indicada com um traço (“—”) significa que o parâmetro não foi calibrado, pois não é definido para o algoritmo.

A Tabela 5.3 apresenta as faixas dos valores testados e retornados pelo *software* IRACE para os parâmetros dos algoritmos desenvolvidos.

Tabela 5.1: Instâncias selecionadas para o ajuste de parâmetros dos algoritmos usando IRACE.

Instâncias	pr03	pr04	pr06	pr12	pr19	pr16
N	144	192	288	96	216	288
M	4	5	7	2	3	6
$ \mathcal{D} $	4	4	4	4	6	4
ρ	2304	3840	8064	768	3888	6912

Tabela 5.2: Grupo de instâncias.

Parâmetro	Algoritmo			
	SGVNSBLA	SGVNS-RVND	SGVNS	BGVNS
<i>iterMax</i>	✓	✓	✓	✓
<i>maxTempo</i>	✓	✓	✓	✓
<i>maxNivel</i>	✓	✓	✓	–
<i>maxPerturbacao</i>	✓	✓	✓	–
<i>maxTreinamento</i>	✓	–	–	–
<i>maxItAdaptativo</i>	✓	–	–	–

O campo “Descrição” faz uma breve descrição dos parâmetros; o segundo campo refere-se aos parâmetros objetos de calibração; no terceiro campo são mostrados os valores definidos para calibração. O último campo representa o valor retornado após a execução do processo de calibração pelo pacote IRACE. O termo “fechado” no campo “Descrição” refere-se aos problemas de roteamento fechado; por outro lado, o termo “aberto” refere-se aos problemas de roteamento aberto. No caso dos parâmetros *maxTreinamento* e *maxItAdaptativo*, foram utilizados percentuais em relação ao valor do parâmetro *iterMax*. Nestes casos, para *maxTreinamento*, foram testados valores representando percentuais de 20%, 30% e 40% de *iterMax*; de outro lado, com relação ao parâmetro *maxItAdaptativo*, foram testados valores representando percentuais de 40%, 60% e 80% de *iterMax*. Para exemplificar, seja *iterMax* = 400, então *maxTreinamento* = {80, 120, 160}. As faixas de valores foram definidas a partir de observações empíricas, nas quais os melhores resultados foram obtidos entre 20% e 40% para *maxTreinamento* e 40% e 80% de *iterMax*, respectivamente.

Devido às diferenças entre os problemas de roteamento de veículos fechados e os de roteamento abertos, o parâmetro *iterMax* foi avaliado em separado para ambos os casos. A referência a cada problema está na descrição da variável. Para o problema fechado, seu valor retornado foi o limitante superior do conjunto de valores informados. Por este motivo, testes empíricos foram realizados acima de 500 iterações sem melhora. Após os testes, foi observado que valores acima deste limite exigem tempos elevados no processamento, com melhorias insignificantes nos resultados. Desta forma, manteve-se o valor retornado.

Na fase de perturbação, a quantidade de operadores é representada pela variável

Tabela 5.3: Faixa dos valores dos parâmetros testados e os resultados retornados pelo IRACE

Descrição	Parâmetros	Valores fornecidos	Valor retornado
Quantidade de iterações sem melhoria (fechado)	<i>iterMax</i>	{400; 450; 500}	500
Quantidade de iterações sem melhoria (aberto)	<i>iterMax</i>	{50; 100; 200; 300}	100
Nível máximo de perturbação	<i>maxNivel</i>	{3; 4; 5; 6; 7}	5
Tempo máximo de processamento (Somente Grupo I)	<i>maxTempo</i>	{30; 60}	60
Quantidade de operadores utilizados na perturbação	<i>maxPerturbacao</i>	{3; 4; 5}	3
Quantidade de iterações na fase de treinamento	<i>maxTreinamento</i>	{0,2; 0,3; 0,4}	0,3
Quantidade de iterações durante a busca SingleLocalSearch	<i>maxItAdaptativo</i>	{0,4; 0,6; 0,8}	0,6

maxPerturbacao. O conjunto de operadores a serem avaliados durante essa fase foi formado pelos operadores *Shift(1,0)-InterDepot*, *Swap(1,1)-InterDepot*, *Eliminates Smallest Route*, *Swap(2,1)-InterDepot* e *Shift(2,0)*, nesta ordem. Como o resultado retornado para *maxPerturbacao* foi igual a três, foram, então, definidos os três primeiros operadores na perturbação.

As instâncias do Grupo II possuem alta complexidade, o que torna difícil a calibração dos parâmetros com elas. Para este conjunto, utilizou-se, como tempo máximo de processamento, o mesmo valor definido em Vidal et al. (2013a), dado por *maxTime* = 300 minutos.

5.3 Vizinhanças e Operadores Utilizados

As estruturas de vizinhança geradas têm características específicas. Devido a essa observação, não foram exploradas as mesmas vizinhanças para todos os problemas de roteamento de veículos com múltiplos depósitos abordados neste trabalho. A seguir, serão relatados os operadores responsáveis pela geração das vizinhanças que foram utilizadas em cada um dos problemas tratados.

Problemas com Roteamento Fechado – Fase de Perturbação: Para os problemas de roteamento fechado PRVJTMD* e PRVJTMD, são usados os seguintes operadores de perturbação:

- (i) *Shift(1,0)-InterDepot*, *Swap(1,1)-InterDepot*, e *Eliminates Smallest Route*.

As vizinhanças associadas foram exploradas na ordem apresentada.

Problemas com Roteamento Fechado – Fase de Busca Local: Para a mesma classe de problemas, foram utilizados os seguintes operadores de busca local:

- (i) Movimento interrota: *Reinsertion*, *Swap*, *Or-opt*, *2-opt* e *3-opt*.
- (ii) Movimento inter-rotas: *Swap(1,1)* e *Shift(1,0)*.

Tabela 5.4: Operadores utilizados pelos problemas tratados.

Fase		Problemas	
		PPRVJTMD*	PRVAJTMD e PRVAJTMD*
Perturbação		Shift(1,0)-InterDepot, Swap(1,1)-InterDepot, Eliminates Smallest Route	Eliminates Route, Shift(1,0)-InterDepot, Swap(1,1)-InterDepot
Busca Local	Intrarota	Reinsertion, Swap, Or-opt, 2-opt, 3-opt	Reinsertion, Swap, Or-opt, 2-opt, 3-opt
	Inter-rota	Swap(1,1), Shift(1,0)	Swap(1,1), Shift(1,0)
	Inter-depósito	Shift(1,0)-InterDepot, Swap(1,1)-InterDepot, Swap(2,2)-InterDepot	Shift(1,0)-InterDepot, Swap(1,1)-InterDepot
		Eliminates Smallest Route	

- (iii) Movimento Inter-depósitos: *Shift(1,0)-InterDepot*, *Swap(1,1)-InterDepot* e *Swap(2,2)-InterDepot*.
- (iv) *Eliminates Smallest Route*.

Problemas com Roteamento Aberto – Fase de Perturbação: Nos dois problemas abordados de roteamento aberto, na fase de perturbação foram utilizados os seguintes operadores, conforme a ordem apresentada:

- (i) *Eliminates Route*, *Shift(1,0)-InterDepot* e *Swap(1,1)-InterDepot*.

Problemas com Roteamento Aberto – Fase de Busca Local: Na fase de busca local, foram definidos os seguintes operadores:

- (i) Movimento interrota: *Reinsertion*, *Swap*, *Or-opt*, *2-opt* e *3-opt*.
- (ii) Movimento inter-rotas: *Swap(1,1)* e *Shift(1,0)*.
- (iii) Movimento inter-depósitos: *Swap(1,1)-InterDepot* e *Shift(1,0)-InterDepot*.

A Tabela 5.4 resume quais são os operadores utilizados em cada variante de problema durante as fases de perturbação e de busca local.

5.4 Experimentos Computacionais para o PRVJTMD*

Esta seção apresenta os resultados encontrados para o PRVJTMD com minimização do número de veículos usados (PRVJTMD*). São apresentados os resultados para os Grupos de instâncias I e II desse problema, descritos na Seção 5.1. São utilizados três algoritmos na resolução do problema. A Subseção 5.4.1 mostra os resultados para o Grupo I, e a Subseção 5.4.2 mostra os resultados para o Grupo II. As subseções que se seguem introduzem um estudo sobre a influência que as estruturas de vizinhança exercem na fase de busca local e na fase de perturbação dos algoritmos. A Subseção 5.4.3 avalia a influência das estruturas de vizinhança utilizadas na fase de busca local. Na Subseção 5.4.3.1, a influência dos operadores com baixa contribuição na fase de busca local é analisada. Em seguida, na Subseção 5.4.4 verificou a influência das vizinhanças utilizadas na fase de perturbação. Na Subseção 5.4.5 é conduzida uma verificação da influência nos resultados ao se eliminar simultaneamente os operadores com baixa contribuição da perturbação e busca local.

5.4.1 Resultados do Grupo I para o PRVJTMD*

Os resultados obtidos pelo SGVNSBLA (Algoritmo 18) foram comparados com os resultados dos algoritmos SGVNS (Algoritmo 23) e BGVNS (Algoritmo 16). Os operadores utilizados na geração das estruturas de vizinhanças na perturbação e na busca local definidas para serem utilizadas pelo SGVNS e BGVNS são as mesmas definidas para o SGVNSBLA na Seção 5.3.

Na Tabela 5.5 são mostrados os resultados alcançados por esses algoritmos ao resolver o PRVJTMD* para as instâncias do Grupo I. Para cada algoritmo, as colunas $|\mathcal{U}|$ e “Distância” mostram os melhores resultados encontrados pelos algoritmos no tocante, respectivamente, ao número de veículos utilizados e à distância total percorrida. A coluna “Tempo” relata o tempo médio de execução encontrado por esses algoritmos em minutos. Na linha “Soma” são mostradas as somas totais dos resultados encontrados. Na linha “Porcentagem” são mostradas as diferenças percentuais entre os resultados dos algoritmos. Essa diferença é calculada pelos valores em “Soma”, considerando os resultados obtidos pelo SGVNSBLA como referência. Nas duas últimas linhas são mostrados o tempo máximo de execução e a quantidade de execuções, ambos por instância, para cada algoritmo. Esses valores seguem conforme definidos na Seção 5.2.

Ao se comparar os três algoritmos, observa-se que o SGVNSBLA se comportou melhor que os algoritmos SGVNS e BGVNS nos três itens listados, ou seja:

Tabela 5.5: Resultados para o PRVJTMD* para SGVNSBLA, SGVNS e BGVNS referentes às instâncias do Grupo I.

Dados das Instâncias					SGVNSBLA			SGVNS			BGVNS		
Nome	$ \mathcal{C} $	$ \mathcal{D} $	$ \mathcal{K}_d $	$ \mathcal{K} $	$ \mathcal{U} $	Distância	Tempo	$ \mathcal{U} $	Distância	Tempo	$ \mathcal{U} $	Distância	Tempo
pr01	48	4	2	8	5	1190,00	0,26	5	1198,82	0,40	5	1197,91	0,35
pr02	96	4	3	12	9	2082,10	1,48	9	2029,35	2,53	9	2021,23	2,75
pr03	144	4	4	16	12	2725,56	3,79	13	2676,56	5,42	13	2758,97	6,64
pr04	192	4	5	20	17	3558,48	8,20	17	3816,50	8,66	16	3848,25	9,51
pr05	240	4	6	24	21	3789,84	17,58	21	3769,81	12,91	21	3899,95	16,37
pr06	288	4	7	28	26	4651,68	28,93	26	4897,11	26,91	26	4929,69	60,00
pr07	72	6	2	12	7	1527,00	0,68	7	1639,05	1,18	7	1734,83	0,53
pr08	144	6	3	18	12	2511,62	4,41	12	2452,78	6,96	12	2732,75	7,42
pr09	216	6	4	24	17	3431,24	11,04	18	3287,81	15,07	18	3389,11	23,79
pr10	288	6	5	30	25	4671,04	22,66	25	4580,05	17,38	26	4532,35	30,94
pr11	48	4	1	4	4	1032,98	0,19	4	1012,47	4,08	4	1005,73	0,44
pr12	96	4	2	8	8	1602,65	2,04	8	1592,03	10,82	8	1664,28	3,87
pr13	144	4	3	12	11	2274,40	5,21	11	2280,04	7,74	11	2865,31	1,98
pr14	192	4	4	16	15	2721,26	10,25	15	2745,77	12,99	15	2793,73	21,78
pr15	240	4	5	20	20	3171,43	19,23	20	3280,77	14,40	20	3325,56	5,92
pr16	288	4	6	24	22	3808,61	30,09	23	3950,76	22,21	23	4079,80	26,55
pr17	72	6	1	6	6	1253,21	0,59	6	1255,93	9,33	6	1254,01	0,50
pr18	144	6	2	12	12	2040,39	4,49	12	1976,29	9,36	11	2348,53	3,57
pr19	216	6	3	18	16	2681,81	13,72	16	2658,08	17,20	16	2825,88	18,35
pr20	288	6	4	24	24	3864,18	29,86	24	3905,85	27,60	24	3899,81	43,96
Soma				336	289	54589,48	214,70	292	55005,83	233,15	291	57107,68	285,21
Porcentagem(%)								+1,04	+0,76	+8,59	+0,69	+4,61	+32,85

Tempo de execução máximo para cada instância $maxTempo = 60$ minutos
 Execuções por instância: 30

(i) verificou-se uma redução de 13,99% no número de veículos utilizados, segundo o SGVNSBLA, contra uma redução de 13,10% obtida pelo SGVNS e 13,39% obtida pelo BGVNS, que utilizaram 3 e 2 veículos a mais, respectivamente. Nos resultados alcançados pelo SVNS, houve aumento de 0,76% na distância total percorrida e de 8,59% no tempo de execução. Em relação ao BGVNS, houve aumento de 4,61% na distância total percorrida e 32,85% no tempo de execução. Os três algoritmos empataram no número de veículos utilizados em 14 das 20 instâncias analisadas. Em 8 dos 14 empates, no entanto, o SGVNSBLA apresentou melhores resultados no que diz respeito à distância; o SGVNS venceu em 4 das 14; o BGVNS ganhou nas duas instâncias restantes. Quanto ao tempo de execução, em 7 das 14 instâncias com empate no número de veículos, o SGVNSBLA apresentou melhores resultados; o SGVNS venceu em outras 3; e o BGVNS venceu nas quatro restantes. No conjunto total das 20 instâncias, o SGVNSBLA superou os demais algoritmos em relação ao número de veículos utilizados em 3 delas; o BGVNS superou em 2 instâncias; em 1 instância, o SGVNSBLA e o SGVNS empataram, mas superaram o BGVNS.

A análise estatística dos resultados envolvendo os três métodos foi realizada no que diz respeito ao número de veículos utilizados e à distância total percorrida. Para normalizar os dados, considerou-se a diferença entre as medianas e os resultados

encontrados para cada instância, na forma:

$$Gap_{veiculo} = \frac{Mediana_{veiculo} - Quantidade_{veiculo}}{Mediana_{veiculo}} \quad (5.1)$$

$$Gap_{distancia} = \frac{Mediana_{distancia} - Distancia}{Mediana_{distancia}} \quad (5.2)$$

Usando-se a Equação (5.1), foi obtida a diferença entre a mediana e a quantidade de veículos encontrada em cada execução. A diferença entre as distâncias encontradas e a mediana foi calculada usando a Equação (5.2).

A verificação da aleatoriedade dos dados se deu através do teste de Durbin-Watson (Montgomery e Runger, 2016). Ele é utilizado para avaliar a correlação entre os dados de um conjunto e segue uma distribuição normal com média 2. Verifica-se que não há correlação quando o teste retorna o valor 2, caso o valor esteja no intervalo $[0, 2[$ existe correlação positiva entre eles e, no intervalo $]2, 4]$, a correlação é negativa. A aleatoriedade dos dados é garantida quando o valor retornado encontra-se próximo ao valor 2 (Montgomery e Runger, 2016). A normalidade dos dados foi verificada através do teste de Shapiro-Wilk (Montgomery e Runger, 2016). Nesse teste, a hipótese nula (H_0), sustenta a suposição de que os dados são normais, o que é verificado através do p -valor retornado pelo teste. Para um nível significância é de 5%, se o p -valor $\leq 0,05$, então H_0 é aceita, caso contrário, não há evidências estatísticas para se aceitá-la (Montgomery e Runger, 2016). A metodologia através do p -valor também é utilizada para a avaliação da homocedasticidade, com o teste de Fligner-Killeen (Montgomery e Runger, 2016) e, para o teste não paramétrico de Kruskal-Wallis (Montgomery e Runger, 2016), para a verificação da existência de diferenças estatísticas entre os algoritmos. Esse último foi escolhido por não requerer suposições sobre a normalidade (Montgomery e Runger, 2016).

A Tabela 5.6 mostra os p -valores obtidos para os testes de hipóteses relativos à aleatoriedade, normalidade e homoscedasticidade. Os testes foram implementados em Python3, com bibliotecas estatísticas da linguagem.

Tabela 5.6: p -valores para testes de hipóteses relativos a aleatoriedade, normalidade e homocedasticidade para o Grupo I com SGVNSBLA, SGVNS e BGVNS.

