Problema de roteamento de veículos com custos de fronteira

Lucas Esperancini Moreira e Moreira

Dissertação de Mestrado do Programa de Pós-Graduação em Ciências de Computação e Matemática Computacional (PPG-CMC)

SERVIÇO DE PÓS-GRADUAÇÃO DO ICMC-USP

Data de Depósito:

Assinatura: _____

Lucas Esperancini Moreira e Moreira

Problema de roteamento de veículos com custos de fronteira

Dissertação apresentada ao Instituto de Ciências Matemáticas e de Computação – ICMC-USP, como parte dos requisitos para obtenção do título de Mestre em Ciências – Ciências de Computação e Matemática Computacional. *VERSÃO REVISADA*

Área de Concentração: Ciências de Computação e Matemática Computacional

Orientadora: Profa. Dra. Franklina Maria Bragion Toledo

USP – São Carlos
Julho de 2018

Ficha catalográfica elaborada pela Biblioteca Prof. Achille Bassi
e Seção Técnica de Informática, ICMC/USP,
com os dados inseridos pelo(a) autor(a)

M835p MOREIRA, LUCAS ESPERANCINI MOREIRA
Problema de roteamento de veículos com custos de
fronteira / LUCAS ESPERANCINI MOREIRA MOREIRA;
orientadora Franklina Maria Bragion Toledo. -- São
Carlos, 2018.
88 p.

Dissertação (Mestrado - Programa de Pós-Graduação
em Matemática) -- Instituto de Ciências Matemáticas
e de Computação, Universidade de São Paulo, 2018.

1. Roteamento de veículos. 2. Heurística. I.
Toledo, Franklina Maria Bragion, orient. II. Título.

Lucas Esperancini Moreira e Moreira

Vehicle routing problem with border costs

Master dissertation submitted to the Institute of Mathematics and Computer Sciences – ICMC-USP, in partial fulfillment of the requirements for the degree of the Master Program in Computer Science and Computational Mathematics. *FINAL VERSION*

Concentration Area: Computer Science and Computational Mathematics

Advisor: Profa. Dra. Franklina Maria Bragion Toledo

USP – São Carlos
July 2018

AGRADECIMENTOS

Primeiramente, agradeço à professora Franklina por todo apoio, do início ao fim, nos momentos fáceis e difíceis, durante toda a pesquisa. Com sua ajuda consegui aprender e crescer muito. Agradeço à minha família (meus pais e meu irmão) pelo apoio ao longo dessa jornada acadêmica. Direciono muita gratidão à querida Érica K. Nishimura que acompanhou de perto essa caminhada com incríveis conversas e momentos de apoio. Agradeço aos meus colegas de laboratório que me ajudaram diretamente ou indiretamente durante a pesquisa. Em particular, agradeço ao Pedro Belin Castellucci pelas muitas sugestões técnicas que ajudaram a moldar este trabalho. Por fim, mas não menos importante, agradeço a todos meus amigos da cidade do clima que fizeram parte dos momentos fora da vida acadêmica.

*“O maior inimigo do conhecimento
não é a ignorância, é a ilusão do conhecimento.”
(Stephen Hawking)*

RESUMO

MOREIRA, L. E. M. **Problema de roteamento de veículos com custos de fronteira**. 2018. 88 p. Dissertação (Mestrado em Ciências – Ciências de Computação e Matemática Computacional) – Instituto de Ciências Matemáticas e de Computação, Universidade de São Paulo, São Carlos – SP, 2018.

O problema de roteamento de veículos é um dos problemas de otimização combinatória mais estudados nas últimas décadas. Neste trabalho, é estudada uma variante do problema de roteamento de veículos capacitado em que são considerados custos adicionais em viagens que cruzam fronteiras entre estados. Duas abordagens foram apresentadas para considerar tal característica: adicionar custos fixos às viagens de clientes de estados diferentes e adicionar custos que consideram a carga do veículo ao cruzar a fronteira e, para ambas, foram apresentados modelos matemáticos. Um *solver* comercial foi utilizado para resolver instâncias conhecidas da literatura e devido à resolução ter atingido o tempo máximo computacional para grande parte dos testes, uma *Variable Neighborhood Descent* com múltiplos inícios foi desenvolvida para a resolução do problema. Os múltiplos inícios são gerados perturbando a solução inicial gerada para a heurística. Como esperado, tanto para a resolução via modelagem quanto a resolução via heurística, considerar custos de fronteira proporcionais a carga apresentaram soluções de melhor qualidade. Essa nova proposta para abordar custos reais de fronteira abre novas possibilidades para considerar custos de fronteira fixos e proporcionais a carga concomitantemente para melhor representar aplicações reais.

Palavras-chave: Roteamento de Veículos Capacitado, VND, Múltiplos inícios, Fronteira.

ABSTRACT

MOREIRA, L. E. M. **Vehicle routing problem with border costs**. 2018. 88 p. Dissertação (Mestrado em Ciências – Ciências de Computação e Matemática Computacional) – Instituto de Ciências Matemáticas e de Computação, Universidade de São Paulo, São Carlos – SP, 2018.

The vehicle routing problem is one of the most studied combinatorial optimization problems in the last decades. In this paper, a variant of vehicle routing problem was studied in which the border costs was added to trips that cross borders. In order to consider such characteristic, two approaches were made: add fixed costs for the trips which clients are from different states and add costs that consider the amount of cargo in the vehicle when it crosses the border. In order to consider such characteristics, models were presented. Instances of literature were solved with a commercial solver and due to high computational time obtained from the exact method, a heuristic with Variable Neighborhood Descent as the local search in a multiple start environment was implemented. The multiple starts were generated making a perturbation in the initial solution obtained for the heuristic. As expected, approaching the problem considering the border cost proportional to the cargo in the vehicle presented better results. This study gives the first results for solving the vehicle routing problem considering real border costs and gives the possibility for solving the problem considering real fixed and proportional costs simultaneously in order to better represent real applications.

Keywords: Capacitated Vehicle Routing, VND, Multiple Start, Border.

LISTA DE ILUSTRAÇÕES

Figura 1 – Um exemplo de grafo completo com 6 vértices.	30
Figura 2 – Exemplo de um grafo para um problema de roteamento de veículos com 25 clientes.	30
Figura 3 – Duas possíveis soluções para o PRV para a Figura 2.	31
Figura 4 – Etapas da heurística de Clarke e Wright (1964)	33
Figura 5 – Etapas da heurística Inserção do Mais Distante	34
Figura 6 – Exemplo com duas rotas	35
Figura 7 – Um movimento 2-opt inter-rotas	35
Figura 8 – Movimentos <i>1-0 exchange</i> e <i>1-1 exchange</i>	36
Figura 9 – Diferença entre o 2-opt e o cross-exchange.	36
Figura 10 – Movimento 2-opt intra-rota	37
Figura 11 – Movimento Or-opt.	37
Figura 12 – Movimento iopt.	38
Figura 13 – Movimento <i>swap</i>	38
Figura 14 – Ilustração das quatro etapas do método de Desrosiers, Dumas e Soumis (1988).	44
Figura 15 – Ilustração do método de Kim, Kim e Sahoo (2006).	46
Figura 16 – Ilustração do método de Dondo e Cerdá (2007).	46
Figura 17 – Etapas do método de Forma, Raviv e Tzur (2015)	48
Figura 18 – Um exemplo com 25 clientes para o CluVRP.	50
Figura 19 – Etapas do método de Battarra, Erdogan e Vigo (2014)	51
Figura 20 – Exemplo de clientes separados em estados.	53
Figura 21 – Rotas em fronteiras.	54
Figura 22 – Exemplo de arestas com custos adicionais.	55
Figura 23 – Comparação entre soluções obtidas pelo PRVC e PRVCFF.	58
Figura 24 – Geração de estados nas instâncias utilizadas.	60
Figura 25 – Diferença entre os custos das soluções para cada instância.	61
Figura 26 – Diferença entre os modelos PRVCFP e PRVCFF.	63
Figura 27 – Geração de múltiplas soluções.	72
Figura 28 – Resultados obtidos para as heurísticas CW-VND e CW-VND-MI comparados com a literatura.	75
Figura 29 – Comparação entre as soluções obtidas pela CW-VND-MI e pelo modelo PRVCFF.	76
Figura 30 – Comparação entre as soluções obtidas pela CW-VND-MI-P e pelo modelo PRVCFP.	76
Figura 31 – Comparação entre as soluções obtidas pela CW-VND-MI e pela CW-VND-MI-P.	77

LISTA DE QUADROS

Quadro 1 – Resumo dos métodos de VND apresentados.	42
Quadro 2 – Resumo de artigos que utilizam agrupamento como estratégia de resolução. . . .	49
Quadro 3 – Resumo dos artigos que abordam o CluVRP.	52

LISTA DE ALGORITMOS

Algoritmo 1 – VARIABLE NEIGHBORHOOD SEARCH	39
Algoritmo 2 – VARIABLE NEIGHBORHOOD DESCENT	39
Algoritmo 3 – VND	70
Algoritmo 4 – VND com múltiplos inícios - VND_MI	71
Algoritmo 5 – Algoritmo da Perturba_Solução.	73

LISTA DE TABELAS

Tabela 1 – Resumo dos resultados obtidos.	62
Tabela 2 – Resultados obtidos para as instâncias do Grupo A.	64
Tabela 3 – Resultados obtidos para as instâncias do Grupo B.	65
Tabela 4 – Resultados obtidos para as instâncias do Grupo P.	66
Tabela 5 – Resultados obtidos para as instâncias do Grupo P.	67
Tabela 6 – Melhoria obtida com as heurísticas implementadas.	74
Tabela 7 – Resultados detalhados obtidos pela heurística desenvolvida e o modelo PRVCFF para as instâncias do grupo A.	86
Tabela 8 – Resultados detalhados obtidos pela heurística desenvolvida e o modelo PRVCFF para as instâncias do grupo B.	87
Tabela 9 – Resultados detalhados obtidos pela heurística desenvolvida e o modelo PRVCFF para as instâncias do grupo P.	88

LISTA DE ABREVIATURAS E SIGLAS

PRV	Problema de Roteamento de Veículos
PRVC	Problema de Roteamento de Veículos Capacitados
PRVCFE	PRVC com Custos de Fronteira Fixos
PRVCFP	PRVC com Custos de Fronteira Proporcionais
VND	<i>Variable Neighborhood Descent</i>
VNS	<i>Variable Neighborhood Search</i>

1	INTRODUÇÃO	25
1.1	Questão de Pesquisa	25
1.2	Contribuição da Pesquisa	26
1.3	Estrutura da Dissertação	26
2	PROBLEMA DE ROTEAMENTO DE VEÍCULOS	29
2.1	Modelo matemático	32
2.1.1	<i>Heurísticas Construtivas</i>	32
2.1.2	<i>Heurística construtiva de economias</i>	32
2.1.3	<i>Inserção do Mais Distante</i>	33
2.2	Vizinhanças	34
2.2.1	<i>Vizinhanças Inter-Rotas</i>	35
2.2.2	<i>Vizinhanças Intra-Rotas</i>	36
2.3	VNS	38
2.4	Revisão Bibliográfica	39
2.5	Considerações Finais	42
3	PROBLEMA DE ROTEAMENTO DE VEÍCULOS COM CLIENTES AGRUPADOS	43
3.1	Agrupamento como Estratégia de Resolução	43
3.2	CluVRP	48
3.3	Considerações finais	51
4	PROBLEMA ESTUDADO	53
4.1	Modelos matemáticos	54
4.1.1	<i>PRVC com Custos de Fronteira Fixos (PRVCFF)</i>	55
4.1.2	<i>PRVC com Custos de Fronteira Proporcionais (PRVCFP)</i>	56
4.1.3	<i>Desigualdade triangular</i>	57
4.2	Exemplo ilustrativo	58
4.3	Experimentos Computacionais	59
4.3.1	<i>Geração de Instâncias</i>	59
4.4	Resultados Computacionais	60
4.5	Considerações Finais	63
5	MÉTODO HEURÍSTICO	69

5.1	Variable Neighborhood Descent	69
5.2	VND com múltiplos inícios	70
5.2.1	<i>Solução Inicial</i>	71
5.2.2	<i>VND - Movimento</i>	71
5.2.3	<i>Perturba_Solução</i>	72
5.2.4	<i>Aplicação da heurística</i>	72
5.3	Resultados computacionais	73
5.3.1	<i>Análise de Resultados - Etapa 1</i>	73
5.3.2	<i>Análise de Resultados - Etapa 2</i>	75
5.4	Considerações finais	77
6	CONCLUSÕES	79
	REFERÊNCIAS	81
APÊNDICE A	RESULTADOS VND	85

INTRODUÇÃO

De acordo com a Confederação Nacional do Transporte, o custo logístico no Brasil, isto é, gastos com transporte, estoque, armazenagem e serviços administrativos, consumiu 12,7% do PIB do país no ano de 2016 ([CONFEDERAÇÃO NACIONAL DO TRANSPORTE, 2016](#)). Portanto, cada vez mais, o processo logístico vem se tornando importante para as grandes empresas se manterem competitivas no mercado.

Para algumas empresas, grande parte do gasto provém do transporte de produtos. Por exemplo, de acordo com [Golden e Wasil \(1987\)](#), o custo de distribuição pode representar até 70% do valor de uma bebida. Dessa forma, qualquer melhoria logística de um produto pode representar uma economia significativa, resultando em alto impacto nos preços aos consumidores.

Na literatura, a primeira vez em que se foi estudado o problema de atender múltiplos clientes com uma frota de veículos foi no artigo de [Dantzig e Ramser \(1959\)](#). Nesse artigo, os autores propõem uma formulação do problema de roteamento de veículos (PRV) para o planejamento da distribuição de gasolina em postos automotivos. O problema consiste em garantir que uma frota de veículos com capacidade limitada atenda a demanda de um conjunto de clientes. O objetivo é encontrar rotas que minimizem a distância total percorrida pelos veículos.

Desde sua publicação, o estudo em roteamento de veículos teve um crescimento exponencial até o ano de 2005 ([EKSIOGLU; VURAL; REISMAN, 2009](#)). Nesse período, diversas variantes do PRV surgiram para representar as necessidades de diferentes situações como por exemplo, janelas de tempo, frota de veículos heterogêneos, entregas fracionadas, entre outras. Uma ampla revisão sobre as inúmeras variações do problema foi realizada por [Toth e Vigo \(2014\)](#).

1.1 Questão de Pesquisa

Neste trabalho, é proposto o estudo de um PRV inspirado em um caso real. O Brasil é um país de proporção continental, por isso algumas empresas de distribuição têm a necessidade de atender clientes em vários estados do país. Devido à vasta área para a qual é necessário determinar as rotas, o

planejamento da sequência de clientes que cada veículo deve atender se torna uma tarefa árdua sem o auxílio de ferramentas de apoio à decisão.

Em especial, neste trabalho é tratada de forma explícita a questão da inspeção fiscal que ocorre na fronteira entre estados. A inspeção fiscal em fronteiras é necessária para evitar a evasão tributária ([PROTOCOLO-ICMS-10/03, 2003](#)). Logo, é obrigatório emitir um passe fiscal e, ao trocar de estado, é necessário que o veículo pare no posto fiscal para que se faça a checagem da documentação e a conferência das mercadorias transportadas. Esse processo, no entanto, interfere no tempo necessário para o veículo viajar entre estados dificultando assim a elaboração de rotas de tempo mínimo.

1.2 Contribuição da Pesquisa

Neste trabalho, é estudado o Problema de Roteamento de Veículos Capacitados (PRVC) que inclui a característica de auditorias de veículos em fronteiras. Duas propostas utilizando métodos exatos são avaliadas. A primeira consiste em adicionar custos fixos em trajetos que incluem o cruzamento de fronteira. A segunda penaliza a mudança de estados considerando a quantidade de carga do veículo ao cruzar a fronteira. Para analisar como a inclusão dessa nova característica impacta os resultados, o estudo é realizado em instâncias do problema de roteamento de veículos capacitados (PRVC) adaptadas. Na primeira abordagem, os custos adicionais para trocar de estado são aplicados em um modelo clássico de roteamento de veículos capacitado de dois índices. No segundo estudo, foi inserido um novo conjunto de restrições para modelar o custo proporcional a carga no momento de cruzar as fronteiras.