SGVNSBLA versus	Medida de performance	Testes estatísticos			
		Durbin-Watson	Shapiro-Wilk	Fligner-Killeen	Kruskal-Wallis
SGVNS	$Gap_{veiculo}$	1,95	4,67 e-38	0,06	0,02
	$Gap_{distancia}$	2,02	5,01 e-18	0,29	0,97
BGVNS	$Gap_{veiculo}$	1,95	4,32 e-31	0,19	2,34 e-19
	$Gap_{distancia}$	1,94	5,26 e-15	0,18	9,96 e-26

Considerando o nível de significância de 5%, a aleatoriedade e homocedasticidade dos dados foram garantidas pelos teste de de Durbin-Watson e Fligner-Killeen, respectivamente. Através do teste de Shapiro-Wilk, foi verificado que a normalidade

dos dados não foi garantida, por esse motivo, o teste para análise de variância de um fator não paramétrico Kruskal-Wallis foi aplicado. No teste formulado, a hipótese nula (H_0) considera que não há diferenças estatísticas entre os resultados encontrados pelos métodos e, como hipótese alternativa (H_1), há diferença entre os resultados encontrados. Foram verificadas as diferenças entre SGVNSBLA $\times$ SGVNS e SGVNSBLA $\times$ BGVNS. A partir dos testes efetuados, com 95% de confiança, há evidências para refutar H_0 , ou seja, esses testes estatísticos mostraram que há diferenças nos resultados encontrados pelos métodos, relativos ao número de veículos utilizados. Ao analisar os p -valores para a distância entre SGVNSBLA $\times$ SGVNS, existe uma forte relação entre os resultados encontrados e, portanto, há evidência estatística para aceitar H_0 . Os resultados encontrados entre SGVNSBLA $\times$ BGVNS, com 95% de confiança, mostraram que há evidências estatísticas para refutar H_0 e, assim, aceitar H_1 . Nesse caso, conclui-se que existem diferenças nos resultados sobre a distância percorrida para os resultados encontrados entre os dois métodos.

Ao aprofundar a comparação entre os três métodos, foi realizada uma análise estatística envolvendo apenas as ocorrências com empate na quantidade de veículos. A Tabela 5.7 mostra os resultados considerando-se a medida de desempenho $Gap_{distance}$, para o nível de significância de 5%.

Tabela 5.7: p -valores para testes de hipóteses de aleatoriedade, normalidade e homocedasticidade para o Grupo I, considerando apenas as situações em que houve empate no número de veículos utilizados.

SGVNSBLA versus	Medida de desempenho	Teste			
		Durbin-Watson	Shapiro-Wilk	Fligner-Killeen	Kruskal-Wallis
SGVNS	$Gap_{distance}$	1,98	7,04 e-17	0,76	0,90
BGVNS	$Gap_{distance}$	2,01	4,69 e-14	0,15	7,44 e-16

Através do teste de Kruskal-Wallis para SGVNSBLA $\times$ SGVNS, não houve evidência estatística que demonstrasse diferenças nos valores de distância total percorrida encontrados. O mesmo não acontece ao se avaliar o SGVNSBLA $\times$ BGVNS. Com o teste de Kruskal-Wallis, foi possível concluir que há diferenças estatísticas entre os resultados encontrados pelos algoritmos para os valores de distâncias totais percorridas encontrados.

Conforme aponta a revisão bibliográfica apresentada na Seção 2.2.1, não foram encontrados na literatura trabalhos que abordem o problema de roteamento de veículos com janela de tempo e múltiplos depósitos com minimização do número de veículos usados, aqui denominado PRVJTMD*. Este é um problema, segundo o conhecimento do autor desta tese, originalmente introduzido em Bezerra et al. (2023). Assim, não é possível realizar uma validação dos resultados aqui expostos frente a quaisquer outros. Optou-se, então por realizar a comparação entre os resultados alcançados pelo SGVNSBLA na solução do PRVJTMD* frente aos melhores resultados encontrados

Tabela 5.8: Resultados do algoritmo SGVNSBLA para o PRVJTMD* sobre as instâncias do Grupo I em comparação com os resultados do algoritmo HGSADC para o PRVJTMD.

Dados das Instâncias					HGSADC		SGVNSBLA					
Nome	$ \mathcal{C} $	$ \mathcal{D} $	$ \mathcal{K}_d $	$ \mathcal{K} $	Distância	Tempo	$ \mathcal{U} $	Distância	Tempo	Tempo ajustado	$ \mathcal{U} - \mathcal{K} $	Porcentagem
pr01	48	4	2	8	1074,12	0,31	5	1190,00	0,26	0,20	-3	-37,50%
pr02	96	4	3	12	1762,21	1,15	9	2082,10	1,48	1,14	-3	-25,00%
pr03	144	4	4	16	2373,65	1,75	12	2725,56	3,79	2,92	-4	-25,00%
pr04	192	4	5	20	2815,48	5,89	17	3558,48	8,20	6,31	-3	-15,00%
pr05	240	4	6	24	2964,65	8,68	21	3789,84	17,58	13,53	-3	-12,50%
pr06	288	4	7	28	3588,78	13,43	26	4651,68	28,93	22,27	-2	-7,14%
pr07	72	6	2	12	1418,22	0,51	7	1527,00	0,68	0,52	-5	-41,67%
pr08	144	6	3	18	2096,73	2,39	12	2511,62	4,41	3,39	-6	-33,33%
pr09	216	6	4	24	2712,56	5,20	17	3431,24	11,04	8,50	-7	-29,17%
pr10	288	6	5	30	3465,92	15,22	25	4671,04	22,66	17,44	-5	-16,67%
pr11	48	4	1	4	1005,73	0,51	4	1032,98	0,19	0,15	0	0,00%
pr12	96	4	2	8	1464,5	1,68	8	1602,65	2,04	1,57	0	0,00%
pr13	144	4	3	12	2001,83	2,94	11	2274,40	5,21	4,01	-1	-8,33%
pr14	192	4	4	16	2195,33	6,55	15	2721,26	10,25	7,90	-1	-6,25%
pr15	240	4	5	20	2433,15	12,56	20	3171,43	19,23	14,81	0	0,00%
pr16	288	4	6	24	2836,67	15,97	22	3808,61	30,09	23,17	-2	-8,33%
pr17	72	6	1	6	1236,24	1,05	6	1253,21	0,59	0,45	0	0,00%
pr18	144	6	2	12	1788,18	3,30	12	2040,39	4,49	3,46	0	0,00%
pr19	216	6	3	18	2261,08	8,59	16	2681,81	13,72	10,56	-2	-11,11%
pr20	288	6	4	24	2993,31	22,18	24	3864,18	29,86	22,99	0	0,00%
Soma				336	44488,34	129,86	289	54589,48	214,68	165,30	-47	
Porcentagem (%)							-13,99	+22,71		+27,29	-13,99	

Tempo de execução máximo para cada instância $maxTempo = 60$ minutos.

Tempo Ajustado: Tempo ajustado para o processador usado pelo algoritmo SGVNSBLA.

Execuções por instância: 30.

na literatura para a solução do PRVJTMD clássico.

A Tabela 5.8 mostra os valores encontrados pelo algoritmo SGVNSBLA em comparação com os resultados encontrados pelo algoritmo HGSADC, apresentados em Vidal et al. (2013a). Nesta tabela, as cinco primeiras colunas correspondem aos dados das instâncias; nas colunas seis e sete são mostrados a distância total percorrida e o tempo médio de execução, em minutos, respectivamente, encontrados pelo algoritmo HGSADC. Como na Tabela 5.5, as colunas $|\mathcal{U}|$ e “Distância” mostram os melhores resultados encontrados pelo algoritmo SGVNSBLA. A coluna “Tempo” registra o tempo médio de execução. A coluna “Tempo Ajustado” mostra o tempo de execução ajustado ao processador utilizado pelo HGSADC. A penúltima coluna mostra a diferença entre o número de veículos utilizados e o número de veículos disponíveis em cada instância. Por fim, na última coluna, tem-se a diferença percentual entre o número de veículos utilizados em relação ao tamanho da frota disponível ($|\mathcal{K}|$). A linha “Soma” apresenta as somas por colunas. A linha “Porcentagem” mostra a diferença percentual entre a quantidade de veículos disponíveis ($|\mathcal{K}|$) e o utilizado ($|\mathcal{U}|$), e a diferença entre os valores alcançados pelo SGVNSBLA e HGSADC em relação à distância total percorrida e tempos encontrados. As três últimas linhas são utilizadas para relembrar parâmetros definidos para o SGVNSBLA.

O campo “Tempo ajustado” representa o valor de tempo computacional ajustado

às diferenças entre as máquinas usadas para a execução dos algoritmos avaliados. Para permitir uma comparação justa entre os resultados obtidos, foram verificados os valores de *Single Thread Rating* dos processadores utilizados pelo HGSADC e SGVNSBLA. Os valores foram obtidos através da informação *Software Passmark*, disponível em <https://www.cpubenchmark.net/>. De acordo com este *software*, os valores de *Single Thread Rating* para os processadores usados pelos algoritmos HGSADC e SGVNSBLA são 1426 e 1098, respectivamente. Portanto, como o algoritmo SGVNSBLA foi testado em uma máquina mais lenta que a usada para avaliar o algoritmo HGSADC, usamos um fator de correção ψ_1 , dado por $\psi_1 = 1098/1426 = 0,77$. Assim, o valor registrado na coluna “Tempo Ajustado” representa o tempo computacional gasto pelo HGSADC, segundo informado em Vidal et al. (2013a), ponderado pelo fator de correção ψ_1 .

A partir dos resultados descritos na Tabela 5.8, é possível notar uma redução de aproximadamente 14% no número total de veículos utilizados, em uma diferença total de 47 veículos. Essa redução ocorre de forma diferente para cada instância, variando entre 0% e 41,67%. As instâncias com janelas de tempo estreitas têm uma maior redução no número de veículos utilizados. Neste caso, entre as dez instâncias, seis apresentam uma redução superior a 20%. Por outro lado, as instâncias com janelas de tempo largas têm uma baixa redução no número de veículos utilizados, e, em seis das dez instâncias, não foi possível obter uma redução no número de veículos utilizados.

5.4.2 Resultados do Grupo II para o PRVJTMD*

A Tabela 5.9 mostra os resultados gerados pelos algoritmos SGVNSBLA, SGVNS e BGVNS para a solução do Grupo II de instâncias. Esta tabela mantém a mesma formatação descrita para a Tabela 5.5.

Os resultados mostram que o algoritmo SGVNSBLA superou os algoritmos SGVNS e BGVNS nos três itens avaliados. O total de veículos disponíveis para o Grupo II de instâncias é 1960 veículos. Os resultados encontrados pelo algoritmo SGVNSBLA proporcionaram uma redução de 457 veículos (correspondendo a 23,32% do total de veículos disponível na frota do Grupo II) no número total de veículos utilizados. A redução obtida pelo algoritmo SGVNS foi de 438 veículos (22,35%); no caso do algoritmo BGVNS, foi obtida uma redução de 413 veículos (21,07%). Ao se considerar os dados obtidos pelo algoritmo SGVNSBLA como referência, houve um aumento de 2,80% na distância total encontrada pelo algoritmo SGVNS, seguida de uma redução de 36,25% no tempo de execução; os resultados encontrados pelo BGVNS mostram um aumento de 6,58% na distância total percorrida, com uma redução de 57,61% no tempo de execução. Em 13 das 28 instâncias, o SGVNSBLA foi superior em relação

Tabela 5.9: Resultados para PRVJTMD* do algoritmo SGVNSBLA versus algoritmos SGVNS e BGVNS referentes as instâncias do Grupo II.

Dados das Instâncias					SGVNSBLA			SGVNS			BGVNS		
Nome	$ \mathcal{C} $	$ \mathcal{D} $	$ \mathcal{K}_d $	$ \mathcal{K} $	$ \mathcal{U} $	Distância	Tempo	$ \mathcal{U} $	Distância	Tempo	$ \mathcal{U} $	Distância	Tempo
pr11a	360	4	10	40	34	8367,96	26,52	34	8726,24	25,88	36	8965,56	13,31
pr12a	480	4	13	52	43	10203,06	79,13	43	10787,73	54,83	45	11300,13	19,22
pr13a	600	4	16	64	52	12050,67	166,16	54	12861,05	56,68	54	13161,64	53,30
pr14a	720	4	19	76	63	14430,54	257,14	64	15272,39	99,36	66	15673,55	62,52
pr15a	840	4	22	88	77	17298,62	266,35	76	16861,08	155,03	80	18407,24	91,22
pr16a	960	4	26	104	87	19526,10	300,00	90	20129,48	167,49	93	21590,78	130,60
pr17a	360	6	7	42	33	8071,20	29,81	32	7909,23	35,81	33	8425,72	16,31
pr18a	520	6	10	60	47	10905,42	67,13	48	11027,97	45,25	50	11840,88	29,97
pr19a	700	6	13	78	63	14193,26	216,85	64	15267,10	105,01	65	15564,61	46,69
pr20a	880	6	16	96	75	15825,91	276,89	76	16251,23	151,12	78	17093,54	105,50
pr21a	420	12	4	48	36	8222,74	29,20	36	8161,57	77,92	37	8143,01	75,39
pr22a	600	12	6	72	49	10648,65	128,15	48	10438,98	135,31	51	12185,14	55,44
pr23a	780	12	8	96	64	13746,12	183,50	64	13745,38	138,61	65	14089,12	97,91
pr24a	960	12	10	120	84	16890,61	242,67	85	17322,22	147,39	85	17868,87	96,16
pr11b	360	4	8	32	27	6068,24	29,91	27	5959,04	24,34	28	6280,37	16,97
pr12b	480	4	11	44	36	7739,38	39,11	36	7945,88	35,91	35	8195,71	34,23
pr13b	600	4	14	56	44	9409,75	95,18	44	9919,54	77,03	45	9821,87	49,86
pr14b	720	4	17	68	54	11283,40	244,13	56	11910,17	92,98	56	12753,49	65,50
pr15b	840	4	20	80	66	13948,48	256,86	66	14101,22	134,17	67	14412,66	94,59
pr16b	960	4	23	92	75	15364,09	290,66	76	15871,36	170,54	75	15290,21	134,04
pr17b	360	6	6	36	26	5839,02	36,87	28	5995,21	27,23	28	6328,40	16,52
pr18b	520	6	9	54	39	8061,21	65,19	40	8388,09	57,77	40	8073,65	29,44
pr19b	700	6	12	72	52	10662,51	166,90	54	11676,04	69,08	55	11819,53	56,60
pr20b	880	6	15	90	69	13743,12	264,70	71	14309,46	115,42	70	14739,72	94,74
pr21b	420	12	4	48	30	6252,96	40,45	30	6138,25	113,45	30	6396,91	64,98
pr22b	600	12	6	72	43	8567,72	72,62	43	9386,33	104,15	44	9016,95	69,82
pr23b	780	12	7	84	58	11495,28	131,89	59	11643,89	135,59	59	12329,20	63,13
pr24b	960	12	8	96	77	14398,34	202,32	78	14264,83	128,28	77	14711,38	98,89
Soma				1960	1503	323214,36	4206,29	1522	332270,96	2681,63	1547	344479,84	1782,86
Porcentagem(%)								1,26	2,80	-36,25	2,93	6,58	-57,61

Tempo de execução máximo para cada instância $maxTempo = 300$ minutos.
 Execuções por instância: 10.

ao número de veículos utilizados; nessas mesmas 13 instâncias, a distância total foi menor, quando comparada com os resultados encontrados pelos outros dois algoritmos; o SGVNS foi superior em 3 instâncias em relação à quantidade de veículos e distância percorrida; e o BGVNS foi superior em apenas 1 instância. Nas 11 instâncias restantes, houve empate no número de veículos utilizados. Dentre elas, o SGVNSBLA gerou a menor distância percorrida em 6 instâncias; o SGVNS em 4 outras; e o BGVNS em apenas uma. Em uma única instância (pr21b) houve empate no número de veículos utilizados entre os três algoritmos. Ocorreu empate no número de veículos utilizados entre SGVNSBLA e SGVNS em 9 de 11 instâncias; em dois outros, houve empate entre SGVNSBLA e BGVNS. Não houve empate para este item envolvendo apenas SGVNS e BGVNS. Em relação ao tempo de execução, em 26 das 28 instâncias o algoritmo BGVNS teve o menor tempo de execução; o SGVNSBLA teve o menor tempo de execução nas duas restantes. Através da análise do tempo de execução dos algoritmos, é possível perceber claramente uma convergência prematura do SGVNS e BGVNS. Ao se avaliar os algoritmos a partir dos três itens relatados, verificou-se o sucesso dos procedimentos adaptativos e “inteligentes” do SGVNSBLA, fato demonstrado pelos resultados obtidos.

A análise estatística dos resultados encontrados para o Grupo II com a aplicação dos algoritmos SGVNSBLA, SGVNS e BGVNS foi realizada considerando o número total de veículos utilizados e a distância total percorrida. A Tabela 5.10 apresenta os p -valores obtidos para os testes de hipóteses para aleatoriedade, normalidade, e homoscedasticidade. Como ocorreu na análise estatística dos resultados do Grupo I, a aleatoriedade e a homoscedasticidade dos dados foram garantidos a partir dos testes de Durbin-Watson e Fligner-Killeen. O teste de Shapiro-Wilk mostrou que a normalidade dos dados não foi garantida e, assim, o teste de hipótese de Kruskal-Wallis foi utilizado para verificar possíveis diferenças estatísticas entre os resultados encontrados pelo SGVNSBLA e os outros dois algoritmos. Como hipótese nula (H_0), consideramos que não há diferenças estatísticas entre os resultados encontrados pelos algoritmos empregados e, como hipótese alternativa (H_1), que há diferenças entre os resultados encontrados. Pelos resultados obtidos com a aplicação do teste de Kruskal-Wallis, há evidências estatísticas para refutar a hipótese H_0 em todos os casos, o que permite concluir que há diferenças estatísticas entre os resultados encontrados pelos três métodos.

Tabela 5.10: p -valores para testes de hipóteses em relação a aleatoriedade, normalidade e homocedasticidade para o Grupo II com SGVNSBLA, SGVNS e BGVNS.