Como uma maneira alternativa para abordar o problema, métodos heurísticos foram desenvolvidos para obter soluções em menor tempo computacional. Devido aos bons resultados reportados na literatura, a heurística estudada foi a *Variable Neighborhood Search* (VNS). O método implementado foi sua variante *Variable Neighborhood Descent* (VND) com uma rotina que gera múltiplos inícios para aplicar a VND. Essa geração de múltiplos inícios é realizada perturbando a solução inicial, que é construída pela heurística de economias de [Clarke e Wright \(1964\)](#) ou Inserção do Mais Distante (do inglês, *farthest insertion*) ([JOHNSON; MCGEOCH, 2007](#)). A heurística desenvolvida para os casos considerando os custos de fronteira fixos e proporcionais a carga. Apesar da simplicidade da heurística proposta, os resultados obtidos por ela foram melhores que os apresentados pelos métodos exatos com tempo limite de 3.600 segundos.

1.3 Estrutura da Dissertação

Esta dissertação está estruturada da seguinte forma: o Capítulo 2 descreve o problema de roteamento de veículos e no capítulo seguinte (Capítulo 3) é apresentada a bibliografia estudada durante a pesquisa. No Capítulo 4, é descrito o problema proposto, além de serem discutidas as duas modelagens estudadas e seus resultados. No Capítulo 5, é detalhada a heurística desenvolvida juntamente com os testes computacionais. Por fim, as conclusões finais e as propostas para pesquisas

futuras são apresentadas no último capítulo.

PROBLEMA DE ROTEAMENTO DE VEÍCULOS

O Problema de Roteamento de Veículos (PRV) foi originalmente proposto por [Dantzig e Ramser \(1959\)](#) para encontrar rotas ótimas de entrega de combustível para postos de gasolina. Segundo [Battarra, Erdogan e Vigo \(2014\)](#), existem muitas variantes do PRV que são uma das classes de problemas de otimização combinatória mais estudadas na literatura. Em geral, o PRV pode ser caracterizado por um conjunto de solicitações de entregas de clientes que devem ser atendidas por uma frota de veículos. Uma solução viável para o problema é caracterizada por um conjunto de rotas que atendem todas as demandas dos clientes. O objetivo mais comum visa minimizar o custo total de transporte associado às rotas ([TOTH; VIGO, 2014](#)).

Mais formalmente, o PRV é definido por um conjunto de N clientes ($\mathcal{N} = \{1, \dots, N\}$) que devem ser atendidos. Cada cliente i possui uma demanda d_i que, por exemplo, representa o peso das mercadorias demandadas. A frota de veículos é composta por K veículos que estão disponíveis para realizar as entregas ($\mathcal{K} = \{1, \dots, K\}$). Todos os veículos, que são idênticos e possuem capacidade Q , devem começar e terminar suas rotas nos depósitos que possuem suas localizações pré-determinadas. É muito frequente haver um único depósito.

O PRV pode ser representado por um grafo. Supondo que há apenas um depósito que é representado pelo nó 0, o conjunto de vértices do grafo é definido como $V = \{0\} \cup \mathcal{N}$ sendo $d_0 = 0$. O conjunto de arestas é definido como $E = \{(i, j) | i \in V, j \in V, i \neq j\}$. Portanto, o grafo definido por $G = (V, E)$ é completo, pois cada vértice possui uma aresta que o conecta a todos os outros vértices do grafo, como ilustrado na Figura 1. Nesta figura, o depósito é o vértice representado por um quadrado (vermelho) e os clientes são os círculos (azuis).

A Figura 2 ilustra um exemplo baseado na instância C201 de [Solomon \(1987a\)](#) considerando apenas os primeiros 25 clientes. O depósito é novamente representado por um quadrado, enquanto os 25 clientes da instância são representados por círculos. Cada nó do grafo da Figura 2 possui uma coordenada x e uma coordenada y que correspondem a sua localização. Apesar de possuir uma aresta entre cada par de nós, essas não são ilustradas na figura para torná-la mais clara.

Uma rota é definida como uma sequência $r = (v_0, v_1, \dots, v_s, v_{s+1})$ de nós do grafo G , no qual

Figura 1 – Um exemplo de grafo completo com 6 vértices.

Fonte: Elaborada pelo autor.

Figura 2 – Exemplo de um grafo para um problema de roteamento de veículos com 25 clientes.

Fonte: Solomon (1987b).

v_0 e v_{s+1} correspondem ao depósito (nó 0) e s é o número de clientes visitados por essa rota. A cada rota r é associado um custo que é definido como $c(r) = \sum_{i=0}^s c_{i,i+1}$. Além disso, a demanda atendida em uma rota não pode exceder a capacidade do veículo, isto é, todas as rotas devem satisfazer $\sum_{i=1}^s d_i \leq Q$.

Na Figura 3, estão representadas duas possíveis soluções para o problema da Figura 2. A solução ilustrada na Figura 3a utiliza apenas uma rota para fazer todas as entregas, enquanto a solução ilustrada na Figura 3b é composta por duas rotas. Conforme a definição, essas rotas possuem uma sequência de nós que representam o caminho que cada veículo deve percorrer, e ambas possuem seu início e fim no depósito.

O problema de roteamento de veículos é muito estudado na literatura, dessa forma, existem diversas revisões sobre o assunto. Uma ampla revisão sobre o problema e suas variantes, bem como métodos exatos e heurísticos para sua resolução pode ser encontrada no livro de Toth e Vigo (2014). Boas revisões podem também ser encontradas em (LAPORTE, 2009), (PARRAGH; DOERNER; HARTL, 2008), (CORDEAU *et al.*, 2002) e (BRAEKERS; RAMAEKERS; NIEUWENHUYSE, 2016).

Figura 3 – Duas possíveis soluções para o PRV para a Figura 2.

(a) Exemplo de solução utilizando um veículo.

(b) Exemplo de solução utilizando dois veículos.

Fonte: Elaborada pelo autor.

O PRV é um problema NP-Difícil (LENSTRA; KAN, 1981). Portanto, resolver instâncias reais de grandes dimensões utilizando métodos exatos em tempo computacional aceitável se torna uma tarefa árdua. Dessa forma, as heurísticas possuem um papel importante para resolver instâncias de tamanho real.

De acordo com Cordeau *et al.* (2002), para avaliar heurísticas, devem ser considerados quatro critérios: precisão, velocidade, simplicidade e flexibilidade. Precisão é o resultado final da solução obtida pela heurística. A velocidade de uma heurística normalmente depende da aplicação do PRV. Simplicidade está relacionada a dificuldade de implementação da heurística desenvolvida e a flexibilidade está associada a possibilidade de incorporar novas restrições. De acordo com os autores, os critérios de precisão e velocidade são comumente avaliados, entretanto, os dois últimos não costumam ser apresentados.

Nesse sentido, heurísticas construtivas são, em geral, simples e fornecem soluções rapidamente. No entanto, nem sempre as soluções obtidas estão suficientemente próximas da solução ótima, ou seja, nem sempre possuem uma boa precisão. As metaheurísticas obtêm resultados melhores, mas demandam mais tempo computacional e frequentemente são mais elaboradas.

Muitas metaheurísticas utilizam buscas locais em soluções para obter mínimos locais (intensificação) e utilizam perturbações para sair destes mínimos locais em buscas de outros (diversificações). Esses processos de intensificação e diversificação apresentaram resultados satisfatórios para problemas de otimização combinatória. Uma dessas metaheurísticas que tem apresentado bons resultados para o PRV e possui grande flexibilidade e simplicidade ao ser implementada é a *Variable Neighborhood Search* (VNS).