SGVNSBLA versus	Medida de performance	Teste			
		Durbin-Watson	Shapiro-Wilk	Fligner-Killeen	Kruskal-Wallis
SGVNS	$Gap_{peiculo}$	1,95	1,02 e-09	0,01	6,35 e-21
	$Gap_{distancia}$	2,08	0,09	1,05 e-06	1,36 e-37
BGVNS	$Gap_{peiculo}$	1,92	5,60 e-08	1,43 e-07	1,52 e-57
	$Gap_{distancia}$	1,94	3,76 e-05	4,31 e-07	4,40 e-73

Da mesma forma que ocorreu para na análise do dados do Grupo I, foi realizada uma análise estatística envolvendo os casos em que ocorreram empates no número de veículos utilizados para o Grupo II. A Tabela 5.11 mostra os resultados encontrados, considerando $Gap_{distancia}$, para um nível de significância de 5%. Estes resultados mostraram que, ao se comparar SGVNSBLA $\times$ SGVNS e SGVNSBLA $\times$ BGVNS, é possível concluir, através do Teste de Kruskal-Wallis, que existem diferenças estatísticas entre os resultados apresentados por estes algoritmos em relação aos valores de distância total percorrida.

Tabela 5.11: p -valores para testes de hipóteses de aleatoriedade, normalidade e homocedasticidade para o Grupo II, considerando apenas as situações em que houve empate no número de veículos utilizados.

SGVNSBLA versus	Medida de performance	Teste			
		Durbin-Watson	Shapiro-Wilk	Fligner-Killeen	Kruskal-Wallis
SGVNS	$Gap_{distancia}$	2,09	0,02	1,90 e-03	4,01 e-09
BGVNS	$Gap_{distancia}$	2,21	0,12	0,05	2,90 e-09

Na Tabela 5.12 são mostrados os resultados encontrados pelo algoritmo SGVNS-

BLA e os melhores resultados conhecidos na literatura para o Grupo II. Os melhores resultados foram encontrados com o algoritmo HGSADC, descrito em [Vidal et al. \(2013b\)](#), para resolver o PRVJTMD clássico. Esta tabela mantém a mesma estrutura descrita para a Tabela 5.8.

Tabela 5.12: Resultados do algoritmo SGVNSBLA para PRVJTMD* para instâncias do Grupo II comparados com os resultados do algoritmo HGSADC para PRVJTMD.

Dados das Instâncias					HGSADC		SGVNSBLA					
Nome	$ \mathcal{C} $	$ \mathcal{D} $	$ \mathcal{K}_d $	$ \mathcal{K} $	Distância	Tempo	$ \mathcal{U} $	Distância	Tempo	Tempo ajustado	$ \mathcal{U} - \mathcal{K} $	Variação (%)
pr11a	360	4	10	40	6720,71	16,81	34	8367,96	26,52	20,42	-6	-15,00
pr12a	480	4	13	52	8179,80	30,00	43	10203,06	79,13	60,93	-9	-17,31
pr13a	600	4	16	64	9667,20	54,85	52	12050,67	166,16	127,94	-12	-18,75
pr14a	720	4	19	76	11124,01	65,65	63	14430,54	257,14	197,99	-13	-17,11
pr15a	840	4	22	88	13013,97	132,44	77	17298,62	266,35	205,09	-11	-12,50
pr16a	960	4	26	104	14299,87	133,63	87	19526,10	300,00	231,00	-17	-16,35
pr17a	360	6	7	42	6304,30	17,23	33	8071,20	29,81	22,95	-9	-21,43
pr18a	520	6	10	60	8308,32	44,25	47	10905,42	67,13	51,69	-13	-21,67
pr19a	700	6	13	78	10677,61	74,42	63	14193,26	216,85	166,97	-15	-19,23
pr20a	880	6	16	96	11963,91	107,37	75	15825,91	276,89	213,20	-21	-21,88
pr21a	420	12	4	48	6260,53	28,00	36	8222,74	29,20	22,48	-12	-25,00
pr22a	600	12	6	72	7985,37	76,05	49	10648,65	128,15	98,67	-23	-31,94
pr23a	780	12	8	96	9937,43	137,72	64	13746,12	183,50	141,29	-32	-33,33
pr24a	960	12	10	120	11923,72	197,17	84	16890,61	242,67	186,85	-36	-30,00
pr11b	360	4	8	32	4839,44	18,04	27	6068,24	29,91	23,03	-5	-15,63
pr12b	480	4	11	44	6063,26	29,09	36	7739,38	39,11	30,11	-8	-18,18
pr13b	600	4	14	56	7254,17	70,99	44	9409,75	95,18	73,29	-12	-21,43
pr14b	720	4	17	68	8732,29	98,92	54	11283,40	244,13	187,98	-14	-20,59
pr15b	840	4	20	80	10439,72	129,48	66	13948,48	256,86	197,78	-14	-17,50
pr16b	960	4	23	92	11483,22	170,31	75	15364,09	290,66	223,80	-17	-18,48
pr17b	360	6	6	36	4806,01	15,78	26	5839,02	36,87	28,39	-10	-27,78
pr18b	520	6	9	54	6526,72	39,45	39	8061,21	65,19	50,20	-15	-27,78
pr19b	700	6	12	72	8227,25	80,55	52	10662,51	166,90	128,51	-20	-27,78
pr20b	880	6	15	90	10325,80	150,74	69	13743,12	264,70	203,82	-21	-23,33
pr21b	420	12	4	48	4866,57	36,75	30	6252,96	40,45	31,15	-18	-37,50
pr22b	600	12	6	72	6488,50	73,28	43	8567,72	72,62	55,92	-29	-40,28
pr23b	780	12	7	84	8523,41	163,99	58	11495,28	131,89	101,55	-26	-30,95
pr24b	960	12	8	96	10890,08	298,51	77	14398,34	202,32	155,78	-19	-19,79
Soma				1960	245833,19	2491,47	1503	323214,36	4206,31	3238,78	-457	
Porcentagem (%)								+31,48		+29,99	-23,32	
Tempo de execução máximo por instância <i>maxTempo</i> : 300 minutos												
Execuções por instância: 10												

Os resultados encontrados para o Grupo II diferem daqueles mostrados para o Grupo I. Para todas as instâncias, houve redução no número de veículos utilizados, com a menor redução sendo 12,50% (instância pr15a), e a maior de 40,28% (instância pr22b). A redução total para os veículos utilizados foi de 23,32%, correspondendo a 457 veículos. Há uma maior redução percentual do número de veículos em relação à frota total do Grupo I para o Grupo II, de 13,99% para 23,32%, respectivamente. No Grupo II, no entanto, o conjunto de instâncias de janelas estreitas tem uma redução maior do que a alcançada pelo grupo de instâncias de janelas largas. Para as instâncias de janelas estreitas, quatro das 14 instâncias têm uma redução maior do que a redução da média geral; para as instâncias de janelas largas, oito das 14

instâncias têm uma redução maior que a média. Portanto, para o Grupo II, há uma menor quantidade de veículos utilizados para as janelas largas, quando comparadas às instâncias com janelas estreitas.

5.4.3 Influência das Estruturas de Vizinhanças na Busca Local

Nesta seção é analisada a influência dos diferentes operadores utilizados na geração das estruturas de vizinhança exploradas na fase de busca local pelo algoritmo SGVNS-BLA. Para avaliar o comportamento desses operadores, foi criada uma variável auxiliar, que armazena o total dos pontos recebidos por cada vizinhança, conforme descrito no Algoritmo 13, apresentado na Seção 4.5.1. Na Figura 5.1 são mostrados os percentuais em relação a cada estrutura obtida a partir desta variável. Na Figura 5.1(a) são mostrados os percentuais em relação aos resultados obtidos com as instâncias do Grupo I. Na Figura 5.1(b) são mostrados os percentuais para o Grupo II.

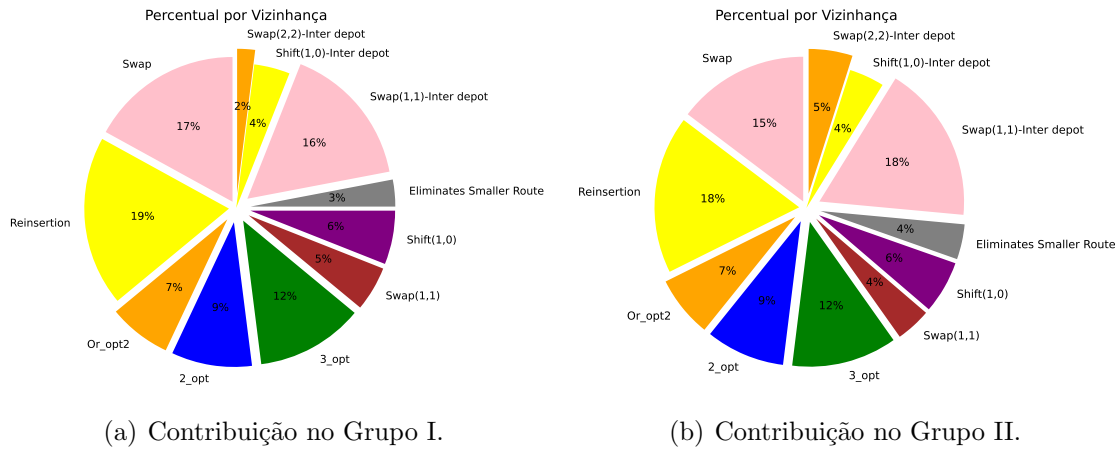


Figura 5.1: Percentual de contribuição dos operadores na busca local.

A partir dos resultados obtidos, pode-se verificar que o comportamento do algoritmo SGVNSBLA em relação aos operadores geradores das estruturas de vizinhança exploradas é o mesmo nos dois grupos. Além disso, os operadores definidos para a busca local *Reinsertion*, *Swap*, *Swap(1,1)-InterDepot* e *3-Opt* têm maior influência na redução das frotas de veículos em ambos os grupos. Juntos, esses operadores representam 64% da pontuação total distribuída para o Grupo I e 63% para o Grupo II.

Na Subseção 5.4.3.1, é analisado o comportamento do algoritmo SGVNSBLA ao eliminar os operadores que geraram as estruturas de vizinhança com menor contribuição no procedimento de busca local. Em seguida, na Subseção 5.4.4, é analisado

o comportamento do algoritmo ao se modificar a configuração dos operadores utilizados na perturbação. Na terceira parte da análise, na Subseção 5.4.5, é verificado o comportamento do SGVNSBLA ao se eliminar os operadores simultaneamente na perturbação e na busca local.

Devido ao comportamento semelhante nos dois grupos de instâncias, optou-se por executar os testes apenas com as instâncias do Grupo I. As configurações para os parâmetros do SGVNSBLA são as mesmas estabelecidas na Seção 5.2. Esses testes foram realizados no *cluster* do CEFET-MG, com processadores AMD Opteron 6376 com 2.30 GHz e sistema operacional Linux 64 bits, o *Single Thread Rating* é igual a 1165 e o fator de correção $\psi_2 = 1165/1426 = 0,82$ quando comparado com a máquina de referência usada no cálculo de ψ_1 .

5.4.3.1 Influência dos Operadores com Menor Contribuição na Fase de Busca Local

Nesta seção é investigada a influência dos operadores que têm uma contribuição inferior a 5% nos resultados do Grupo I, ou seja, *Swap(2,2)*, *Eliminates Smaller Route*, e *busca local de operadores Shift(1,0)*. Para isso, o algoritmo SGVNSBLA foi executado eliminando os operadores geradores das vizinhança com menor contribuição na fase de busca local. Como foi verificada a influência na fase de busca local, os operadores definidos para a fase de perturbação foram mantidos os mesmos.

Além do conjunto original com os onze operadores de buscas locais descritos na Seção 5.3, denominados nesta seção por $\mathcal{LSO}_{11}$, também foram analisados os desempenhos do algoritmo usando dois outros conjuntos de operadores na busca local, denominados $\mathcal{LSO}_9$ e $\mathcal{LSO}_8$. Esses conjuntos de operadores contém os nove e oito operadores mais influentes nos resultados do algoritmo proposto. Na Tabela 5.13 são mostrados os conjuntos e seus operadores.

Tabela 5.13: Subconjuntos definidos para analisar a influência dos operadores na busca local.

Conjunto	IR	<i>Shift(1,0)</i>	<i>Swap(1,1)</i>	Inter-Depósitos			<i>Eliminates Smaller Route</i>
				<i>Shift(1,0)</i>	<i>Swap(1,1)</i>	<i>Swap(2,2)</i>	
$\mathcal{LSO}_{11}$	✓	✓	✓	✓	✓	✓	✓
$\mathcal{LSO}_9$	✓	✓	✓	–	✓	–	✓
$\mathcal{LSO}_8$	✓	✓	✓	–	✓	–	–

A sigla *IR* representa o conjunto formado por todos os operadores interrota em que $IR = \{Swap, Reinsertion, Or-opt2, 2-Opt, 3-Opt\}$, descritos na 5.3. O conjunto $\mathcal{LSO}_{11}$ é o conjunto de todos os operadores de vizinhança descritos na Seção 4.2. $\mathcal{LSO}_9$ é um conjunto composto por nove operadores, tal que, $\mathcal{LSO}_9 = \mathcal{LSO}_{11} \setminus \{Shift(1,0)\text{-InterDepot}, Swap(2,2)\text{-InterDepot}\}$. O subconjunto $\mathcal{LSO}_8$ foi criado a

partir de $\mathcal{LSO}_9$, de forma que $\mathcal{LSO}_8 = \mathcal{LSO}_9 \setminus \{\textit{Eliminates Smaller Route}\}$.

Ao se utilizar os operadores definidos em $\mathcal{LSO}_8$, 6,50% dos resultados foram inviáveis. Por este motivo, os seus resultados foram eliminados das análises.

Na Tabela 5.14 são mostrados os resultados obtidos pelo SGVNSBLA com os operadores definidos em $\mathcal{LSO}_9$ e $\mathcal{LSO}_{11}$ para o Grupo I. Nesta tabela é mantida a mesma nomenclatura definida na Tabela 5.8.

Tabela 5.14: Resultados do algoritmo SGVNSBLA usando os conjuntos $\mathcal{LSO}_9$ e $\mathcal{LSO}_{11}$ de operadores de busca local em instâncias do Grupo I.

Algoritmo					HGSADC		SGVNSBLA						
Conjunto de operadores de busca local					—		$\mathcal{LSO}_9$				$\mathcal{LSO}_{11}$		
Instância	$ \mathcal{C} $	$ \mathcal{D} $	$ \mathcal{K}_d $	$ \mathcal{K} $	Distância	Tempo	$ \mathcal{U} $	Distância	Tempo	Tempo ajustado	$ \mathcal{U} $	Distância	Tempo
pr01	48	4	2	8	1074,12	0,31	5	1221,36	0,30	0,24	5	1190,00	0,20
pr02	96	4	3	12	1762,21	1,15	9	1967,90	1,64	1,34	9	2082,10	1,14
pr03	144	4	4	16	2373,65	1,75	12	2631,95	3,72	3,04	12	2725,56	2,92
pr04	192	4	5	20	2815,48	5,89	17	3527,35	6,81	5,56	17	3558,48	6,31
pr05	240	4	6	24	2964,65	8,68	21	3757,40	14,16	11,57	21	3789,84	13,53
pr06	288	4	7	28	3588,78	13,43	26	4619,54	25,33	20,69	26	4651,68	22,27
pr07	72	6	2	12	1418,22	0,51	7	1554,67	0,68	0,55	7	1527,00	0,52
pr08	144	6	3	18	2096,73	2,39	12	2584,57	4,03	3,29	12	2511,62	3,39
pr09	216	6	4	24	2712,56	5,20	17	3637,90	10,50	8,58	17	3431,24	8,50
pr10	288	6	5	30	3465,92	15,22	26	4478,62	17,60	14,38	25	4671,04	17,44
pr11	48	4	1	4	1005,73	0,51	4	1005,73	0,25	0,20	4	1032,98	0,15
pr12	96	4	2	8	1464,50	1,68	8	1631,58	2,14	1,75	8	1602,65	1,57
pr13	144	4	3	12	2001,83	2,94	11	2229,73	3,88	3,17	11	2274,40	4,01
pr14	192	4	4	16	2195,33	6,55	15	2709,21	9,80	8,01	15	2721,26	7,90
pr15	240	4	5	20	2433,15	12,56	19	3293,20	15,42	12,59	20	3171,43	14,81
pr16	288	4	6	24	2836,67	15,97	23	3684,07	21,84	17,84	22	3808,61	23,17
pr17	72	6	1	6	1236,24	1,05	6	1250,64	0,69	0,57	6	1253,21	0,45
pr18	144	6	2	12	1788,18	3,30	12	1942,50	4,42	3,61	12	2040,39	3,46
pr19	216	6	3	18	2261,08	8,59	16	2591,18	12,11	9,90	16	2681,81	10,56
pr20	288	6	4	24	2993,31	22,18	24	3785,40	24,26	19,82	24	3864,18	22,99
Soma				336	44488,34	129,86	290	54104,50		140,70	289	54589,48	165,30
Porcentagem(%)							-13,69	21,62		12,97	-13,99	22,71	27,29

Tempo de execução máximo por instância (*maxTempo*): 60 minutos

Execuções por instância: 30

Na sequência é feita a avaliação estatística dos resultados encontrados. Para avaliação dos diferentes valores retornados nas diversas instâncias, considerou-se a diferença entre os resultados encontrados por instância e sua mediana. Foram avaliados a quantidade de veículos e o tempo de execução. Para os veículos, foi utilizada a Equação (5.1), como definida na Seção 5.4.1. O gap do tempo de execução (*runtime*) é calculado conforme a expressão:

$$Gap_{runtime} = \frac{Mediana_{runtime} - Runtime}{Mediana_{runtime}} \quad (5.3)$$

A Figura 5.2 mostra os *boxplots* dos resultados encontrados pelo SGVNSBLA para o Grupo I com os operadores definidos em $\mathcal{LSO}_9$ e $\mathcal{LSO}_{11}$. Na Figura 5.2(a), os *boxplots* são relativos à quantidade de veículos encontrados e, na Figura 5.2(b), são mostrados os resultados dos tempos de execução.

Foi verificado se houve diferença estatística do algoritmo SGVNSBLA com as estruturas de vizinhanças geradas pelos dois conjuntos de operadores ao nível de

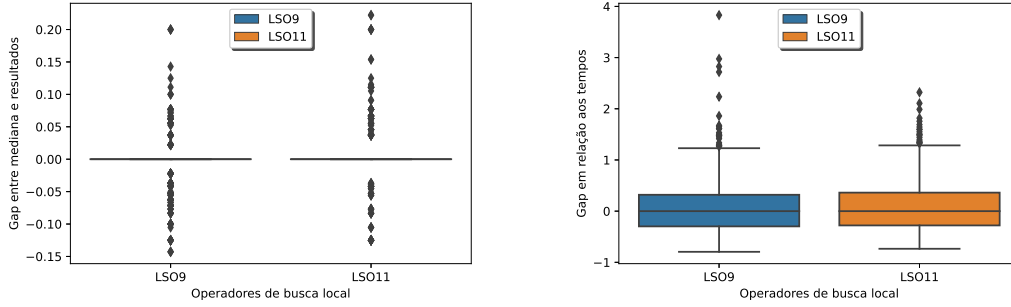
(a) *Boxplots* para a diferença do número de veículos.(b) *Boxplots* para o tempo de execução.

Figura 5.2: *Boxplots* com resultados do SGVNSBLA para Grupo I com operadores definidos em $\mathcal{LSO}_9$ e $\mathcal{LSO}_{11}$.

significância de 5%. Como hipótese nula (H_0), foi considerada a ausência de diferença estatística entre os resultados e, como hipótese alternativa (H_1), a existência de diferença estatística nos resultados.

Na Tabela 5.15 são mostrados p -valores com seus respectivos testes. Como a normalidade das amostras não foi garantida, o Teste de Kruskal-Wallis foi aplicado.

Tabela 5.15: p -valores para testes de aleatoriedade, normalidade e homocedasticidade.

Medida de desempenho	Teste			
	Durbin-Watson	Shapiro-Wilk	Fligner-Killeen	Kruskal-Wallis
$Gap_{veiculo}$	1,90	7,06 e-36	0,85	6,43 e-07
$Gap_{runtime}$	2,03	4,48 e-26	0,99	0,48

Com base no teste de Kruskal-Wallis, com 95% de confiança, não há evidências estatísticas para refutar a hipótese nula H_0 em relação ao tempo de execução. Verifica-se que não há diferença significativa no tempo de execução quando o SGVNSBLA opera a busca local com os conjuntos $\mathcal{LSO}_9$ ou $\mathcal{LSO}_{11}$. Por outro lado, em relação ao número de veículos, o p -valor encontrado foi de 6,43 e - 07, muito abaixo de 0,05. Então, com 95% de confiança, há diferença estatística nos resultados encontrados pelo SGVNSBLA quando se consideram os conjuntos de operadores $\mathcal{LSO}_9$ e $\mathcal{LSO}_{11}$.

Os dados apresentados mostram que o SGVNSBLA na configuração com os operadores definidos em $\mathcal{LSO}_{11}$ apresenta melhores resultados que sua versão com os operadores definidos pelo conjunto $\mathcal{LSO}_9$. Apesar de ambos apresentarem comportamentos parecidos, ao se observar os *boxplots*, os resultados encontrados com $\mathcal{LSO}_{11}$ produziram menos pontos distantes da mediana (*outliers*). Este resultado é melhor observado no tempo de execução na Figura 5.2(b).

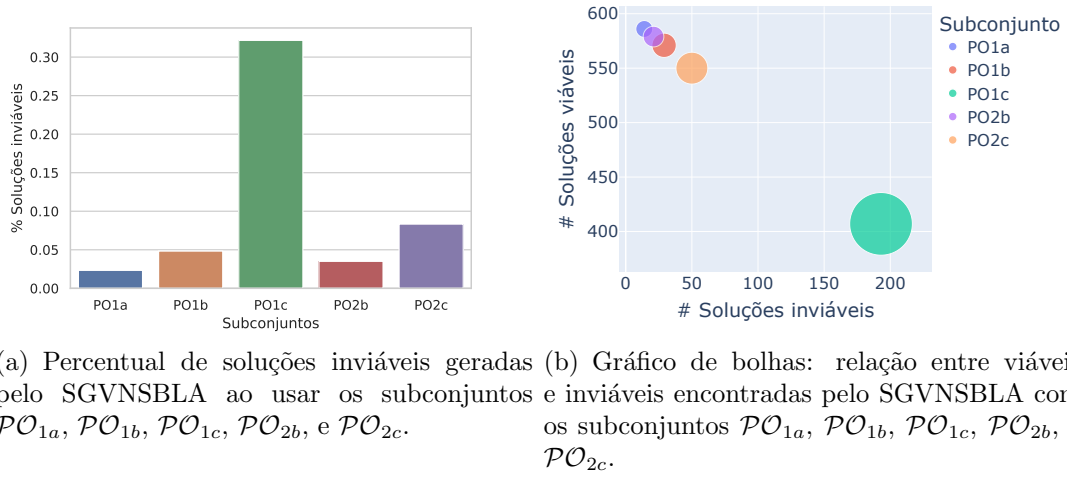


Figura 5.3: Gráficos mostrando o número de inviabilidades encontradas nos subconjuntos $\mathcal{PO}_a - \mathcal{PO}_{2c}$.

5.4.4 Influência das Vizinhanças na Perturbação

Nesta seção é analisada a influência dos operadores utilizados na geração das estruturas de vizinhança definidas na Seção 5.3 para a fase de perturbação do algoritmo SGVNSBLA aplicado à solução do PRVJTMD*. Foram mantidas as mesmas estruturas de vizinhanças geradas pelos operadores definidos no conjunto $\mathcal{LSO}_{11}$. A Tabela 5.16 mostra os subconjuntos formados pelas possíveis combinações dos três operadores utilizados na fase de perturbação. O conjunto $\mathcal{PO}_3$ é listado apenas para informar os operadores originais usados pelo algoritmo SGVNSBLA. Em cada subconjunto, são eliminados um ou dois operadores.

Tabela 5.16: Subconjuntos de operadores usados para analisar a sua influência na perturbação.

Subconjuntos	Operadores de Busca Local	Operadores de Perturbação		
	$\mathcal{LSO}_{11}$	<i>Shift(1, 0)-InterDepot</i>	<i>Swap(1,1)-InterDepot</i>	<i>Eliminates Smaller Route</i>
$\mathcal{PO}_3$	✓	✓	✓	✓
$\mathcal{PO}_{1a}$	✓	✓	–	–
$\mathcal{PO}_{1b}$	✓	–	✓	–
$\mathcal{PO}_{1c}$	✓	–	–	✓
$\mathcal{PO}_{2a}$	✓	✓	✓	–
$\mathcal{PO}_{2b}$	✓	✓	–	✓
$\mathcal{PO}_{2c}$	✓	–	✓	✓

A Figura 5.3(a) mostra a porcentagem de soluções inviáveis quando o algoritmo SGVNSBLA utiliza os subconjuntos $\mathcal{PO}_{1a}$, $\mathcal{PO}_{1b}$, $\mathcal{PO}_{1c}$, $\mathcal{PO}_{2b}$, e $\mathcal{PO}_{2c}$ como operadores de perturbação. A Figura 5.3(b) ilustra, com um gráfico de bolhas, a relação entre soluções viáveis e inviáveis, geradas por cada conjunto de operadores.

O algoritmo SGVNSBLA tem o pior desempenho com o subconjunto $\mathcal{PO}_{1c}$, encontrando mais de 30% das soluções inviáveis. Esse resultado mostra que a combinação

Tabela 5.17: Resultados do SGVNSBLA com o subconjunto $\mathcal{PO}_{2a}$.

Instância	HGSADC			$\mathcal{PO}_{2a}$			
	$ \mathcal{K} $	Distância	Tempo	$ \mathcal{U} $	Distância	Tempo	Tempo ajustado
pr01	8	1074,12	0,31	5	1223,19	0,39	0,32
pr02	12	1762,21	1,15	9	2053,28	2,52	2,06
pr03	16	2373,65	1,75	12	2733,25	5,63	4,60
pr04	20	2815,48	5,89	17	3515,26	14,82	12,11
pr05	24	2964,65	8,68	21	3722,81	23,86	19,49
pr06	28	3588,78	13,43	26	4764,20	39,41	32,20
pr07	12	1418,22	0,51	7	1605,97	1,12	0,92
pr08	18	2096,73	2,39	12	2443,41	6,92	5,66
pr09	24	2712,56	5,20	17	3310,29	16,58	13,54
pr10	30	3465,92	15,22	25	4661,87	34,75	28,39
pr11	4	1005,73	0,51	4	1005,73	0,43	0,35
pr12	8	1464,50	1,68	8	1618,91	3,01	2,46
pr13	12	2001,83	2,94	11	2432,16	6,10	4,98
pr14	16	2195,33	6,55	14	2736,13	14,14	11,55
pr15	20	2433,15	12,56	20	3168,53	24,51	20,02
pr16	24	2836,67	15,97	23	3631,15	37,99	31,04
pr17	6	1236,24	1,05	6	1262,39	1,10	0,90
pr18	12	1788,18	3,30	12	2063,66	7,75	6,33
pr19	18	2261,08	8,59	16	2760,53	20,05	16,38
pr20	24	2993,31	22,18	24	3801,72	35,80	29,24
Soma	336	44488,34	129,86	289	54514,44	296,88	242,54
Porcentagem(%)				-13,99	22,54	128,61	86,77

dos operadores de perturbação *Shift(1, 0)-InterDepot* e *Swap(1,1)-InterDepot* fornece a diversidade necessária para o algoritmo SGVNSBLA na busca de soluções viáveis no PRVJTMD*. Com os operadores definidos no subconjunto $\mathcal{PO}_{2a}$, todas as soluções geradas pelo algoritmo SGVNSBLA foram viáveis. Portanto, são mostrados na Tabela 5.17 apenas os resultados gerados com este subconjunto de operadores na perturbação.

Os resultados obtidos com os operadores definidos para a fase de perturbação em $\mathcal{PO}_{2a}$ e para a fase de busca local em $\mathcal{LSO}_{11}$ mostraram-se os mesmos quando comparados em relação à quantidade de veículos encontrados; foram superiores, no entanto, quando comparados quanto à distância total percorrida, contudo, com um valor de tempo de execução computacional 46,73% maior.

5.4.5 Influência da Eliminação Simultânea dos Operadores com Baixa Contribuição

Nesta subseção são analisados os resultados encontrados ao se eliminar simultaneamente os operadores definidos como de baixa contribuição ao SGVNSBLA nas fases de perturbação e de busca local. A Tabela 5.18 apresenta os subconjuntos formados ao se eliminar as estruturas de vizinhanças geradas pelo operadores *Shift(1,0)-InterDepot* e *Eliminates Smaller Route*. Estes operadores ofereceram contribuições inferiores à

5%, de acordo com a Figura 5.1(a).

Tabela 5.18: Subconjuntos com operadores usados para analisar a influência dos operadores nas fases de busca local e de perturbação.

Subconjuntos	Operadores de Busca Local	Operadores de Perturbação		
		<i>Shift(1, 0)-InterDepot</i>	<i>Swap(1,1)-InterDepot</i>	<i>Eliminates Smaller Route</i>
$\mathcal{LSP}\mathcal{O}_{10a}$	$\mathcal{LSO}_{11} \setminus \{Eliminates\ Smaller\ Route\}$	✓	✓	–
$\mathcal{LSP}\mathcal{O}_{10b}$	$\mathcal{LSO}_{11} \setminus \{Shift(1,0)-InterDepot\}$	–	✓	✓

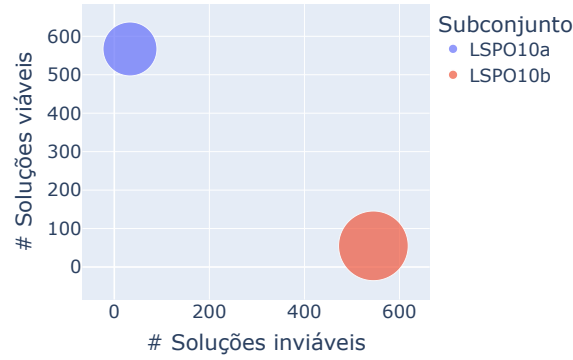


Figura 5.4: Gráfico de bolhas referente ao número de soluções viáveis e inviáveis encontradas pelo SGVNSBLA com os operadores definidos em $\mathcal{LSP}\mathcal{O}_{10a}$ e $\mathcal{LSP}\mathcal{O}_{10b}$.

A Figura 5.4 mostra o gráfico de bolhas relativo ao número de soluções viáveis e inviáveis geradas pelo SGVNSBLA usando os operadores de perturbação e busca local $\mathcal{LSP}\mathcal{O}_{10a}$ e $\mathcal{LSP}\mathcal{O}_{10b}$. O algoritmo não alcançou soluções viáveis com os operadores propostos. Esses resultados mostram que as vizinhanças geradas pelos operadores *Shift(1,0)-InterDepot* e *Eliminates Smaller Route* são essenciais para o método proposto, seja na fase de perturbação ou na fase de busca local.

5.5 Experimentos Computacionais para o PRVAJTMD e o PRVAJTMD*

Esta seção apresenta os resultados obtidos para a solução dos problemas PRVAJTMD e PRVAJTMD*. Para a solução desses dois problemas, foram utilizados os algoritmos SGVNSBLA (Algoritmo 10) e SGVNS-RVND (Algoritmo 23). Segundo o conhecimento do autor desta tese, não há instâncias especializadas para o Problema de Roteamento Aberto de Veículos com Múltiplos Depósitos e Janelas de Tempo. Assim, os testes foram aplicados para os mesmos grupos de instâncias usadas na Seção 5.4 para a solução do Problema de Roteamento fechado de Veículos PRVJTMD*, propostos em Cordeau et al. (2001) e Vidal et al. (2013a) e descritas na Seção 5.1.

Esta seção está organizada da seguinte forma. A Subseção 5.5.1 apresenta os resultados computacionais obtidos pelos algoritmos SGVNSBLA e SGVNS-RVND quando aplicados para a solução das instâncias do Grupo I; a Subseção 5.5.2, por sua vez, apresenta os resultados obtidos na aplicação dos mesmos algoritmos à solução das instâncias do Grupo II. A Subseção 5.5.3 introduz análises estatísticas referentes aos resultados encontrados e ao desempenho dos dois algoritmos.

5.5.1 Resultados do Grupo I para os problemas PRVAJTMD e PRVAJTMD*

Nesta seção são apresentados os resultados obtidos na solução de instâncias do Grupo I pelos algoritmos SGVNSBLA e SGVNS-RVND para a solução do PRVAJTMD, ou seja, visando a redução das distâncias totais percorridas, como também para a solução do PRVAJTMD*, ou seja, visando (i) a redução do número de veículos e (ii) a redução da distância total percorrida. A Subseção 5.5.1.1 mostra os resultados obtidos com o algoritmo SGVNSBLA; a Subseção 5.5.1.2 mostra os resultados encontrados pelo SGVNS-RVND e a Subseção 5.5.1.3 apresenta um comparativo entre as soluções obtidas por esses algoritmos.

5.5.1.1 Resultados obtidos pelo SGVNSBLA no Grupo I

Os melhores resultados encontrados pelo algoritmo SGVNSBLA para a solução dos problemas PRVAJTMD e PRVAJTMD* usando as instâncias do Grupo I são mostrados na Tabela 5.19. Como não foram encontrados, na literatura, trabalhos que possam ser utilizados como referência para a solução dos mesmos problemas, esses resultados são comparados com aqueles relatados por Vidal et al. (2013a) para a solução PRVJTMD usando o algoritmo HGSADC e envolvendo o mesmo grupo de instâncias.

Nesta tabela, as cinco primeiras colunas mostram as características das instâncias. A sexta coluna mostra o resultado de referência, advindo Vidal et al. (2013a), com o posto acima. As quatro próximas colunas mostram, respectivamente, os resultados relativos à solução do problema PRVAJTMD, ou seja: o valor de distância total percorrida (“Distância”); o número de veículos utilizados (“ $|\mathcal{U}|$ ”); o tempo médio de processamento computacional (“Tempo”); e a diferença entre os valores de distância total percorrida entre os resultados fornecidos pelo HGSADC e o encontrado pelo SGVNSBLA (“Redução da distância”). As colunas seguintes tratam dos resultados relativos à solução do problema PRVAJTMD*: o valor de distância total percorrida (“Distância”); o número de veículos utilizados (“ $|\mathcal{U}|$ ”); o tempo médio de processamento computacional (“Tempo”); e a diferença do número de veículos utilizados entre

Tabela 5.19: Resultados para o PRVAJTMD e o PRVAJTMD* encontrados pelo SGVNSBLA aplicado ao Grupo I.