Devido a vasta literatura na área, a revisão bibliográfica aqui apresentada está restrita aos artigos mais relevantes para o estudo proposto. Do ponto de vista de modelagem, é apresentado apenas o modelo de dois índices. Em seguida, são resumidos os trabalhos que propõem heurísticas VNS para resolução do PRV. Antes, no entanto, a fim de facilitar a compreensão das heurísticas, são descritas de

forma sucinta as heurísticas construtivas utilizadas como solução inicial, as vizinhanças e o esquema geral de uma VNS.

2.1 Modelo matemático

Nesta seção é apresentada uma formulação de dois índices por fluxo de veículos (IRNICH; TOTH; VIGO, 2014) para o PRV. As variáveis x_{ij} assumem valor 1 se um veículo realiza a viagem do cliente i para o cliente j e 0 caso contrário.

$$\min \sum_{i \in \mathcal{N} \cup \{0\}} \sum_{j \in \mathcal{N} \cup \{0\}} T_{ij} x_{ij} \quad (2.1)$$

$$\text{s.a.} \quad \sum_{j \in \mathcal{N} \cup \{0\}} x_{ij} = 1 \quad \forall i \in \mathcal{N} \cup \{0\} \quad (2.2)$$

$$\sum_{i \in \mathcal{N} \cup \{0\}} x_{ij} = 1 \quad \forall j \in \mathcal{N} \cup \{0\} \quad (2.3)$$

$$\sum_{j \in \mathcal{N}} x_{0j} \leq K \quad (2.4)$$

$$\sum_{\substack{i, j \in \mathcal{N}: \\ i \in S, j \notin S}} x_{ij} \geq r(S) \quad \forall S \subset \mathcal{N}, S \neq \emptyset \quad (2.5)$$

$$x_{ij} \in \{0, 1\} \quad \forall i, j \in \mathcal{N} \cup \{0\} \quad (2.6)$$

A função objetivo (2.1) visa minimizar o tempo de viagem de todos os veículos. O parâmetro T_{ij} é o tempo de viagem de um cliente i para o cliente j com os custos adicionais entre clientes de estados diferentes. As restrições (2.2) e (2.3) garantem o fluxo dos veículos entre os clientes e o depósito. A restrição (2.4) restringe o número máximo de veículos a serem utilizados. As restrições (2.5) eliminam subciclos no grafo e garantem o sequenciamento de carga dos veículos. Nessas restrições, S é um subconjunto de todos os vértices $\mathcal{N}$ e $r(S)$ representa o número mínimo de veículos necessários para atender os clientes no conjunto S . Por fim, as restrições (2.6) definem as variáveis x_{ij} como binárias.

2.1.1 Heurísticas Construtivas

Nesta seção, são apresentadas duas heurísticas construtivas, frequentemente utilizadas para geração de soluções iniciais para o PRV. Estas heurísticas constroem, a partir de uma solução vazia, uma solução completa para o problema. A primeira é a heurística construtiva de Clarke e Wright (1964) e a segunda foi proposta por Johnson e McGeoch (2007), chamada de inserção do mais distante (traduzido do inglês, *farthest insertion*).

2.1.2 Heurística construtiva de economias

A heurística construtiva de economias foi proposta por Clarke e Wright (1964). Ela é uma heurística de simples implementação com bons resultados para o PRV (CORDEAU *et al.*, 2002).

A heurística consiste em, inicialmente, atribuir uma rota a cada cliente individualmente. Na etapa seguinte, calcula-se as economias de unir dois clientes em uma mesma rota. O cálculo das economias é dado por:

$$s_{ij} = c_{0i} + c_{j0} - c_{ij}, \quad \forall i, j \in \mathcal{N}. \quad (2.7)$$

em que s_{ij} representa a economia obtida ao unir os clientes i e j em uma mesma rota; no lado direito da equação, a última parcela é o custo de unir os clientes i e j , enquanto as duas primeiras parcelas são as viagens que seriam retiradas caso a alteração fosse realizada. Dessa forma, s_{ij} é positivo quando os custos de ir a cada um dos clientes e voltar é maior que ir de um cliente ao outro e, portanto, a alteração é vantajosa. Da mesma forma, caso s_{ij} seja negativo então é desvantajoso fazer a viagem entre os clientes e eliminar as viagens diretamente ao depósito. Após o cálculo das economias, as rotas vão sendo unidas em sequência da melhor economia até a menos vantajosa.

Na Figura 4, o procedimento da heurística é ilustrado. A Figura 4a ilustra a primeira etapa que consiste em atribuir a cada cliente uma rota que inicia e é finalizada no depósito. As figuras seguintes ilustram as etapas posteriores em que duas viagens até o depósito são trocadas por uma viagem entre os clientes. Por exemplo, na Figura 4b as viagens (3, 1) e (1, 5) são trocadas pela viagem (3, 5).

Figura 4 – Etapas da heurística de Clarke e Wright (1964)

Fonte: Elaborada pelo autor.

2.1.3 Inserção do Mais Distante

A segunda heurística construtiva aqui apresentada é a Inserção do Mais Distante que foi proposta para o PRV por Johnson e McGeoch (2007). Essa heurística consiste em iniciar a primeira rota com o cliente mais longe do depósito. O próximo cliente escolhido a ser adicionado a rota é aquele que estiver mais distante dos vértices da rota atual. Esse cliente é então adicionado a rota de

forma a causar o menor impacto em seu custo total. Essas iterações são realizadas de forma a não violar as restrições do problema.

A Figura 5 ilustra as etapas da Inserção do Mais Distante. Na primeira etapa, é adicionado na primeira rota, o cliente mais distante do depósito, neste caso o cliente 5. Na etapa seguinte, ilustrada na Figura 5b, o cliente 1 é o cliente mais distante do cliente 5 e, portanto, é adicionado a rota. Como não é possível adicionar mais clientes à rota devido a capacidade do veículo, o veículo retorna ao depósito. A Figura 5c ilustra o processo sendo iniciado novamente e a Figura 5d mostra a solução final.

Figura 5 – Etapas da heurística Inserção do Mais Distante

Fonte: Elaborada pelo autor.

Apesar dessa heurística possuir uma decisão local de colocar o cliente mais distante da rota, essa decisão pode não resultar necessariamente em um prejuízo ao custo da solução final. Como os clientes mais distantes devem estar em alguma das rotas, colocá-los no início gera uma rota que, eventualmente, ao inserir outros clientes, não resultam em um grande impacto na solução como um todo (LAWLER *et al.*, 1985).

2.2 Vizinhos

A vizinhança de uma solução é definida por todas as possíveis soluções que podem ser obtidas realizando uma perturbação específica na solução, ou seja, um movimento. Diferentes vizinhanças foram propostas na literatura para o PRV. Elas são divididas em duas categorias: inter-rotas e intra-rotas. As vizinhanças intra-rotas consistem em perturbar uma única rota da solução. Por outro lado, as vizinhanças inter-rotas perturbam pelo menos duas rotas da solução. Em todas as vizinhanças, em cada movimento da vizinhança, as restrições do problema são consideradas, ou seja, o movimento é realizado apenas se não violar nenhuma restrição e se houver uma melhoria nos custos das rotas.

2.2.1 Vizinhanças Inter-Rotas

Para ilustrar os movimentos das vizinhanças inter-rotas, a Figura 6 é utilizada como exemplo. As vizinhanças apresentadas nessa seção são *2-opt* inter-rotas, a classe de vizinhanças λ *exchange* e *cross exchange*.

Figura 6 – Exemplo com duas rotas

Fonte: Elaborada pelo autor.

- *2-opt* inter-rotas:

Uma vizinhança comumente utilizada em buscas locais de roteamento de veículos é *2-opt* inter-rotas. Primeiramente proposto por [Potvin et al. \(1996\)](#) e semelhante à vizinhança *2-opt* intra-rotas, um movimento da vizinhança *2-opt* inter-rotas consiste em remover dois arcos e realocá-los. A diferença consiste em remover arcos de rotas diferentes. A Figura 7 ilustra um movimento *2-opt*. Nesse caso, dois arcos de rotas diferentes foram removidos e dois arcos foram recolocados. No exemplo, os arcos (2,3) e (7,8) foram removidos e trocados pelos arcos (7,3) e (2,8). Dessa forma, cada rota é dividida em duas sub-rotas que são trocadas.

Figura 7 – Um movimento 2-opt inter-rotas

Fonte: Elaborada pelo autor.

- λ *interchange*:

λ *interchange* é uma classe de vizinhança muito utilizada que foi proposta por [Osman \(1993\)](#). O movimento dessa vizinhança consiste em trocar até λ clientes de rotas diferentes. A notação para uma vizinhança dessa classe é da seguinte forma: λ_1, λ_2 *exchange*. Considerando que as rotas r_1 e r_2 sejam alteradas e $\lambda_1, \lambda_2 \leq \lambda$, λ_1 clientes são removidos da rota r_1 e colocados na rota r_2 e, analogamente, λ_2 clientes são removidos da rota r_2 e colocados na rota r_1 . Um exemplo é a vizinhança 1-0 *exchange* que consiste em remover um cliente de uma rota e colocá-lo em uma outra. A Figura 8a ilustra um movimento dessa vizinhança. O cliente 2 pertencia a uma rota e após o movimento foi realocado para outra rota. Semelhante ao 1-0 *exchange*, na 1-1 *exchange* dois clientes de rotas diferentes são trocados. A Figura 8b ilustra um movimento dessa vizinhança. Note que desta vez, os clientes 2 e 9 foram trocados de rotas.

Figura 8 – Movimentos *1-0 exchange* e *1-1 exchange*.(a) Um movimento do *1-0 exchange* (b) Um movimento do *1-1 exchange*

Fonte: Elaborada pelo autor.

- *cross exchange*:

A vizinhança *cross exchange* é semelhante à *2-opt* inter-rotas. Nas duas vizinhanças, o movimento troca uma sequência de clientes entre duas rotas diferentes. Na vizinhança *cross exchange*, dadas duas rotas, em cada uma é selecionada uma sub-rota com tamanho pré-determinado. Em seguida, as duas sub-rotas são trocadas nas rotas originais. Essa troca de sub-rotas implica em uma modificação de quatro arestas, enquanto no movimento *2-opt* duas são alteradas. A Figura 9a ilustra um movimento *2-opt* entre duas rotas e a Figura 9b ilustra um movimento de um *cross-exchange*. No exemplo ilustrado, uma sub-rota possui tamanho 2 e a outra sub-rota possui tamanho 3, respectivamente, (2,3) e (7,8,9).

Figura 9 – Diferença entre o *2-opt* e o *cross-exchange*.(a) Um movimento *2-opt* em uma rota. (b) Um movimento sem a inversão no *2-opt* em uma rota.

Fonte: Elaborada pelo autor.

2.2.2 Vizinhanças Intra-Rotas

Nesta seção, são apresentadas cinco vizinhanças intra-rotas. A primeira apresentada é a *2-opt*, em seguida a *Or-opt*, *iopt*, *swap* e, por último, *reverse*.

- *2-opt* intra-rota:

A vizinhança mais comum intra-rota é a *2-opt*. Um movimento *2-opt* intra-rota consiste em realizar a remoção e a colocação de dois arcos de uma mesma rota. Esse movimento impõe, no entanto, que seja realizada uma inversão do trajeto dentro da rota para que a solução continue factível.

A Figura 10a ilustra um exemplo de uma única rota em que é realizado um movimento da vizinhança *2-opt* intra-rota. A Figura 10b ilustra um movimento realizado utilizando os arcos (2,3) e (7,8) que são removidos e, em seguida, são adicionados os arcos (2,8) e (7,3). Entretanto,

como o movimento é realizado em apenas uma rota, a viagem realizada entre os clientes 3 e 8 deve ser invertida para que a rota seja factível. A Figura 10c ilustra a rota infactível que seria obtida caso a sub-rota (3, ..., 8) não fosse invertida.

Figura 10 – Movimento 2-opt intra-rota

- Or-opt:

Outro exemplo de vizinhança intra-rota é Or-opt (OR, 1977). Neste caso, é realizado o deslocamento de um subconjunto de clientes de uma mesma rota para uma outra posição de rota. O movimento dessa vizinhança é realizado da seguinte forma: é escolhido um valor de $k \in \{1, 2, 3\}$ e, em seguida, é realocado uma sub-rota de k clientes para uma posição diferente da que foi retirada na rota. A Figura 11a ilustra um movimento dessa vizinhança. Nessa figura, k é igual a 2, portanto dois clientes são deslocados (neste exemplo, os clientes 2 e 3).

Figura 11 – Movimento Or-opt.

- iopt:

A vizinhança iopt (BRÄYSY, 2003) é uma generalização do Or-opt. Da mesma forma como o Or-opt, essa vizinhança consiste em realocar uma sequência de clientes de uma rota para outra posição dentro da mesma. Entretanto, a vizinhança iopt considera a inserção dos clientes preservando a sequência ou invertendo-a. É escolhido o caso que obtiver a melhor rota. A Figura 12 ilustra um movimento da vizinhança no caso invertendo a sequência dos clientes inseridos. Na Figura 12a mostra os mesmos clientes a serem realocados que no caso da vizinhança Or-opt. A Figura 12b ilustra a realocação dos clientes dos clientes 2 e 3.

- swap:

Essa vizinhança consiste em trocar dois clientes dentro de uma mesma rota. Essa vizinhança é uma versão intra-rota da vizinhança 1-1 *exchange* inter-rotas. A Figura 13 ilustra um movimento dessa vizinhança.

Figura 12 – Movimento iopt.

Fonte: Elaborada pelo autor.

Figura 13 – Movimento *swap*

Fonte: Elaborada pelo autor.

- *reverse*:

Essa vizinhança inverte a sequência dos clientes que são atendidos em uma rota. Essa vizinhança foi encontrada apenas em um problema de roteamento de veículos com coleta e entrega simultânea (*vehicle routing problem with simultaneous pickup and delivery*, em inglês). Nesse caso, o movimento é considerado apenas se carga máxima do veículo durante a rota é reduzida.

2.3 VNS

A *Variable Neighborhood Search* (VNS) foi proposta por [Mladenović e Hansen \(1997\)](#). O método consiste em realizar uma busca através de uma mudança sistemática de vizinhanças tendo como objetivo obter melhores mínimos locais. De acordo com [Hansen e Mladenović \(2014\)](#), a VNS é baseada em três fatos:

- O mínimo local em relação a uma estrutura de vizinhança não é necessariamente o mínimo local para uma vizinhança diferente.
- O mínimo global é um mínimo local em relação a todas as vizinhanças possíveis.
- Para muitos problemas, os mínimos locais em relação a algumas vizinhanças são relativamente próximos.

A heurística, da forma proposta em ([MLADENOVIĆ; HANSEN, 1997](#)), inicia selecionando o conjunto de vizinhanças a ser utilizado ($\mathcal{V}_k, k = 1, \dots, k_{max}$), uma solução inicial (x') e as condições de parada. Assim, a VNS da forma proposta pelos autores é resumida no Algoritmo 1.

De acordo com [Hansen e Mladenović \(2001\)](#), para algumas aplicações em que o ótimo local de uma vizinhança não está necessariamente dentro de outra vizinhança (ou seja, não são os mesmos),

Algoritmo 1 – VARIABLE NEIGHBORHOOD SEARCH

```

1: procedimento VNS(vizinhanças, Dados)
2:    $x \leftarrow \text{GeraçãoSoluçãoInicial}()$ 
3:   enquanto Condição de parada seja satisfeita faça
4:      $k \leftarrow 1$ 
5:     enquanto  $k \leq k_{max}$  faça
6:       Shaking: Gere um ponto aleatório  $x'$  vizinho de  $x$  da  $k$ -ésima vizinhança
7:       Local Search: Aplique uma busca local com  $x'$  como ponto inicial e atribua a solução
         obtida a  $x''$ 
8:       Move or Not: Se  $\text{valor}(x'') < \text{valor}(x)$ , atualize a melhor solução ( $x \leftarrow x'$ ) e reinicialize
         as vizinhanças ( $k \leftarrow 1$ ). Do contrário, atualize  $k$ , ou seja, faça  $k \leftarrow k + 1$ .
9:     fim enquanto
10:  fim enquanto
11:  retorna melhor-solução
12: fim procedimento

```

a troca de vizinhanças pode ser realizada durante a busca local. O Algoritmo 2 resume essa variante que é chamada de *Variable Neighborhood Descent* (VND).