					HGSADC	SGVNSBLA										Diferenças entre PRVAJTMD* e PRVAJTMD	
Dados das Instâncias					PRVJTMD	PRVAJTMD				PRVAJTMD*							
Nome	C	D	K _d	K	Distância	Distância	U	Tempo	Redução da distância (%)	Distância	U	Tempo	Redução dos veículos (%)	Distância	Veículos		
pr01	48	4	2	8	1074,12	894,40	8	0,13	-16,73	1148,26	5	0,11	-37,50	253,86	-3		
pr02	96	4	3	12	1762,21	1409,67	12	1,04	-20,01	1905,25	9	0,66	-25,00	495,58	-3		
pr03	144	4	4	16	2373,65	2176,59	16	1,70	-8,30	2708,86	13	1,28	-18,75	532,27	-3		
pr04	192	4	5	20	2815,48	2457,30	20	4,90	-12,72	2965,44	17	3,09	-15,00	508,14	-3		
pr05	240	4	6	24	2964,65	2700,15	24	9,36	-8,92	3560,15	21	5,15	-12,50	860,00	-3		
pr06	288	4	7	28	3588,78	3366,38	28	19,07	-6,20	4863,02	25	9,58	-10,71	1496,64	-3		
pr07	72	6	2	12	1418,22	1150,48	12	0,33	-18,88	1697,82	7	0,21	-41,67	547,34	-5		
pr08	144	6	3	18	2096,73	1880,75	18	1,99	-10,30	2786,63	12	1,20	-33,33	905,88	-6		
pr09	216	6	4	24	2712,56	2501,83	24	5,22	-7,77	3365,84	18	2,76	-25,00	864,01	-6		
pr10	288	6	5	30	3465,92	3226,73	30	14,24	-6,90	4678,28	25	5,21	-16,67	1451,55	-5		
pr11	48	4	1	4	1005,73	855,80	4	54,79	-14,91	893,53	4	0,76	-	37,73	-		
pr12	96	4	2	8	1464,50	1156,77	8	6,17	-21,01	1160,24	8	1,21	-	3,47	-		
pr13	144	4	3	12	2001,83	1807,46	12	3,87	-9,71	1937,90	11	2,53	-8,33	130,44	-1		
pr14	192	4	4	16	2195,33	2023,56	16	7,01	-7,82	2097,55	15	6,57	-6,25	73,99	-1		
pr15	240	4	5	20	2433,15	2199,44	20	14,31	-9,61	2201,06	20	8,69	-	1,62	-		
pr16	288	4	6	24	2836,67	2655,24	24	20,35	-6,40	2886,99	23	15,17	-4,17	231,75	-1		
pr17	72	6	1	6	1236,24	1075,36	6	31,59	-13,01	1093,04	6	5,99	-	17,68	-		
pr18	144	6	2	12	1788,18	1563,94	12	9,51	-12,54	1748,47	11	2,82	-8,33	184,53	-1		
pr19	216	6	3	18	2261,08	1925,18	18	8,41	-14,86	2237,48	16	6,14	-11,11	312,30	-2		
pr20	288	6	4	24	2993,31	2863,97	24	50,17	-4,32	2832,50	24	13,76	-	31,47	-		
Soma				336	44488,34	39891,00	336	264,16		48768,31	290	92,89		8877,31	-46		
Diferença percentual em relação ao PRVJTMD						-10,33	-			9,62	-13,69						
Diferença percentual entre PRVAJTMD* e PRVAJTMD														22,25	-13,69		

os resultados fornecidos pelo HGSADC e o encontrado pelo SGVNSBLA (“Redução da distância”). A linha “Soma” mostra o número total de veículos disponíveis e a distância percorrida total para o PRVJTMD; a distância total, o número de veículos utilizados e o tempo médio total gasto em todas as instâncias para o PRVAJTMD e PRVJTMD*; a diferença da distância total e número total de veículos utilizados entre os dois problemas. Para cada instância, o algoritmo foi executado 30 vezes, com $maxTempo = 60$ minutos.

A partir do valor de referência definido neste trabalho como sendo os melhores valores conhecidos para o PRVJTMD, verificou-se que, para o PRVAJTMD, houve uma redução de 20,33% na distância total encontrada, com a utilização do número total dos veículos disponíveis. Devido ao fato do veículo não retornar ao depósito de origem, houve redução na distância em todas as instâncias. Esse mesmo fato ocorreu em 9 das 20 instâncias para o caso da solução do PRVAJTMD*, com uma redução dos veículos utilizados em 15 das instâncias. Dentre as cinco que não ocorreram redução na quantidade de veículos, duas não a permitem, por possuir apenas um veículo por depósito. Houve uma redução de 13,69% na quantidade de veículos, valor muito próximo ao encontrado para o PRVJTMD*, mostrado na Tabela 5.5 (Seção 5.4.1).

5.5.1.2 Resultados obtidos pelo SGVNS-RVND no Grupo I

Na Tabela 5.20 são apresentados os resultados obtidos pela aplicação do algoritmo SGVNS-RVND para a solução de instâncias do Grupo I para os problemas PRVAJTMD e PRVAJTMD*. A tabela mantém as mesmas características da Ta-

Tabela 5.20: Resultados para PRVAJTMD e PRVAJTMD* encontrados pelo SGVNS-RVND aplicado ao Grupo I.

					HGSADC	SGVNS-RVND											
Dados das Instâncias					PRVJTMD	PRVAJTMD				PRVAJTMD*				Diferenças entre PRVAJTMD* e PRVAJTMD			
Nome	C	D	K _d	K	Distância	Distância	U	Tempo	Redução da distância (%)	Distância	U	Tempo	Redução dos veículos (%)	Distância	Veículos		
pr01	48	4	2	8	1074,12	886,48	8	0,51	-17,47	1,105,09	6	0,20	-25,00	218,61	-2		
pr02	96	4	3	12	1762,21	1,388,35	12	2,75	-21,22	1,602,58	10	1,00	-16,67	214,23	-2		
pr03	144	4	4	16	2373,65	2,157,48	16	2,95	-9,11	2,736,44	12	1,82	-25,00	578,96	-4		
pr04	192	4	5	20	2815,48	2,522,99	20	13,28	-10,39	2,644,36	18	5,78	-10,00	121,37	-2		
pr05	240	4	6	24	2964,65	2,639,74	24	14,13	-10,96	2,809,95	22	19,17	-8,33	170,21	-2		
pr06	288	4	7	28	3588,78	3,401,57	28	27,36	-5,22	3,387,00	27	20,08	-3,57	-14,57	-1		
pr07	72	6	2	12	1418,22	1,151,73	12	0,90	-18,79	1,340,38	9	0,46	-25,00	188,65	-3		
pr08	144	6	3	18	2096,73	1,852,31	18	4,85	-11,66	2,191,60	13	1,32	-27,78	339,29	-5		
pr09	216	6	4	24	2712,56	2,402,34	24	7,47	-11,44	2,935,67	19	3,83	-20,83	533,33	-5		
pr10	288	6	5	30	3465,92	3,072,93	30	32,82	-11,34	3,614,71	27	23,24	-10,00	541,78	-3		
pr11	48	4	1	4	1005,73	890,86	4	0,50	-11,42	884,18	4	0,40	-	-6,68	-		
pr12	96	4	2	8	1464,5	1,172,55	8	2,79	-19,94	1,188,69	8	2,65	-	16,14	-		
pr13	144	4	3	12	2001,83	1,717,10	12	6,92	-14,22	1,949,21	11	5,94	-8,33	232,11	-1		
pr14	192	4	4	16	2195,33	1,932,09	16	14,63	-11,99	2,069,50	15	14,27	-6,25	137,41	-1		
pr15	240	4	5	20	2433,15	2,097,23	20	25,64	-13,81	2,137,34	20	27,31	-	40,11	-		
pr16	288	4	6	24	2836,67	2,640,57	24	48,94	-6,91	2,680,17	23	48,14	-4,17	39,60	-1		
pr17	72	6	1	6	1236,24	1,070,43	6	1,09	-13,41	1,070,72	6	60,00	-	0,29	-		
pr18	144	6	2	12	1788,18	1,521,80	12	10,84	-14,90	1,757,01	11	10,70	-8,33	235,21	-1		
pr19	216	6	3	18	2261,08	1,905,24	18	17,57	-15,74	2,099,35	16	22,99	-11,11	194,11	-2		
pr20	288	6	4	24	2993,31	2,693,34	24	53,20	-10,02	2,660,01	24	24,58	-	-33,33	-		
Soma				336	44488,34	39117,13	336	289,14		42863,96	301	293,88		3746,83	-35		
Diferença percentual em relação ao PRVJTMD						-12,07	-			-3,65	-10,42						
Diferença percentual entre PRVAJTMD* e PRVAJTMD														-9,58	-10,42		

bela 5.19.

Os resultados mostram uma redução de aproximadamente 12,07% na distância total percorrida para o PRVAJTMD para todas as instâncias avaliadas quando comparados com o PRVJTMD. Os resultados do PRVAJTMD* mostram que o número de veículos utilizados sofreu uma redução de 10,42% quando comparado com o total de veículos disponíveis. Dos 336 veículos disponíveis no Grupo I, 301 foram utilizados no caso da solução encontrada pelo algoritmo SGVNS-RVND. Mesmo não sendo o objetivo associado ao problema, a distância total do Grupo I diminuiu 3,65%, quando comparada com a solução do PRVJTMD via HGSADC.

5.5.1.3 Comparativo entre os resultados dos algoritmos SGVNSBLA e SGVNS-RVND no Grupo I

A Tabela 5.21 mostra a comparação entre os resultados encontrados pelos algoritmos SGVNSBLA e SGVNS-RVND.

A diferença entre os valores de distância total percorrida apresentados na coluna “SGVNS-RVND - SGVNSBLA” mostra que o SGVNS-RVND teve desempenho superior ao SGVNSBLA ao encontrar menores distâncias a serem percorridas. Em 75% das instâncias avaliadas o SGVNS-RVND obteve menores distâncias, com uma diferença total de 1,98%.

No que diz respeito ao problema PRVAJTMD*, ou seja, para o problema de roteamento aberto de veículos com janela de tempo e múltiplos depósitos visando a minimização do número de veículos usados, a Tabela 5.22 mostra os resultados

Tabela 5.21: Grupo I: comparativo entre os resultados dos algoritmos SGVNSBLA e SGVNS-RVND para a solução do PRVAJTMD.

Instância					SGVNSBLA			SGVNS-RVND			(SGVNS-RVND) - (SGVNSBLA)	
	$ C $	$ D $	$ K_d $	$ K $	Distância	$ U $	Tempo	Distância	$ U $	Tempo	Distância	Veículos
pr01	48	4	2	8	894,40	8	0,13	886,48	8	0,51	-7,92	-
pr02	96	4	3	12	1409,67	12	1,04	1388,35	12	2,75	-21,32	-
pr03	144	4	4	16	2176,59	16	1,7	2157,48	16	2,95	-19,11	-
pr04	192	4	5	20	2457,30	20	4,9	2522,99	20	13,28	65,69	-
pr05	240	4	6	24	2700,15	24	9,36	2639,74	24	14,13	-60,41	-
pr06	288	4	7	28	3366,38	28	19,07	3401,57	28	27,36	35,19	-
pr07	72	6	2	12	1150,48	12	0,33	1151,73	12	0,90	1,25	-
pr08	144	6	3	18	1880,75	18	1,99	1852,31	18	4,85	-28,44	-
pr09	216	6	4	24	2501,83	24	5,22	2402,34	24	7,47	-99,49	-
pr10	288	6	5	30	3226,73	30	14,24	3072,93	30	32,82	-153,80	-
pr11	48	4	1	4	855,80	4	54,79	890,86	4	0,50	35,06	-
pr12	96	4	2	8	1156,77	8	6,17	1172,55	8	2,79	15,78	-
pr13	144	4	3	12	1807,46	12	3,87	1717,10	12	6,92	-90,36	-
pr14	192	4	4	16	2023,56	16	7,01	1932,09	16	14,63	-91,47	-
pr15	240	4	5	20	2199,44	20	14,31	2097,23	20	25,64	-102,21	-
pr16	288	4	6	24	2655,24	24	20,35	2640,57	24	48,94	-14,67	-
pr17	72	6	1	6	1075,36	6	31,59	1070,43	6	1,09	-4,93	-
pr18	144	6	2	12	1563,94	12	9,51	1521,80	12	10,84	-42,14	-
pr19	216	6	3	18	1925,18	18	8,41	1905,24	18	17,57	-19,94	-
pr20	288	6	4	24	2863,97	24	50,17	2693,34	24	53,20	-170,63	-
Soma					39891,00	336	264,16	39117,13	336	289,14	-773,87	-
Diferença Percentual entre SGVNS-RVND e SGVNSBLA											-1,98	

encontrados pelos algoritmos SGVNSBLA e SGVNS-RVND para o PRVAJTMD*.

O algoritmo SGVNSBLA produziu os melhores resultados para o número de veículos usados quando soluciona o Grupo I de instâncias. O algoritmo obteve uma solução com uma diferença total de 11 (3,65%) veículos a menos em relação ao resultado encontrado pelo algoritmo SGVNS-RVND e uma redução (ou economia) na frota total disponível de 46 veículos (13,69%).

5.5.2 Resultado do Grupo II para o PRVAJTMD e o PRVAJTMD*

Nesta seção serão apresentados os resultados obtidos pelos algoritmos SGVNSBLA e SGVNS-RVND aplicados à solução de instâncias do Grupo II dos problemas PRVAJTMD e PRVAJTMD*. A Seção 5.5.2.1 mostra os resultados obtidos com o algoritmo SGVNSBLA; a Seção 5.5.2.2 mostra os resultados encontrados pelo SGVNS-RVND e, para finalizar, a Seção 5.5.2.3 introduz uma análise comparativa entre ambos os algoritmos.

Nos testes efetuados com as instâncias do Grupo II para estes dois algoritmos, ocorreram 20 execuções, com tempo máximo de 5 horas para cada instância.

5.5.2.1 Resultados obtidos pelo SGVNSBLA para o Grupo II

Os resultados encontrados pelo algoritmo SGVNSBLA para a solução do Grupo II de instâncias no caso dos problemas PRVAJTMD e PRVAJTMD* são mostrados na

Tabela 5.22: Comparativo entre SGVNSBLA e SGVNS-RVND para o PRVAJTMD* sobre resultados do Grupo I.

Instância					SGVNSBLA			SGVNS-RVND			SGVNS-RVND – SGVNSBLA	
	$ C $	$ D $	$ K_d $	$ K $	Distância	$ U $	Tempo	Distância	$ U $	Tempo	Distância	Veículos
pr01	48	4	2	8	1148,26	5	0,11	1105,09	6	0,20	-43,17	1
pr02	96	4	3	12	1905,25	9	0,66	1602,58	10	1,00	-302,67	1
pr03	144	4	4	16	2708,86	13	1,28	2736,44	12	1,82	27,58	-1
pr04	192	4	5	20	2965,44	17	3,09	2644,36	18	5,78	-321,08	1
pr05	240	4	6	24	3560,15	21	5,15	2809,95	22	19,17	-750,20	1
pr06	288	4	7	28	4863,02	25	9,58	3387,00	27	20,08	-1476,02	2
pr07	72	6	2	12	1697,82	7	0,21	1340,38	9	0,46	-357,44	2
pr08	144	6	3	18	2786,63	12	1,20	2191,60	13	1,32	-595,03	1
pr09	216	6	4	24	3365,84	18	2,76	2935,67	19	3,83	-430,17	1
pr10	288	6	5	30	4678,28	25	5,21	3614,71	27	23,24	-1063,57	2
pr11	48	4	1	4	893,53	4	0,76	884,18	4	0,40	-9,35	–
pr12	96	4	2	8	1160,24	8	1,21	1188,69	8	2,65	28,45	–
pr13	144	4	3	12	1937,90	11	2,53	1949,21	11	5,94	11,31	–
pr14	192	4	4	16	2097,55	15	6,57	2069,50	15	14,27	-28,05	–
pr15	240	4	5	20	2201,06	20	8,69	2137,34	20	27,31	-63,72	–
pr16	288	4	6	24	2886,99	23	15,17	2680,17	23	48,14	-206,82	–
pr17	72	6	1	6	1093,04	6	5,99	1070,72	6	60,00	-22,32	–
pr18	144	6	2	12	1748,47	11	2,82	1757,01	11	10,70	8,54	–
pr19	216	6	3	18	2237,48	16	6,14	2099,35	16	22,99	-138,13	–
pr20	288	6	4	24	2832,50	24	13,76	2660,01	24	24,58	-172,49	–
Soma				336	48768,31	290	92,89	42863,96	301	293,88	-5904,35	11
Diferença Percentual entre SGVNS-RVND e SGVNSBLA											-12,11	3,79

Tabela 5.23. Esta tabela mantém a mesma formatação descrita para a Tabela 5.19.

Em relação ao problema de referência (PRVJTMD) e sua solução de referência, advinda de Vidal et al. (2013b), obteve-se um decréscimo de 15,98% na distância total percorrida quando o algoritmo SGVNSBLA é utilizado para encontrar melhores distâncias, acompanhado de uma redução de 12,09% no número de veículos utilizados. No processo de redução dos veículos, houve redução de 25,15% na quantidade total de veículos usada. O tempo total gasto pelo algoritmo na resolução do PRAJTMD* é 52,94% menor que o tempo total gasto na solução do PRAJTMD. Neste caso, foi observado que o algoritmo tende a convergir para uma região com uma quantidade menor de veículos com os operadores de redução de veículos; em seguida, ocorre uma exploração nesta vizinhança por rotas menores. Pequenas variações na distância ocorrem em grande quantidade, o que explica a grande variação no tempo entre os dois problemas tratados.

5.5.2.2 Resultados obtidos pelo SGVNS-RVND no Grupo II

A Tabela 5.24 mostra os resultados encontrados pelo algoritmo SGVNS-RVND para a solução de instâncias do Grupo II dos problemas PRVAJTMD e PRVAJTMD*. Esta tabela mantém a mesma formatação descrita para a Tabela 5.19.

Em todas as instâncias avaliadas houve redução na distância total percorrida e na quantidade de total de veículos utilizados. No caso do PRVAJTMD, houve redução de 10,43% na distância total em relação ao PRVJTMD. Os resultados encontrados pelo

Tabela 5.23: Resultados para o PRAJTMD e o PRVAJTMD* encontrados pelo SGVNSBLA aplicado ao Grupo II.