São considerados dados a solução inicial e as vizinhanças utilizadas. No Algoritmo 1, em cada vizinhança é gerado um ponto aleatório (linha 6) e, em seguida, uma busca local é realizada no novo ponto (linha 7). O processo *shaking* (linha 6) é realizado para evitar que ocorram ciclagens durante a busca (MLADENOVIĆ; HANSEN, 1997). Apenas ao fim da busca local a troca de vizinhanças é feita (linha 8). Por outro lado, no Algoritmo 2, a troca de vizinhanças ocorre durante a busca local (linhas 5 e 6). A troca de vizinhanças é realizada nos dois algoritmos da mesma forma: se a nova solução obtida for melhor que a atual, então a vizinhança é reiniciada ($k \leftarrow 1$). Do contrário, a busca é realizada na próxima vizinhança ($k \leftarrow k + 1$).

Algoritmo 2 – VARIABLE NEIGHBORHOOD DESCENT

```

1: procedimento VND(vizinhanças, Dados)
2:    $x \leftarrow \text{GeraçãoSoluçãoInicial}()$ 
3:    $k \leftarrow 1$ 
4:   enquanto  $k \leq \text{tamanho}(\text{vizinhanças})$  faça
5:     First or Best Neighbor: É obtido o primeiro ou o melhor vizinho de  $x$  em relação a
       vizinhança  $k$  e atribuído a  $x''$ 
6:     Move or Not: Se  $\text{valor}(x'') < \text{valor}(x)$ , atualize a melhor solução ( $x \leftarrow x'$ ) e reinicialize as
       vizinhanças ( $k \leftarrow 1$ ). Do contrário, atualize  $k$ , ou seja, faça  $k \leftarrow k + 1$ 
7:   fim enquanto
8:   retorna  $x$ 
9: fim procedimento

```

2.4 Revisão Bibliográfica

Durante a pesquisa, não foram encontrados artigos que tratam o PRV como proposto neste trabalho, logo a revisão bibliográfica foi restrita aos trabalhos em que métodos VND foram desenvolvidos para tratar o PRV capacitado.

Kytöjoki *et al.* (2007) foram os primeiros a implementar com sucesso uma VND para o PRV capacitado para instâncias de grande porte. Os autores utilizaram uma VND na etapa de construção da solução inicial e posteriormente como busca local. Na etapa de busca local, os autores utilizam uma perturbação de *Guided Local Search* e, por isso, denominam a busca local de GVNS.

Para a solução inicial, é determinado se os clientes estão posicionados de forma simétrica ou assimétrica. Em seguida, para gerar as rotas é escolhido um cliente de forma aleatória e os clientes seguintes vão sendo inseridos na rota de forma a causar o menor impacto na distância total da rota. Depois de k clientes inseridos, é realizado um procedimento de VND com as vizinhanças intra-rotas (o valor de k depende se os clientes são simétricos ou assimétricos em relação ao depósito). Esse procedimento se repete a cada duas rotas finalizadas e, em seguida, é realizado um procedimento de VND com as vizinhanças inter-rotas. Na etapa de busca local é realizado um procedimento VND com as vizinhanças inter-rotas e intra-rotas da seguinte forma: para cada par de rotas é executada uma VND com as vizinhanças inter-rotas até que não haja nenhuma melhora nas duas rotas. Em seguida, é realizada uma VND com as vizinhanças intra-rotas. A VND com vizinhanças inter-rotas na busca local é aplicada de forma diferente da VND aplicada na construção da solução inicial, pois todas as rotas são utilizadas na busca local enquanto durante a construção de uma solução, as rotas utilizadas são as duas últimas.

Conforme mencionado, os autores utilizam como estratégia de perturbação uma *Guided Local Search* (GLS). Uma GLS consiste em penalizar uma característica do problema após obter um mínimo local (VOUDOURIS; TSANG, 2010). Em (KYTÖJOKI *et al.*, 2007), os autores utilizam a GLS após a busca local realizada em cada par de rotas se houver uma melhora. Esse procedimento repete os seguintes passos: penaliza-se uma viagem longa e, em seguida, é executada uma nova VND com as vizinhanças inter e intra-rotas. Quando uma mesma viagem é penalizada um número de vezes determinado pelo usuário, a perturbação é finalizada e um novo par de rotas é escolhido para passar pela busca local.

Prins (2009) desenvolveu uma heurística *iterated local search* (ILS) que utiliza múltiplas soluções de partida para o PRV. Em cada iteração da ILS, é utilizada uma solução inicial diferente. Na primeira iteração, é utilizada a heurística de Clarke e Wright (1964) para obter uma solução inicial. As demais soluções iniciais são obtidas utilizando a *Random Nearest Neighborhood Heuristic*. A busca local é semelhante ao método VND. As vizinhanças são utilizadas de forma sequencial e a busca para apenas quando não houver melhora em nenhuma das vizinhanças. A perturbação é uma *Evolutionary Local Search* (ELS). Para a busca local são utilizadas 5 vizinhanças: *2-opt*, *1-1 exchange*, *Or-opt*, *cross exchange*, e *crossover*. A última é uma vizinhança proposta pelo autor que consiste em transformar uma rota em duas sub-rotas que iniciem e terminem no depósito.

Chen, Huang e Dong (2010) desenvolveram uma heurística ILS que utiliza uma VND como busca local. Na perturbação é realizado um movimento aleatório da vizinhança *cross exchange* para tentar sair dos mínimos locais. Os autores chamam essa combinação de IVND. As vizinhanças utilizadas durante a VND são: *1-0 exchange*, *1-1 exchange*, *2-opt* inter-rotas e *2-opt* intra-rotas. A heurística construtiva de Clarke e Wright (1964) é utilizada como solução inicial do método.

Subramanian *et al.* (2010) abordam o PRV com coletas e entregas simultâneas (em inglês, *Vehicle Routing Problem with Simultaneous Pickup and Delivery*, VRPSPD). Para sua resolução foi desenvolvido um algoritmo paralelo que combina as metaheurísticas ILS e uma variação da VND. Em resumo, o método pode ser descrito em três etapas. Primeiramente, uma solução inicial é obtida atribuindo-se aleatoriamente um cliente a um dos veículos disponíveis, tem-se assim a primeira rota inicial. Para cada um dos demais clientes é calculado o custo de inseri-lo em uma rota já existente ou criar uma nova rota (se ainda houver veículo disponível). É tomada a decisão de menor custo. Em seguida, três processos de perturbação são utilizados:

- *Ejection chain*: desloca um cliente aleatório de uma rota r_1 para uma rota r_2 , e em seguida um cliente aleatório de r_2 para r_3 e assim por diante. Esse processo é continuado até que o cliente da última rota volte para r_1 . Essa perturbação é realizada apenas quando há, no máximo, 12 rotas.
- *Double swap*: consiste na realização de dois movimentos 1-1 *exchange* de forma aleatória.
- *Double bridge*: um movimento da vizinhança 2-2 *exchange* é realizado de forma aleatória em cada rota. Se houver mais de 15 rotas, este processo é realizado apenas em $\frac{1}{3}$ das rotas para evitar perturbações exageradas.

Na terceira etapa, é realizada uma busca local utilizando uma VND. A cada passo da VND a ordem das vizinhanças é escolhida de forma aleatória. As vizinhanças inter-rotas utilizadas são: 1-0; 1-1; 2-0; 2-1; 2-2; *exchanges* e *cross exchange*; e intra-rotas: Or-opt, 2-opt, *swap* e *reverse*. Primeiramente, uma vizinhança de inter-rotas é escolhida e em seguida, o melhor movimento é feito. Se houve uma melhora, então é realizada uma busca em uma vizinhança de intra-rotas. Como dito anteriormente, esse processo continua até que não haja nenhuma melhora em nenhuma das vizinhanças de inter-rotas.