					HGSADC	SGVNSBLA								Diferenças entre PRVAJTMD* e PRVAJTMD	
					PRVJTMD	PRVAJTMD				PRVAJTMD*					
Instância	C	D	K _d	K	Distância	Distância	U	Tempo	Redução da distância (%)	Distância	U	Tempo	Redução dos veículos (%)	Distância	Veículos
pr11a	360	4	10	40	6720,71	5557,53	39	24,66	-17,31	6139,64	31	13,75	-22,50	582,11	-8
pr12a	480	4	13	52	8179,80	6700,00	49	60,63	-18,09	7745,77	41	26,14	-21,15	1045,77	-8
pr13a	600	4	16	64	9667,20	8143,56	58	104,92	-15,76	9212,94	51	45,88	-20,31	1069,38	-7
pr14a	720	4	19	76	11124,01	9160,79	69	173,77	-17,65	10312,82	61	82,44	-19,74	1152,03	-8
pr15a	840	4	22	88	13013,97	10526,67	82	250,22	-19,11	12191,06	71	122,24	-19,32	1664,39	-11
pr16a	960	4	26	104	14299,87	11784,71	92	292,00	-17,59	14364,60	82	139,67	-21,15	2579,89	-10
pr17a	360	6	7	42	6304,30	5353,51	41	23,90	-15,08	7445,35	31	9,63	-26,19	2091,84	-10
pr18a	520	6	10	60	8308,32	6939,04	56	77,48	-16,48	9675,01	45	23,61	-25,00	2735,97	-11
pr19a	700	6	13	78	10677,61	8681,25	69	147,80	-18,70	10774,15	59	53,74	-24,36	2092,90	-10
pr20a	880	6	16	96	11963,91	10137,58	87	236,47	-15,27	12331,83	73	83,91	-23,96	2194,25	-14
pr21a	420	12	4	48	6260,53	5612,94	44	40,79	-10,34	7221,59	34	13,35	-29,17	1608,65	-10
pr22a	600	12	6	72	7985,37	7294,24	66	86,77	-8,65	10306,89	47	27,67	-34,72	3012,65	-19
pr23a	780	12	8	96	9937,43	8886,52	77	163,62	-10,58	11858,31	62	53,72	-35,42	2971,79	-15
pr24a	960	12	10	120	11923,72	10787,06	98	238,07	-9,53	13695,34	83	89,73	-30,83	2908,28	-15
pr11b	360	4	8	32	4839,44	3995,45	32	35,49	-17,44	4577,75	27	28,65	-15,63	582,30	-5
pr12b	480	4	11	44	6063,26	5095,10	40	84,50	-15,97	5632,93	35	57,17	-20,45	537,83	-5
pr13b	600	4	14	56	7254,17	6232,23	54	153,91	-14,09	6952,11	44	71,68	-21,43	719,88	-10
pr14b	720	4	17	68	8732,29	7131,76	60	209,28	-18,33	10162,23	54	118,31	-20,59	3030,47	-6
pr15b	840	4	20	80	10439,72	8351,09	73	248,08	-20,01	9946,15	66	155,47	-17,50	1595,06	-7
pr16b	960	4	23	92	11483,22	9371,46	77	297,92	-18,39	11403,05	73	224,61	-20,65	2031,59	-4
pr17b	360	6	6	36	4806,01	4125,94	34	33,30	-14,15	5192,61	27	16,99	-25,00	1066,67	-7
pr18b	520	6	9	54	6526,72	5428,54	48	81,24	-16,83	6057,07	40	32,12	-25,93	628,53	-8
pr19b	700	6	12	72	8227,25	6910,97	61	169,08	-16,00	7623,41	53	85,72	-26,39	712,44	-8
pr20b	880	6	15	90	10325,80	8314,45	73	268,08	-19,48	9028,76	69	122,98	-23,33	714,31	-4
pr21b	420	12	4	48	4866,57	4469,11	40	47,34	-8,17	6049,57	30	19,23	-37,50	1580,46	-10
pr22b	600	12	6	72	6488,50	5564,69	52	139,76	-14,24	8077,73	44	44,96	-38,89	2513,04	-8
pr23b	780	12	7	84	8523,41	7194,30	67	200,33	-15,59	9437,63	58	82,04	-30,95	2243,33	-9
pr24b	960	12	8	96	10890,08	8798,53	85	282,78	-19,21	10768,65	76	118,00	-20,83	1970,12	-9
Soma			1960		245833,19	206549,02	1723	4172,19		254184,95	1467	1963,41		47635,93	-256
Diferença percentual em relação ao PRVJTMD						-15,98	-12,09			3,40	-25,15				
Diferença percentual entre PRVAJTMD* e PRVAJTMD														23,06	-14,86

algoritmo SGVNS-RVND para o PRVAJTMD* mostraram uma redução de 24,13% no total de veículos e 6,14% na distância total em comparação com os resultados da solução de referência do PRVJTMD. Dentre esses resultados, a menor redução no número dos veículos foi de 15,87%, para a instância pr11b, e a maior foi de 38,89%, para a instância pr22b. A diferença na solução dos dois problemas ocorreu de forma acentuada na quantidade dos veículos, com uma diminuição de 23,59% no número total de veículos usados. Na distância total percorrida, houve uma diferença de 4,84%. A disparidade na quantidade de veículos apresenta um comportamento normal do algoritmo, devido às diferenças entre eles.

5.5.2.3 Comparativo entre SGVNSBLA e SGVNS-RVND no Grupo II

Esta seção introduz uma análise comparativa a respeito do comportamento dos dois métodos utilizados na resolução de instâncias do Grupo II para os problemas PRVAJTMD e PRVAJTMD*. A Tabela 5.25 apresenta os resultados encontrados pelos algoritmos SGVNSBLA e SGVNS-RVND para o PRVAJTMD. Na última coluna da tabela, a diferença entre os métodos é mostrada. Na última linha, essa diferença é apresentada de forma percentual.

Os resultados mostram que, para o PRVJTMD, o algoritmo SGVNSBLA teve desempenho superior ao algoritmo SGVNS-RVND. Ele encontrou melhores resultados

Tabela 5.24: Resultados para o PRVAJTMD e o PRVAJTMD* encontrados pelo SGVNS-RVND aplicado ao Grupo II.

Instância					PRVJTMD	PRVAJTMD				PRVAJTMD*				Diferenças entre PRVAJTMD* e PRVAJTMD	
	C	D	K _d	K	Distância	Distância	U	Tempo	Redução da distância (%)	Distância	U	Tempo	Redução dos veículos (%)	Distância	Veículos
pr11a	360	4	10	40	6720,71	5757,87	40	39,10	14,33	6117,39	32	29,40	20,00	359,52	-8
pr12a	480	4	13	52	8179,80	7123,59	52	66,08	12,91	7500,05	41	63,84	21,15	376,46	-11
pr13a	600	4	16	64	9667,20	8383,24	64	109,17	13,28	8506,39	51	107,59	20,31	123,15	-13
pr14a	720	4	19	76	11124,01	9780,85	76	158,81	12,07	10809,87	59	145,45	22,37	1029,02	-17
pr15a	840	4	22	88	13013,97	11544,52	88	173,51	11,29	12421,66	72	178,98	18,18	877,14	-16
pr16a	960	4	26	104	14299,87	13221,55	104	180,03	7,54	13266,47	87	179,07	16,35	44,92	-17
pr17a	360	6	7	42	6304,30	5560,62	42	27,33	11,80	6056,34	32	20,25	23,81	495,72	-10
pr18a	520	6	10	60	8308,32	7346,70	60	60,89	11,57	7577,97	46	57,38	23,33	231,27	-14
pr19a	700	6	13	78	10677,61	9328,91	78	139,97	12,63	9992,12	59	134,03	24,36	663,21	-19
pr20a	880	6	16	96	11963,91	10750,93	96	171,27	10,14	11504,57	74	160,60	22,92	753,64	-22
pr21a	420	12	4	48	6260,53	5798,82	48	35,01	7,37	6083,25	34	33,50	29,17	284,43	-14
pr22a	600	12	6	72	7985,37	7511,24	72	77,99	5,94	8139,98	50	62,03	30,56	628,74	-22
pr23a	780	12	8	96	9937,43	9427,49	96	107,70	5,13	10050,26	65	117,75	32,29	622,77	-31
pr24a	960	12	10	120	11923,72	11484,88	119	162,05	3,68	12012,55	85	166,60	29,17	527,67	-34
pr11b	360	4	8	32	4839,44	4071,32	32	40,39	15,87	4332,77	27	42,96	15,63	261,45	-5
pr12b	480	4	11	44	6063,26	5311,61	44	78,26	12,40	6055,14	35	83,00	20,45	743,53	-9
pr13b	600	4	14	56	7254,17	6457,07	56	132,25	10,99	6726,56	44	132,72	21,43	269,49	-12
pr14b	720	4	17	68	8732,29	7649,92	68	136,02	12,40	7888,94	55	186,02	19,12	239,02	-13
pr15b	840	4	20	80	10439,72	9048,86	80	179,24	13,32	9499,00	66	180,02	17,50	450,14	-14
pr16b	960	4	23	92	11483,22	10427,21	91	180,02	9,20	10149,21	74	179,94	19,57	-278,00	-17
pr17b	360	6	6	36	4806,01	4210,63	36	33,34	12,39	4590,55	27	30,14	25,00	379,92	-9
pr18b	520	6	9	54	6526,72	5772,21	54	71,29	11,56	6086,92	39	90,74	27,78	314,71	-15
pr19b	700	6	12	72	8227,25	7349,44	72	103,88	10,67	7247,77	53	149,35	26,39	-101,67	-19
pr20b	880	6	15	90	10325,80	8866,85	90	167,42	14,13	9011,14	69	176,05	23,33	144,29	-21
pr21b	420	12	4	48	4866,57	4790,29	48	26,49	1,57	5046,17	30	42,65	37,50	255,88	-18
pr22b	600	12	6	72	6488,50	6320,29	67	48,36	2,59	6173,67	44	106,93	38,89	-146,62	-23
pr23b	780	12	7	84	8523,41	7580,44	79	110,74	11,06	7796,45	59	173,26	29,76	216,01	-20
pr24b	960	12	8	96	10890,08	9212,93	94	160,76	15,40	10106,99	78	165,47	18,75	894,06	-16
Soma			1960		245833,19	220090,28	1946	2977,37		230750,15	1487	3195,72		10659,87	-459
Diferença percentual em relação ao PRVJTMD						-10,43	-0,71				-6,14	-24,13			
Diferença Percentual entre PRVAJTMD* e PRVAJTMD														4,84	-23,59

para a distância percorrida em todas as instâncias e, em apenas uma, ocorreu empate no número de veículos. Em relação ao tempo médio de processamento, o SGVNS-RVND foi 28,64% mais rápido.

A Tabela 5.26 apresenta a comparação entre os resultados encontrados pelos dois algoritmos referentes à solução de instâncias do Grupo II para o PRVAJTMD*.

A partir dessa tabela, observa-se que o algoritmo SGVNSBLA encontrou uma quantidade menor de veículos usados em 13 instâncias, empatou em 13 outras e foi pior em 2 restantes. No valor total de veículos usados, ele encontrou uma solução com vinte veículos a menos. O tempo total gasto foi 38,56% menor que o obtido pelo SGVNS-RVND.

Portanto, como apontam as tabelas e a análise realizada, o algoritmo SGVNSBLA mostrou-se superior em termos de resultados obtidos nos dois problemas avaliados, em especial no problema PRVAJTMD*. Neste último caso, além de encontrar uma solução com vinte veículos a menos, o tempo de processamento também mostrou-se muito inferior ao tempo de processamento do SGVNS-RVND.

5.5.3 Análise Estatística - PRVAJTMD e PRVAJTMD*

Nesta seção, o desempenho dos métodos SGVNSBLA e SGVNS-RVND aplicados aos problemas de roteamento de veículos abertos são avaliados estatisticamente.

A análise estatística envolvendo os dois métodos foram referentes ao número de

Tabela 5.25: Comparativo entre SGVNSBLA e SGVNS-RVND para PRVAJTMD com resultados do Grupo II.

Instância					SGVNSBLA			SGVNS-RVND			SGVNS-RVND - SGVNSBLA	
	$ \mathcal{C} $	$ \mathcal{D} $	$ \mathcal{K}_d $	$ \mathcal{K} $	Distância	$ \mathcal{U} $	Tempo	Distância	$ \mathcal{U} $	Tempo	Distância	Veículos
pr11a	360	4	10	40	5557,53	39	24,66	5757,87	40	39,10	200,34	1
pr12a	480	4	13	52	6700,00	49	60,63	7123,59	52	66,08	423,59	3
pr13a	600	4	16	64	8143,56	58	104,92	8383,24	64	109,17	239,68	6
pr14a	720	4	19	76	9160,79	69	173,77	9780,85	76	158,81	620,06	7
pr15a	840	4	22	88	10526,67	82	250,22	11544,52	88	173,51	1017,85	6
pr16a	960	4	26	104	11784,71	92	292,00	13221,55	104	180,03	1436,84	12
pr17a	360	6	7	42	5353,51	41	23,90	5560,62	42	27,33	207,11	1
pr18a	520	6	10	60	6939,04	56	77,48	7346,70	60	60,89	407,66	4
pr19a	700	6	13	78	8681,25	69	147,80	9328,91	78	139,97	647,66	9
pr20a	880	6	16	96	10137,58	87	236,47	10750,93	96	171,27	613,35	9
pr21a	420	12	4	48	5612,94	44	40,79	5798,82	48	35,01	185,88	4
pr22a	600	12	6	72	7294,24	66	86,77	7511,24	72	77,99	217,00	6
pr23a	780	12	8	96	8886,52	77	163,62	9427,49	96	107,70	540,97	19
pr24a	960	12	10	120	10787,06	98	238,07	11484,88	119	162,05	697,82	21
pr11b	360	4	8	32	3995,45	32	35,49	4071,32	32	40,39	75,87	–
pr12b	480	4	11	44	5095,10	40	84,50	5311,61	44	78,26	216,51	4
pr13b	600	4	14	56	6232,23	54	153,91	6457,07	56	132,25	224,84	2
pr14b	720	4	17	68	7131,76	60	209,28	7649,92	68	136,02	518,16	8
pr15b	840	4	20	80	8351,09	73	248,08	9048,86	80	179,24	697,77	7
pr16b	960	4	23	92	9371,46	77	297,92	10427,21	91	180,02	1055,75	14
pr17b	360	6	6	36	4125,94	34	33,30	4210,63	36	33,34	84,69	2
pr18b	520	6	9	54	5428,54	48	81,24	5772,21	54	71,29	343,67	6
pr19b	700	6	12	72	6910,97	61	169,08	7349,44	72	103,88	438,47	11
pr20b	880	6	15	90	8314,45	73	268,08	8866,85	90	167,42	552,40	17
pr21b	420	12	4	48	4469,11	40	47,34	4790,29	48	26,49	321,18	8
pr22b	600	12	6	72	5564,69	52	139,76	6320,29	67	48,36	755,60	15
pr23b	780	12	7	84	7194,30	67	200,33	7580,44	79	110,74	386,14	12
pr24b	960	12	8	96	8798,53	85	282,78	9212,93	94	160,76	414,40	9
Soma				1960	206549,02	1723	4172,19	220090,28	1946	2977,37	13541,26	223
Diferença Percentual entre SGVNS-RVND e SGVNSBLA											6,56	12,94

veículos usados e distância total percorrida encontradas pelos algoritmos. Os dados foram normalizados a partir da diferença entre as medianas e os resultados encontrados para cada instância. Os dados foram normalizados em relação as medianas, conforme descrito na Seção 5.4.1.

A Subseção 5.5.3.1 apresenta a análise do desempenho do algoritmo SGVNSBLA; a Seção 5.5.3.2 apresenta a avaliação do desempenho do algoritmo SGVNS-RVND.

5.5.3.1 Avaliação do desempenho do algoritmo SGVNSBLA para PRVAJTMD e PRVAJTMD*

As verificações estatísticas a respeito dos valores produzidos pelo algoritmo SGVNSBLA são mostradas e discutidas nesta seção. A avaliação é guiada no sentido de entender o desempenho do algoritmo em relação ao dois problemas tratados. Foram gerados três *boxplots*, que são mostrados na Figura 5.5. A Figura 5.5(a) mostra os resultados para o PRVAJTMD*, apresentando a distribuição dos dados em torno da mediana para o número de veículos usados. A distribuição dos dados para distância total percorrida é mostrada na Figura 5.5(b).

No *boxplot* mostrado na Figura 5.5(a), com a distribuição amostral em torno da

Tabela 5.26: Comparativo entre SGVNSBLA e SGVNS-RVND para PRVAJTMD*: resultados do Grupo II.

Instância	SGVNSBLA				SGVNS-RVND			SGVNS-RVND - SGVNSBLA	
	$ \mathcal{C} $	$ \mathcal{D} $	$ \mathcal{K}_d $	$ \mathcal{K} $	Distância	$ \mathcal{U} $	Tempo	Distância	Veículos
pr11a	360	4	10	40	6139,64	31	13,75	6117,39	1
pr12a	480	4	13	52	7745,77	41	26,14	7500,05	-
pr13a	600	4	16	64	9212,94	51	45,88	8506,39	-
pr14a	720	4	19	76	10312,82	61	82,44	10809,87	-2
pr15a	840	4	22	88	12191,06	71	122,24	12421,66	1
pr16a	960	4	26	104	14364,60	82	139,67	13266,47	5
pr17a	360	6	7	42	7445,35	31	9,63	6056,34	1
pr18a	520	6	10	60	9675,01	45	23,61	7577,97	1
pr19a	700	6	13	78	10774,15	59	53,74	9992,12	-
pr20a	880	6	16	96	12331,83	73	83,91	11504,57	1
pr21a	420	12	4	48	7221,59	34	13,35	6083,25	-
pr22a	600	12	6	72	10306,89	47	27,67	8139,98	3
pr23a	780	12	8	96	11858,31	62	53,72	10050,26	3
pr24a	960	12	10	120	13695,34	83	89,73	12012,55	2
pr11b	360	4	8	32	4577,75	27	28,65	4332,77	-
pr12b	480	4	11	44	5632,93	35	57,17	6055,14	-
pr13b	600	4	14	56	6952,11	44	71,68	6726,56	-
pr14b	720	4	17	68	10162,23	54	118,31	7888,94	1
pr15b	840	4	20	80	9946,15	66	155,47	9499,00	-
pr16b	960	4	23	92	11403,05	73	224,61	10149,21	1
pr17b	360	6	6	36	5192,61	27	16,99	4590,55	-
pr18b	520	6	9	54	6057,07	40	32,12	6086,92	-1
pr19b	700	6	12	72	7623,41	53	85,72	7247,77	-
pr20b	880	6	15	90	9028,76	69	122,98	9011,14	-
pr21b	420	12	4	48	6049,57	30	19,23	5046,17	-
pr22b	600	12	6	72	8077,73	44	44,96	6173,67	-
pr23b	780	12	7	84	9437,63	58	82,04	7796,45	1
pr24b	960	12	8	96	10768,65	76	118,00	10106,99	2
Soma			1960		254184,95	1467,0	1963,41	230750,15	20
Diferença Percentual entre SGVNS-RVND e SGVNSBLA								-9,22	1,36

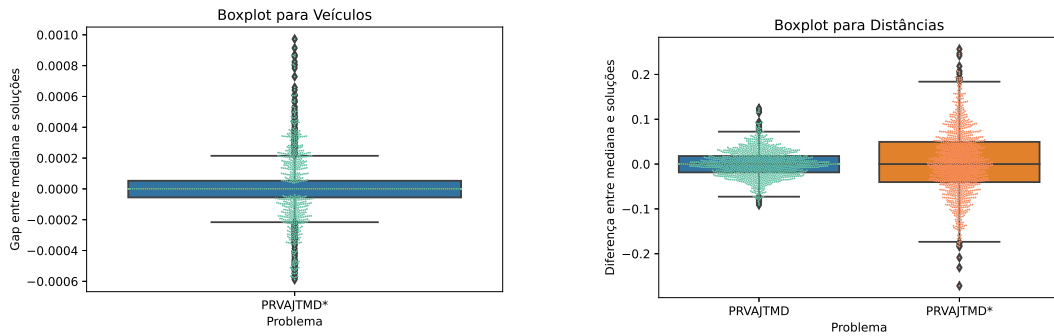
mediana para o número de veículos usados, é possível observar que há uma concentração em torno da mediana. Observa-se uma situação semelhante na distribuição amostral no PRVAJTMD para a distância percorrida, mostrada na Figura 5.5(b). Ao se observar a distribuição para a distância percorrida no PRVAJTMD*, percebe-se maior dispersão nos dados. Essas informações podem ser verificadas a partir dos valores de quartis, mostrados na Tabela 5.27.

Tabela 5.27: Distribuição por quartis para o SGVNSBLA.

Distribuição	Gap para Veículos	Gap para Distâncias	
	PRVAJTMD*	PRVAJTMD	PRVAJTMD*
Mínimo	-0,000589	-0,091082	-2,716582e-01
25%	-0,000055	-0,018509	-4,027291e-02
50%	0,000000	0,000000	-2,986398e-08
75%	0,000053	0,017961	4,956314e-02
Máximo	0,000972	0,124461	2,564894e-01

Os valores de *Gap* para o número de veículos usados mostram uma variação entre o mínimo encontrado e o máximo de 0,16%. Em 50% dos resultados, o erro foi de 0,01%. Para as distâncias percorridas, no caso do PRVAJTMD, houve uma diferença de 3,65% em 50% da amostra. Para o PRAJTMD*, essa diferença foi de 8,98%.

Esses resultados mostram que o SGVNSBLA comporta-se de maneira estável

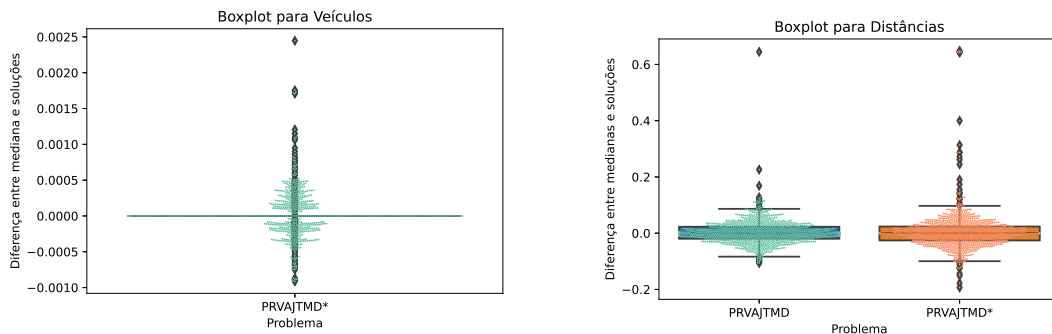
(a) *Gap* para número de veículos usados.(b) *Gap* para distância percorrida.Figura 5.5: *Boxplots* dos resultados do algoritmo SGVNSBLA.

em relação à variável a ser otimizada, fato esse verificado nos valores de distância percorrida para o PRVAJTMD, como também nos valores de número de veículos usados para o PRVAJTMD*. Em ambos os casos, houve uma diferença menor que 5% em 50% das amostras.

5.5.3.2 Avaliação do desempenho do algoritmo SGVNS-RVND para o PRVAJTMD e o PRVAJTMD*

Nesta seção, o desempenho do algoritmo SGVNS-RVND é avaliado estatisticamente. A metodologia adotada é semelhante à usada para a avaliação do algoritmo SGVNSBLA.

Os resultados alcançados pelo SGVNS-RVND em relação à minimização do número de veículos usados não apresentam grandes variações em seu comportamento. Na Figura 5.6(a), observa-se que a distribuição encontra-se em torno da mediana. Dos 50% dos resultados entre o primeiro quartil e terceiro quartil não há variação. A amplitude total foi de 0,34%. Esses resultados mostram a estabilidade do algoritmo SGVNS-RVND em relação à redução do número de veículos usados.

(a) *Gap* para número de veículos usados.(b) *Gap* para distância percorrida.Figura 5.6: *Boxplots* dos resultados do algoritmo SGVNS-RVND.

A Figura 5.6(b) mostra os valores de erro relativos às distâncias percorridas, encontradas nos dois problemas tratados. A partir de sua observação, é possível verificar a estabilidade no comportamento do algoritmo SGVNS-RVND em relação aos valores de distância percorrida encontrados. Para o PRVAJTMD, há uma variação de 4,27% entre 50% dos resultados em torno da mediana, e de 4,95% para o PRVAJTMD*.

Os valores de distância percorrida encontrados pelo algoritmo SGVNS-RVND para o PRVAJTMD* apresentam maior quantidade de *outliers*. Eles indicam uma maior variação entres esses valores. Tais variações não são atípicas, uma vez que PRVAJTMD* não se prioriza a redução da distância percorrida. A variação para os erros relativos dos valores de distância percorrida ficou abaixo de 5% em torno da mediana. Ao analisar essa variação, conclui-se que não há diferenças estatísticas significativas para os dois problemas avaliados. Na Tabela 5.28 são mostrados os resultados dos quartis encontrados para o SGVNS-RVND.

Tabela 5.28: Distribuição por quartis para o algoritmo SGVNS-RVND.

Distribuição	Gap para número de veículos usados	Gap para distância percorrida	
	PRVAJTMD*	PRVAJTMD	PRVAJTMD*
Mínimo	-0,000913	-1,056211e-01	-0,193106
25%	0,000000	-1,982226e-02	-0,025732
50%	0,000000	-2,492419e-08	0,000000
75%	0,000000	2,286653e-02	0,023724
Máximo	0,002446	6,449326e-01	0,648508

5.6 Resumo do Capítulo

Este capítulo apresentou os experimentos computacionais realizados com os algoritmos propostos no Capítulo 4 para a solução de instâncias dos problemas PRVJTMD*, PRVAJTMD e PRVAJTMD*. Foram apresentados os ajustes dos parâmetros utilizados nos quatro algoritmos utilizados e apresentados os resultados encontrados para o PRVJTMD* e para os dois problemas com roteamento aberto com janelas de tempo e múltiplos depósitos.

O PRVJTMD* foi resolvido com os algoritmos SGVNSBLA, SGVNS e BGVNS. O resultados encontrados pelo SGVNSBLA foram comparados com os dos outros dois algoritmos. Os operadores responsáveis pela geração das estruturas de vizinhanças durante a resolução do problema foram avaliadas de forma estatística. Os resultados encontrados permitiram avaliar a contribuição de cada operador durante a perturbação e busca local, o que permitiu verificar a contribuição de cada estrutura de vizinhança na resolução do problema.

Os algoritmos SGVNSBLA e SGVNS-RVND foram utilizados na solução dos problemas de roteamento aberto PRVAJTMD e PRVAJTMD*. Os resultados encontrados por cada algoritmo foram relatados de forma independente e comparados entre

si. Também foi realizada uma análise estatística dos métodos e da influência das estruturas de vizinhança no desempenho desses algoritmos.

Capítulo 6

Conclusões Gerais

Esta tese discutiu problemas de roteamento de veículos com janelas de tempo e múltiplos depósitos. O problema básico apresentado é o Problema de Roteamento de Veículos com Janelas de Tempo e Múltiplos Depósitos (PRVJTMD), que é um problema com roteamento fechado, ou seja, os veículos saem de um depósito e retornam a ele após cumprir sua rota. O objetivo desse problema é a minimização dos custos de transporte, associados unicamente à distância total percorrida. O segundo problema abordado foi o Problema de Roteamento de Veículos com Janelas de Tempo e Múltiplos Depósitos visando a minimização do número total de veículos usados (PRVJTMD*), que também é um problema de roteamento fechado, mas que se diferencia do PRVJTMD ao incluir, na função objetivo a ser minimizada, o número total de veículos usados e priorizar o atendimento desse objetivo em relação ao de minimizar a distância total percorrida. O terceiro problema tratado é o Problema de Roteamento Aberto de Veículos com Janelas de Tempo e Múltiplos Depósitos (PRAVJTMD), no qual o veículo não retorna ao depósito de origem de sua rota após cumpri-la. Nesse problema, do mesmo modo como no PRVJTMD, o objetivo é a minimização da distância total percorrida. O quarto e último problema estudado é o Problema de Roteamento Aberto de Veículos com Janelas de Tempo e Múltiplos Depósitos visando a minimização do número total de veículos usados (PRAVJTMD*). Esse problema é a formulação em roteamento aberto do PRVJTMD*.

Diante destes problemas, três aspectos foram abordados ao longo do trabalho. O primeiro foi propor soluções que objetivavam a redução da frota de veículos; o segundo foi a elaboração de métodos que solucionem os problemas de roteamento de veículos com janelas de tempo e múltiplos depósitos; e o último aspecto, foi verificar como os operadores utilizados nas buscas locais se comportam no processo de refinamento das soluções. Em resposta às questões abordadas, foi feita uma revisão bibliográfica dos problemas elencados, os modelos matemáticos atribuídos a cada um, e as prin-

cipais técnicas utilizadas na solução destes problemas. A proposição dos problemas PRVJTMD* e PRAVJTMD* é, assim, uma das principais contribuições dessa tese.

A partir do estudo dos trabalhos relativos à temática abordada, o algoritmo denominado SGVNSBLA foi desenvolvido e proposto como método de solução dos PRJTMDs abordados. Ele tem como base o algoritmo SGVNS, que, por sua vez, é baseado na meta-heurística VNS. Neste algoritmo, a perturbação ocorre de maneira gradativa, em que um mesmo operador é aplicado à solução uma ou várias vezes, de forma gradativa. Esta ação proporciona um distanciamento do ótimo local a cada novo vizinho gerado. Após a perturbação, a heurística de busca local VND é aplicada, retornando o ótimo local. A principal mudança proposta entre os algoritmos SGVNS e SGVNSBLA está na fase de busca local. Neste trabalho, foi desenvolvida uma metodologia de seleção dos operadores que atuam no refinamento da solução. Este método tem, como objetivo, selecionar operadores mais adaptados às vizinhanças exploradas na busca local, permitindo melhorar os resultados encontrados e reduzir o tempo computacional gasto.

No decorrer da tese, o desempenho do algoritmo SGVNSBLA foi comparado com o dos algoritmos SGVNS, BGVNS e SGVNS-RVND. O desempenho desses algoritmos foi avaliado por meio de testes computacionais em 68 instâncias. Destas, 48 foram base para avaliação nos problemas estudados, com um total de 7970 testes, sendo 3890 foram com o algoritmo SGVNSBLA, 880 com o algoritmo SGVNS, 880 com algoritmo BGVNS e 2320 com o algoritmo SGVNS-RVND. De acordo com os resultados obtidos, o algoritmo SGVNSBLA obteve melhores resultados em todos os conjuntos de testes realizados para a quantidade de veículos usados e as distâncias totais percorridas. Em relação ao tempo computacional gasto, ele obteve tempos menores em um dos conjuntos de testes para o PRVJTMD* e nos conjuntos de testes para os dois problemas de roteamento de veículos aberto apresentados.

Por fim, pode-se concluir que os métodos computacionais desenvolvidos podem ser utilizados em processos de otimização como uma boa alternativa em problemas de roteamento de veículos com múltiplos depósitos.

6.1 Trabalhos Publicados

No decorrer do desenvolvimento desta tese os seguintes artigos foram publicados:

- (i) Bezerra, S. N.; Souza, S. R. e Souza, M. J. F. (2018). A GVNS algorithm for solving the multi-depot vehicle routing problem. *Electronic Notes in Discrete Mathematics*, v. 66, p. 167 – 174. ISSN 1571-0653. 5th International Conference on Variable Neighborhood Search. <https://www.sciencedirect.com/>

[science/article/abs/pii/S1571065318300684](https://www.sciencedirect.com/science/article/abs/pii/S1571065318300684).

- (ii) Bezerra, Sinaide Nunes; Souza, Marcone Jamilson Freitas e de Souza, Sérgio Ricardo (2023). A variable neighborhood search-based algorithm with adaptive local search for the vehicle routing problem with time windows and multi-depots aiming for vehicle fleet reduction. *Computers & Operations Research*, v. 149, p. 106016. ISSN 0305-0548. doi: <https://doi.org/10.1016/j.cor.2022.106016>. URL <https://www.sciencedirect.com/science/article/pii/S030505482200>.
- (iii) Bezerra, Sinaide Nunes; Souza, Marcone Jamilson Freitas e de Souza, Sérgio Ricardo. (2023). A General VNS for the Multiple Depot Open Vehicle Routing Problem with Time Windows. *Optimization Letters*. Aceito para publicação.

6.2 Propostas de Trabalhos Futuros

Embora esses algoritmos tenham obtido um bom desempenho computacional, algumas técnicas ainda merecem uma melhor atenção. Como trabalhos futuros, podem se incorporar técnicas que desativem algumas das vizinhanças utilizadas na perturbação e/ou na busca local após encontrarem uma boa solução para o problema. Esta ação pode proporcionar que melhores soluções sejam encontradas pelos algoritmos. Isso porque alguns dos operadores utilizados na busca são mais aptos em reduzir a quantidade de veículos, ou na procura por menores rotas. Como exemplo, podem ser citadas as estruturas de vizinhanças de destruição e reconstrução, que são especialmente importantes na redução da quantidade de veículos. Em problemas no qual o objetivo principal é a redução da distância, a exploração dessas vizinhanças pode elevar muito o tempo computacional. Diante desta proposta, testes mais profundos deste mecanismo nos problemas estudados e em outras variantes do problema também devem ser considerados.

Outra questão a ser explorada são os modelos matemáticos propostos para o PRVJTMD* e para os dois problemas de roteamento abertos. Sua implementação em resolvedores como CPLEX podem ser uma boa oportunidade para verificar sua acurácia e posterior comparação dos resultados encontrados com os produzidos nesta tese. As propostas desenvolvidas fornecem uma boa base heurística na geração de soluções baseadas na exploração de vizinhanças. Os resultados apresentados fornecem uma visão a respeito do comportamento dessas vizinhanças. Foram encontrados resultados inéditos para o PRVJTMD aberto, que podem contribuir para a geração de outras pesquisas no tema.

Bibliografia

Abdallah, Marwa Ben e Ennigrou, Meriem. (2020). Hybrid multi-agent approach to solve the multi-depot heterogeneous fleet vehicle routing problem with time window (mdhfvrtw). Madureira, Ana Maria; Abraham, Ajith; Gandhi, Niketa e Varela, Maria Leonilde, editors, *Hybrid Intelligent Systems*, volume 923, p. 376–386, Cham. Springer International Publishing. doi: https://doi.org/10.1007/978-3-030-14347-3_37.

Aziz, Rafidah Abdul; Ayob, Masri; Othman, Zalinda; Ahmad, Zulkifli e Sabar, Nasser R. Nov(2017). An adaptive guided variable neighborhood search based on honey-bee mating optimization algorithm for the course timetabling problem. *Soft Computing*, v. 21, n. 22, p. 6755–6765. ISSN 1433-7479.

Aziz, Rafidah Abdul; Ayob, Masri; Othman, Zalinda e Sarim, Hafiz Mohd. (2013). Adaptive guided variable neighborhood search. *Journal of Applied Sciences*, v. 13, n. 6, p. 883–888. ISSN 1812-5654. doi: 10.3923/jas.2013.883.888.

Belloso, Javier; Juan, Angel e Faulin, Javier. 02(2019). An iterative biased-randomized heuristic for the fleet size and mix vehicle-routing problem with backhauls. *International Transactions in Operational Research*, v. 26, p. 289–301.

Bezerra, S. N.; Souza, S. R. e Souza, M. J. F. (2018). A gvns algorithm for solving the multi-depot vehicle routing problem. *Electronic Notes in Discrete Mathematics*, v. 66, p. 167 – 174. ISSN 1571-0653. 5th International Conference on Variable Neighborhood Search.

Bezerra, Sinaide Nunes; Souza, Marcone Jamilson Freitas e de Souza, Sérgio Ricardo. (2023). A variable neighborhood search-based algorithm with adaptive local search for the vehicle routing problem with time windows and multi-depots aiming for vehicle fleet reduction. *Computers & Operations Research*, v. 149, p. 106016. ISSN 0305-0548. doi: <https://doi.org/10.1016/j.cor.2022.106016>. URL <https://www.sciencedirect.com/science/article/pii/S0305054822002465>.

Brandão, José. 09(2004). A tabu search algorithm for the open vehicle routing problem. *European Journal of Operational Research*, v. 157, p. 552–564.

Brandão, José. 07(2020). A memory-based iterated local search algorithm for the multi-depot open vehicle routing problem. *European Journal of Operational Research*, v. 284.