Subramanian *et al.* (2010), Subramanian, Uchoa e Ochi (2013) propuseram uma matheurística para tratar o problema. O algoritmo paralelo de Subramanian *et al.* (2010) é utilizado para gerar um *pool* de rotas viáveis. Em seguida, um modelo de particionamento de conjuntos é resolvido para encontrar as melhores rotas de forma que todos os clientes sejam atendidos. De acordo com Uchoa *et al.* (2017), os resultados apresentados por Subramanian, Uchoa e Ochi (2013) estão no estado da arte para o PRV capacitado.

Recentemente, Amous *et al.* (2017) desenvolveram uma VND para o PRV. Os autores utilizam um procedimento de *shake* em cada iteração da VND. Esse procedimento consiste em uma perturbação de uma das vizinhanças utilizadas que é escolhida de forma aleatória. As vizinhanças utilizadas pelos autores são a Or-opt, iopt, *cross exchange* e da classe de λ *interchange*. A solução inicial é gerada de maneira aleatória.

Quadro 1 – Resumo dos métodos de VND apresentados.

Autores	Solução Inicial	Vizinhança	Perturbação	Movimento	Variação
Kytöjoki et al. (2007)	Própria	<i>2-opt*</i> , <i>Or-opt*</i> , <i>3-opt*</i> , <i>1-0</i> ; <i>1-1 exchange</i> , <i>2-opt</i> , <i>cross exchange</i>	<i>guided local search</i>	Primeira descida	CVRP
Prins (2009)	Clarke e Wright (1964) , <i>Nearest Neighborhood Search</i>	<i>Or-opt*</i> , <i>2-opt*</i> , <i>2-opt</i> , <i>1-1 exchange</i> , <i>crossover</i> , <i>cross exchange</i>	<i>random exchange</i>	Primeira descida	CVRP, DVRP
Chen, Huang e Dong (2010)	Clarke e Wright (1964)	<i>2-opt*</i> , <i>1-0</i> ; <i>1-1 exchange</i> , <i>2-opt</i>	<i>random cross exchange</i>	Máxima descida	CVRP
Subramanian et al. (2010)	Própria	<i>Or-opt*</i> , <i>2-opt*</i> , <i>swap*</i> , <i>reverse*</i> , <i>1-1</i> ; <i>1-0</i> ; <i>2-0</i> ; <i>2-1</i> ; <i>2-2 exchanges</i> , <i>cross exchange</i>	<i>ejection chain</i> , <i>double swap</i> , <i>double bridge</i>	Primeira descida	CVRP, VRPSPD
Amous et al. (2017)	Aleatória	<i>iopt*</i> , <i>Or-opt*</i> , <i>cross exchange</i> , λ <i>inter-change</i>	<i>shake</i>	Primeira Descida	CVRP

2.5 Considerações Finais

A VNS é uma metaheurística de simples implementação e que tem apresentado bons resultados para o PRV e, por isso, foi priorizada durante a revisão bibliográfica deste trabalho. Todos os trabalhos encontrados na literatura que utilizam essa metaheurística fazem uso da variante VND. A Tabela 1 apresenta um resumo dos artigos aqui revisados. Na primeira coluna, tem-se os autores e as demais colunas trazem, respectivamente, a solução inicial, as vizinhanças, a perturbação, o tipo de estratégia da busca local, e a variante do PRV abordada. Na última coluna, são utilizadas as siglas em inglês, sendo elas: CVRP - *Capacitated Vehicle Routing Problem*, DVRP - *Distance-constrained Vehicle Routing Problem* e VRPSPD - *Vehicle Routing Problem with Simultaneous Pickup and Delivery*. As vizinhanças intra-rotas são destacadas com *.

PROBLEMA DE ROTEAMENTO DE VEÍCULOS COM CLIENTES AGRUPADOS

Dado que o problema em que foi inspirado este trabalho possui clientes divididos geograficamente em estados, nesta seção, são apresentados trabalhos da literatura em que o PRV apresenta clientes que podem ser divididos em grupos. Na literatura, foram encontrados estudos que utilizam a estratégia de agrupamento de clientes em duas situações. Na primeira, o agrupamento de clientes é uma estratégia de resolução para problemas com um elevado número de clientes, ou seja, não existe informação *a priori* que defina grupos de clientes. O agrupamento dos clientes é realizado para reduzir a dimensão do problema tratado e, consequentemente, resolvê-lo de forma mais eficiente. Na segunda, há informação sobre a região a que cada cliente pertence, logo o agrupamento de clientes é tratado como um dado inicial do problema. As seções seguintes descrevem os trabalhos encontrados considerando essas duas abordagens.

3.1 Agrupamento como Estratégia de Resolução

[Desrosiers, Dumas e Soumis \(1988\)](#) apresentaram um dos primeiros trabalhos em que a ideia de agrupamento de clientes foi utilizada como estratégia de resolução de um PRV. O trabalho aborda o transporte de deficientes físicos, mais especificamente, um problema de *dial-a-ride* com janelas de tempo. Logo, cada cliente deve ser embarcado em um local e desembarcado em outro, respeitando sua janela de tempo. A proposta de resolução do problema é baseada no método *cluster first, route second*, em que os agrupamentos são definidos inicialmente e, em seguida, para cada agrupamento é associada uma rota. A separação dos clientes em grupos é realizada para diminuir a dimensão do problema.

A heurística desenvolvida pelos autores é dividida em quatro etapas. Inicialmente, é obtida uma solução viável para o problema. Em seguida, na primeira etapa, cada rota da solução inicial é otimizada utilizando o método de programação dinâmica de [Desrosiers, Dumas e Soumis \(1986\)](#). Note que, em cada rota no momento em que o veículo estiver com um único cliente, ele pode ficar vazio se o desembarque desse cliente ocorrer antes do embarque do próximo cliente da rota. Os clientes que

ficam na trajetória entre dois pontos em que um veículo fica vazio durante a rota, definem o que os autores denominam *mini-cluster*. A cada *mini-cluster* é associada uma janela de tempo cujo início e término são, respectivamente, o início da janela de tempo do primeiro cliente a ser embarcado e do último a ser desembarcado dentro deste segmento de rota. Cada *mini-cluster* define um novo nó do problema, reduzindo, portanto, sua dimensão. Na terceira etapa, o PRV com o número de nós reduzido é resolvido utilizando um método de geração de colunas, que resulta nas melhores rotas entre os *mini-clusters*. Finalmente, na última etapa, é novamente aplicado o método de programação dinâmica em cada rota.

Na Figura 14 são apresentadas as quatro etapas do algoritmo. A Figura 14a mostra uma solução otimizada pelo método de programação dinâmica. A Figura 14b ilustra *mini-clusters* que são representados por cores diferentes e estão dentro de quadrados ou de círculos. Vale lembrar que um veículo está vazio cada vez que entra ou sai de um *mini-cluster*. Na terceira etapa, Figura 14c, é ilustrada uma solução para o roteamento considerando os vértices iniciais e finais de cada *mini-cluster*. Este PRV com um menor número de vértices é resolvido utilizando um método exato. A Figura 14d ilustra a solução final após completar as rotas em cada *mini-cluster*.

Figura 14 – Ilustração das quatro etapas do método de Desrosiers, Dumas e Soumis (1988).

Fonte: Elaborada pelo autor.

Kim, Kim e Sahoo (2006) abordam o problema de coleta de resíduos sólidos considerando o horário de almoço dos motoristas. Diferentemente de Desrosiers, Dumas e Soumis (1988) que

utilizaram grupos de clientes para reduzir a dimensão do problema, os autores utilizam esta estrutura para diminuir o número de rotas que se cruzam e balancear as horas de trabalho dos motoristas.

A ideia geral para criar os agrupamentos é escolher de forma aleatória V vértices (número de veículos) como centros dos grupos e, em seguida, os demais vértices são adicionados aos grupos cujo centro está mais próximo. Na etapa seguinte, é encontrado um novo centro para cada grupo considerando todos os seus vértices. A partir dos novos centros, é calculado o “grande centroide” que marca o centro geométrico de todos os grupos. Assim, começando pelo cliente mais distante do “grande centroide”, todos os clientes são realocados ao grupo cujo centroide esteja mais próximo. Esse processo de obter os centroides, o grande centroide e realocar os clientes é continuado até que não haja mais nenhuma mudança.