- Bräysy, Olli e Gendreau, Michel. (2005)a. Vehicle routing problem with time windows, part i: Route construction and local search algorithms. *Transportation Science*, v. 39, n. 1, p. 104–118. doi: 10.1287/trsc.1030.0056.
- Bräysy, Olli e Gendreau, Michel. 02(2005)b. Vehicle routing problem with time windows, part ii: Metaheuristics. *Transportation Science*, v. 39, p. 119–139. doi: 10.1287/trsc.1030.0057.
- Chen, Binhui; Qu, Rong; Bai, Ruibin e Laesanklang, Wasakorn. 08(2019). A variable neighborhood search algorithm with reinforcement learning for a real-life periodic vehicle routing problem with time windows and open routes. *RAIRO - Operations Research*, v. 54.
- Christiaens, Jan e Vanden Berghe, Greet. (2020). Slack induction by string removals for vehicle routing problems. *Transportation Science*, v. 54, n. 2, p. 417–433. doi: 10.1287/trsc.2019.0914.
- Cordeau, J. F.; G, Laporte e A, Mercier. 08(2001). A unified tabu search heuristic for vehicle routing problems with time windows. *Journal of the Operational Research Society*, v. 52, p. 928–936.
- Cordeau, J. F.; Gendreau, M. e Laporte, G. (1997). A tabu search heuristic for periodic and multi-depot vehicle routing problems. *Networks*, v. 30, n. 2, p. 105–119. ISSN 1097-0037.
- Cordeau, J. F.; Laporte, G. e Mercier, A. 05(2004). Improved tabu search algorithm for the handling of route duration constraints in vehicle routing problems with time windows. *Journal of the Operational Research Society*, v. 55, p. 542–546. doi: 10.1057/palgrave.jors.2601707.
- Cordeau, Jean-François; Gendreau, Michel; Hertz, Alain; Laporte, Gilbert e Sormany, Jean-Sylvain. 12(2005). *New Heuristics for the Vehicle Routing Problem*, p. 279–297. Springer US. ISBN 0-387-24971-0. doi: 10.1007/0-387-24977-X_9.
- Cordeau, Jean-François e Maischberger, Mirko. 09(2012). A parallel iterated tabu search heuristic for vehicle routing problems. *Computers & OR*, v. 39, p. 2033–2050. doi: 10.1016/j.cor.2011.09.021.
- Dantzig, G. B. e Ramser, J. H. (1959). The truck dispatching problem. *Management Science*, v. 6, n. 1, p. 80–91. ISSN 00251909, 15265501.
- Gillett, Billy E e Johnson, Jerry G. (1976). Multi-terminal vehicle-dispatch algorithm. *Omega*, v. 4, n. 6, p. 711–718.
- Hansen, Pierre; Mladenović, Nenad e Moreno Pérez, José A. (2010). Variable neighbourhood search: methods and applications. *Annals of Operations Research*, v. 175, n. 1, p. 367–407. ISSN 1572-9338.
- Hansen, Pierre; Mladenović, Nenad; Todosijević, Raca e Hanafi, Saïd. September(2017). Variable neighborhood search: basics and variants. *EURO Journal on Computational Optimization*, v. 5, n. 3, p. 423–454.

- Karakostas, Panagiotis; Panoskaltsis, Nicki; Mantalaris, Athanasios e Georgiadis, Michael. 05(2020)a. Optimization of car t-cell therapies supply chains. *Computers & Chemical Engineering*, v. 139, p. 106913. doi: 10.1016/j.compchemeng.2020.106913.
- Karakostas, Panagiotis; Sifaleras, Angelo e Georgiadis, Michael. 04(2020)b. Adaptive variable neighborhood search solution methods for the fleet size and mix pollution location-inventory-routing problem. *Expert Systems with Applications*, v. 153, p. 113444. doi: 10.1016/j.eswa.2020.113444.
- Lahyani, Rahma; Gouguenheim, Anne-Lise e Coelho, Leandro C. (2019). A hybrid adaptive large neighbourhood search for multi-depot open vehicle routing problems. *International Journal of Production Research*, v. 57, n. 22, p. 6963–6976. doi: 10.1080/00207543.2019.1572929.
- Lalla-Ruiz, Eduardo e Mes, Martijn. (2021). Mathematical formulations and improvements for the multi-depot open vehicle routing problem. *Optimization Letters*, v. 15, p. 271–286.
- Laporte, Gilbert. (2009). Fifty years of vehicle routing. *Transportation Science*, v. 43, n. 4, p. 408–416. doi: 10.1287/trsc.1090.0301.
- Lenstra, J. K. e Kan, A. H. G. Rinnooy. (1981). Complexity of vehicle routing and scheduling problems. *Networks*, v. 11, n. 2, p. 221–227. doi: <https://doi.org/10.1002/net.3230110211>.
- Li, Jian; Li, Yang e Pardalos, Panos M. Feb(2016). Multi-depot vehicle routing problem with time windows under shared depot resources. *Journal of Combinatorial Optimization*, v. 31, n. 2, p. 515–532. ISSN 1573-2886. doi: 10.1007/s10878-014-9767-4.
- Li, Yongbo; Soleimani, Hamed e Zohal, Mostafa. (2019). An improved ant colony optimization algorithm for the multi-depot green vehicle routing problem with multiple objectives. *Journal of Cleaner Production*, v. 227, p. 1161–1172. doi: <https://doi.org/10.1016/j.jclepro.2019.03.185>.
- López-Ibáñez, M.; Dubois-Lacoste, J.; L. P. Cáceres, L. P.; Birattari, M. e Stützle, T. (2016). The irace package: Iterated racing for automatic algorithm configuration. *Operations Research Perspectives*, v. 3, p. 43–58.
- Masmoudi, Mohamed; Hosny, Manar e Koç, Çağrı. 03(2021). The fleet size and mix vehicle routing problem with synchronized visits. *Transportation Letters The International Journal of Transportation Research*, v. . doi: 10.1080/19427867.2021.1888196.
- Mladenović, N. e Hansen, P. (1997). Variable neighborhood search. *Computers & Operations Research*, v. 24, n. 11, p. 1097 – 1100. ISSN 0305-0548.
- Montgomery, D. C. e Runger, G. C. (2016). *Estatística Aplicada e Probabilidade para Engenheiros*. LTC Editora.

- Montoya-Torres, Jairo R.; Franco, Julián López; Isaza, Santiago Nieto; Jiménez, Heriberto Felizzola e Herazo-Padilla, Nilson. (2015). A literature review on the vehicle routing problem with multiple depots. *Computers & Industrial Engineering*, v. 79, p. 115 – 129. ISSN 0360-8352. doi: <https://doi.org/10.1016/j.cie.2014.10.029>.
- Nalepa, Jakub e Blocho, Mirosław. (2016). Adaptive memetic algorithm for minimizing distance in the vehicle routing problem with time windows. *Soft Comput.*, v. 20, n. 6, p. 2309–2327.
- Noori, Siamak e Ghannadpour, S. Farid. Jun(2012). High-level relay hybrid metaheuristic method for multi-depot vehicle routing problem with time windows. *Journal of Mathematical Modelling and Algorithms*, v. 11, n. 2, p. 159–179. ISSN 1572-9214. doi: 10.1007/s10852-011-9171-3.
- Oliveira, Fernando Bernardes; Enayatifar, Rasul; Sadaei, Hossein Javedani; Guimarães, Frederico Gadelha e Potvin, Jean-Yves. (2016). A cooperative coevolutionary algorithm for the multi-depot vehicle routing problem. *Expert Systems with Applications*, v. 43, p. 117 – 130. ISSN 0957-4174.
- Ombuki, Beatrice; Ross, Brian J. e Hanshar, Franklin. February(2006). Multi-objective genetic algorithms for vehicle routing problem with time windows. *Applied Intelligence*, v. 24, n. 1, p. 17–30.
- Osman, Ibrahim. 12(1993). Meta-strategy simulated annealing and tabu search algorithms for the vehicle routine problem. *Annals of Operations Research*, v. 41, p. 421–451. doi: 10.1007/BF02023004.
- Pisinger, David e Ropke, Stefan. August(2007). A general heuristic for vehicle routing problems. *Comput. Oper. Res.*, v. 34, n. 8, p. 2403–2435. ISSN 0305-0548.
- Polacek, M.; Hartl, R. F.; Doerner, K. e Reimann, M. Dec(2004). A variable neighborhood search for the multi depot vehicle routing problem with time windows. *Journal of Heuristics*, v. 10, n. 6, p. 613–627. ISSN 1572-9397. doi: 10.1007/s10732-005-5432-5.
- Polacek, Michael; Benkner, Siegfried; Doerner, Karl F. e Hartl, Richard F. Dec(2008). A cooperative and adaptive variable neighborhood search for the multi depot vehicle routing problem with time windows. *Business Research*, v. 1, n. 2, p. 207–218. ISSN 2198-2627. doi: 10.1007/BF03343534.
- Potvin, Jean-Yves e Rousseau, Jean-Marc. Dec(1995). An exchange heuristic for routing problems with time windows. *Journal of the Operational Research Society*, v. 46, n. 12, p. 1433–1446. ISSN 1476-9360. doi: 10.1057/jors.1995.204.
- Rabbouch, Bochra; Mraihi, Rafaa e Saâdaoui, Foued. (2018). A recent brief survey for the multi depot heterogenous vehicle routing problem with time windows. Abraham, Ajith; Muhuri, Pranab Kr.; Muda, Azah Kamilah e Gandhi, Niketa, editors, *Hybrid Intelligent Systems*, volume 734, p. 147–157, Cham. Springer International Publishing. doi: https://doi.org/10.1007/978-3-319-76351-4_15.

- Rahmani Hosseinabadi, Ali Asghar; Vahidi, Javad; Balas, Valentina e Mirkamali, Seyedsaeid. 05(2018). OVRP_GELS: Solving open vehicle routing problem using the gravitational emulation local search algorithm. *Neural Computing and Applications*, v. 29.
- Redi, A. A. N. Perwira; Maghfiroh, Meilinda F. N. e Yu, Vincent F. (2013). An improved variable neighborhood search for the open vehicle routing problem with time windows. *2013 IEEE International Conference on Industrial Engineering and Engineering Management*, p. 1641–1645, (2013).
- Rego, Marcelo Ferreira e Souza, Marcone Jamilson Freitas. (2019). Smart general variable neighborhood search with local search based on mathematical programming for solving the unrelated parallel machine scheduling problem. *Proceedings of the 21st International Conference on Enterprise Information Systems (ICEIS 2019)*, volume 2, p. 287–295. INSTICC, SciTePress, (2019). ISBN 978-989-758-372-8. doi: 10.5220/0007703302870295.
- Ren, Xuan; Huang, Hui; Feng, Shuo e Liang, Gongqian. 12(2020). An improved variable neighborhood search for bi-objective mixed-energy fleet vehicle routing problem. *Journal of Cleaner Production*, v. 275, p. 124155. doi: 10.1016/j.jclepro.2020.124155.
- Repoussis, Panagiotis; Tarantilis, C e Ioannou, George. 03(2007). The open vehicle routing problem with time windows. *Journal of The Operational Research Society - J OPER RES SOC*, v. 58, p. 355–367.
- Rezgui, Dhekra; Chaouachi Siala, Jouhaina; Aggoune-Mtalaa, Wassila e Bouziri, Hend. (2019). Application of a variable neighborhood search algorithm to a fleet size and mix vehicle routing problem with electric modular vehicles. *Computers & Industrial Engineering*, v. 130, p. 537–550. ISSN 0360-8352. doi: <https://doi.org/10.1016/j.cie.2019.03.001>.
- Salhi, Said; Imran, Arif e Wassan, Niaz A. December(2014). The multi-depot vehicle routing problem with heterogeneous vehicle fleet: Formulation and a variable neighborhood search implementation. *Computers & Operations Research*, v. 52, n. PB, p. 315–325. ISSN 0305-0548.
- Schrage, Linus. (1981). Formulation and structure of more complex/realistic routing and scheduling problems. *Networks*, v. 11, n. 2, p. 229–232.
- Shen, Ling; Tao, Fengming e Wang, Songyi. 09(2018). Multi-depot open vehicle routing problem with time windows based on carbon trading. *International Journal of Environmental Research and Public Health*, v. 15, p. 2025.
- Solomon, Marius M. (1987). Algorithms for the vehicle routing and scheduling problems with time window constraints. *Operations Research*, v. 35, n. 2, p. 254–265.
- Soto, María; Sevaux, Marc; Rossi, André e Reinholz, Andreas. (2017). Multiple neighborhood search, tabu search and ejection chains for the multi-depot open vehicle routing problem. *Computers & Industrial Engineering*, v. 107, p. 211–222. ISSN 0360-8352. doi: <https://doi.org/10.1016/j.cie.2017.03.022>.

- Souza, M.J.F.; Coelho, I.M.; Ribas, S.; Santos, H.G. e Merschmann, L.H.C. (2010). A hybrid heuristic algorithm for the open-pit-mining operational planning problem. *European Journal of Operational Research*, v. 207, n. 2, p. 1041–1051. ISSN 0377-2217.
- Subramanian, A.; Drummond, L.M.A.; Bentes, C.; Ochi, L.S. e Farias, R. (2010). A parallel heuristic for the vehicle routing problem with simultaneous pickup and delivery. *Computers & Operations Research*, v. 37, n. 11, p. 1899 – 1911. ISSN 0305-0548. Metaheuristics for Logistics and Vehicle Routing.
- Subramanian, Anand; Penna, Puca Huachi Vaz; Ochi, Luiz Satoru e Souza, Marcone Jamilson Freitas. (2013)a. Um algoritmo heurístico baseado em *Iterated Local Search* para problemas de roteamento de veículos. Lopes, Heitor Silvério; deAbreu Rodrigues, Luiz Carlos e Steiner, Maria Teresinha Arns, editors, *Meta-Heurísticas em Pesquisa Operacional*, Capítulo 11, p. 165–180. Omnipax, Curitiba, PR, 1 edição. ISBN 978-85-64619-10-4. doi: 10.7436/2013.mhpo.11.
- Subramanian, Anand; Uchoa, Eduardo e Ochi, Luiz Satoru. (2013)b. A hybrid algorithm for a class of vehicle routing problems. *Computers & Operations Research*, v. 40, n. 10, p. 2519 – 2531.
- Taillard, Éric; Badeau, Philippe; Gendreau, Michel; Guertin, François e Potvin, Jean-Yves. 05(1997). A tabu search heuristic for the vehicle routing problem with soft time windows. *Transportation Science*, v. 31, p. 170–186. doi: 10.1287/trsc.31.2.170.
- Tamashiro, Hiroshi; Nakamura, Morikazu; Okazaki, Takeo e Kang, Dongshik. (2010). A tabu search approach combined with an extended saving method for multi-depot vehicle routing problems with time windows. *Biomedical Soft Computing and Human Sciences*, v. 15, p. 31–39. doi: 10.24466/ijbschs.15.1_29.
- Tansini, L. e Viera, O. Mar(2006). New measures of proximity for the assignment algorithms in the mdvrptw. *Journal of the Operational Research Society*, v. 57, n. 3, p. 241–249. ISSN 1476-9360. doi: 10.1057/palgrave.jors.2601979.
- Tarantilis, C.D. e Kiranoudis, Chris. 01(2002). Distribution of fresh meat. *Journal of Food Engineering*, v. 51, p. 85–91. doi: 10.1016/S0260-8774(01)00040-1.
- Ting, Ching-Jung e Chen, Chia-Ho. 12(2008). Combination of multiple ant colony system and simulated annealing for the multidepot vehicle-routing problem with time windows. *Transportation Research Record*, v. 2089, p. 85–92. doi: 10.3141/2089-11.
- Toth, Paolo e Vigo, Daniele. (2002). *The Vehicle Routing Problem*. Monographs on Discrete Mathematics and Applications. Society for Industrial and Applied Mathematics (SIAM), Philadelphia.
- Vidal, Thibaut; Crainic, Teodor Gabriel; Gendreau, Michel; Lahrichi, Nadia e Rei, Walter. May(2012). A hybrid genetic algorithm for multidepot and periodic vehicle routing problems. *Operations Research*, v. 60, n. 3, p. 611–624.

- Vidal, Thibaut; Crainic, Teodor Gabriel; Gendreau, Michel e Prins, Christian. (2013)a. Heuristics for multi-attribute vehicle routing problems: A survey and synthesis. *European Journal of Operational Research*, v. 231, n. 1, p. 1–21.
- Vidal, Thibaut; Crainic, Teodor Gabriel; Gendreau, Michel e Prins, Christian. (2013)b. A hybrid genetic algorithm with adaptive diversity management for a large class of vehicle routing problems with time-windows. *Computers & Operations Research*, v. 40, n. 1, p. 475–489. ISSN 0305-0548. doi: <https://doi.org/10.1016/j.cor.2012.07.018>.
- Vidal, Thibaut; Crainic, Teodor Gabriel; Gendreau, Michel e Prins, Christian. (2014). Implicit depot assignments and rotations in vehicle routing heuristics. *European Journal of Operational Research*, v. 237, n. 1, p. 15 – 28. ISSN 0377-2217. doi: <https://doi.org/10.1016/j.ejor.2013.12.044>.
- Wang, Jiahai; Weng, Taiyao e Zhang, Qingfu. (2019)a. A two-stage multiobjective evolutionary algorithm for multiobjective multidepot vehicle routing problem with time windows. *IEEE Transactions on Cybernetics*, v. 49, n. 7, p. 2467–2478. doi: [10.1109/TCYB.2018.2821180](https://doi.org/10.1109/TCYB.2018.2821180).
- Wang, Yong; Assogba, Kevin; Fan, Jianxin; Xu, Maozeng; Liu, Yong e Wang, Haizhong. (2019)b. Multi-depot green vehicle routing problem with shared transportation resource: Integration of time-dependent speed and piecewise penalty cost. *Journal of Cleaner Production*, v. 232, p. 12–29. ISSN 0959-6526. doi: <https://doi.org/10.1016/j.jclepro.2019.05.344>.