Os clientes são adicionados respeitando as restrições de capacidade de cada agrupamento, que não pode ultrapassar a capacidade dos veículos. Quando o grupo cujo centroide está mais próximo do cliente já atingiu seu limite de capacidade, então o cliente é alocado ao segundo grupo mais próximo. Uma outra restrição considerada ao adicionar um cliente a um grupo é a questão de janelas de tempo. Se um cliente, ao ser adicionado a um grupo, gerar um conflito de horário, então o cliente é adicionado a outro grupo. Após a alocação de todos os clientes, uma extensão da heurística de inserção de [Solomon \(1987b\)](#) proposta pelos autores é utilizada para criar uma rota para cada grupo, começando daquele com maior quantidade de clientes até o que possuir a menor quantidade. Assim que uma rota for obtida para um grupo, os clientes não atendidos são realocados a outros grupos.

Enquanto no trabalho de [Kim, Kim e Sahoo \(2006\)](#) é proposto um algoritmo em que é possível os clientes serem realocados na etapa de roteamento, [Dondo e Cerdá \(2007\)](#) constroem agrupamentos de forma que um mesmo veículo possa atender todos os clientes que pertencem a um mesmo grupo. Ou seja, depois da etapa de construção dos grupos, os clientes pertencem a um único grupo e não são realocados nas etapas seguintes assim como [Desrosiers, Dumas e Soumis \(1988\)](#). Em [Dondo e Cerdá \(2007\)](#), a utilização de agrupamentos é realizada para diminuir a dimensão de um problema de roteamento de veículos com janelas de tempo, múltiplos depósitos e veículos heterogêneos.

A heurística proposta pelos autores é composta por três fases. Na primeira, são criados os grupos de forma que seja factível um único veículo atender todos os clientes desse grupo. Para a criação dos grupos, os clientes são ordenados pelo início das janelas de tempo e, em seguida, são adicionados ao grupo se estas restrições não forem sobrepostas e as capacidades dos veículos não forem violadas.

Em seguida, é criada uma nova estrutura para que cada grupo se torne um cliente para o problema de roteamento de veículos. Portanto, visto que cada grupo passa a ser um vértice do grafo do PRV é necessário que todos tenham suas próprias janelas de tempo, tempos de serviço e demandas. Para tanto, é utilizado como início e fim das janelas de tempo, o horário mais cedo e mais tarde entre todos os clientes do grupo, respectivamente. A demanda é o acúmulo das demandas individuais dos clientes que pertencem ao grupo. E por fim, são calculados os centroides de cada grupo que representa a posição do grupo para o novo PRV. Na segunda fase, é utilizado um modelo de roteamento de veículos com janelas de tempo, múltiplos depósitos e veículos heterogêneos. Nesse modelo, são

utilizados os grupos como clientes resultando assim na diminuição do problema em relação ao original. Por fim, na terceira fase é realizado um roteamento de veículos dentro dos grupos respeitando as janelas de tempo de cada cliente.

As Figuras 15 e 16 ilustram a diferença entre os grupos que seriam gerados por Kim, Kim e Sahoo (2006) e Dondo e Cerdá (2007). Cada grupo é representado por uma cor e formatos diferentes e possui uma forma geométrica envolvendo os clientes. Na Figura 15a, estão representados os grupos inicialmente gerados no algoritmo de Kim, Kim e Sahoo (2006), enquanto a Figura 15b ilustra uma solução gerada pelos autores. Na Figura 16a, estão os clientes que poderiam ser gerados pelo algoritmo de Dondo e Cerdá (2007), enquanto a Figura 16b ilustra uma solução final. É possível notar que na solução final do algoritmo de Kim, Kim e Sahoo (2006) um cliente foi alterado de grupo, enquanto na solução final de Dondo e Cerdá (2007) todos os clientes permanecem em seu grupo inicial.

Figura 15 – Ilustração do método de Kim, Kim e Sahoo (2006).

Fonte: Elaborada pelo autor.

Figura 16 – Ilustração do método de Dondo e Cerdá (2007).

Fonte: Elaborada pelo autor.

Schuijbroek, Hampshire e Hoeve (2013) desenvolveram uma heurística *cluster first, route second* para resolver o balanceamento de estações públicas de bicicletas. Nessa classe de problemas,

o objetivo é manter todas as estações com uma quantidade suficiente de bicicletas. A quantidade de bicicletas necessária em cada estação (demanda) é definida a partir de dados históricos. Utilizando esses dados, grupos são gerados de forma que as estações com excesso de bicicletas possam repor as estações com falta.

A heurística proposta pelos autores consiste de três etapas. Na primeira etapa, é determinada a demanda de cada estação. A partir da demanda e do estoque de cada estação é possível determinar quais possuem excesso e falta de bicicletas. Na segunda etapa, os agrupamentos são gerados de forma que as demandas das estações possam ser atendidas por outras estações de um mesmo grupo, ou seja, os agrupamentos são criados para que as estações com excesso de bicicletas possam repor outras estações do agrupamento que tenham falta de bicicleta. Diferente dos trabalhos de [Kim, Kim e Sahoo \(2006\)](#) e [Dondo e Cerdá \(2007\)](#), cada agrupamento corresponde a uma rota completa de um determinado veículo. Tanto para encontrar a solução dos agrupamentos quanto para a criação de rotas são utilizados métodos exatos de programação linear inteira. O modelo para a criação dos agrupamentos é uma aproximação do *Maximum Spanning Star* ([THACH et al., 2008](#)). Uma vez definidos os agrupamentos, o PRV em que os nós são os grupos é resolvido tendo como objetivo minimizar as rotas atendendo as demandas de cada agrupamento. Uma vez definidos os agrupamentos, para cada veículo é associado um grupo e resolvido o roteamento no grupo utilizando um *software* comercial.

[Forma, Raviv e Tzur \(2015\)](#) também utilizam agrupamento no contexto de balanceamento de estoques de bicicletas em estações públicas. O problema abordado pelos autores é a reposição de bicicletas durante a noite na cidade de Paris. Caminhões são utilizados para levar bicicletas de estações com excesso para as estações em que há falta. Os autores apresentam uma matheurística com três etapas para resolver o problema. A primeira etapa inicia-se configurando cada estação como um grupo. Em seguida, para cada par de grupos é avaliado o custo de união dos mesmos, ou seja, calcula-se o número de bicicletas faltantes no grupo resultante, ou seja, calcula-se o número de bicicletas faltantes no grupo resultante. Para avaliar o custo, são utilizados dados que *a priori* estima o número de bicicletas que faltariam ou sobriariam em um conjunto de estações. São, então, escolhidas as uniões que resultem nas menores faltas, e o processo se repete até que o diâmetro máximo de cada grupo, passado como parâmetro, não seja ultrapassado. Depois que não é realizada mais nenhuma união de grupos, é utilizado um método de programação inteira para realizar as rotas entre os mesmos. No modelo, cada grupo é representado pelo vértice mais próximo de seu centroide. O percurso de cada veículo é limitado pelo horário noturno previsto para a reposição das bicicletas. Portanto, no final da segunda etapa cada veículo possui uma sequência de grupos que deve atender. Essa sequência define grupos vizinhos. Na terceira etapa, para cada rota é gerado um grafo de tal forma que, entre cada par de grupos vizinhos, todos os vértices de um grupo sejam adjacentes aos vértices do outro grupo, e todos os vértices de um grupo sejam adjacentes entre si. A geração desse grafo permite que a rota final de cada veículo transite entre vértices de grupos vizinhos. Por fim, é realizado o roteamento nesse grafo final utilizando um modelo de programação linear inteira.

A Figura 17 ilustra as três etapas do método de [Forma, Raviv e Tzur \(2015\)](#). A Figura 17a representa os grupos que foram inicialmente gerados cada grupo está representado em uma cor e

formato. Na Figura 17b, está ilustrada a sequência de grupos que cada veículo deve atender e o centroide de cada grupo está representado por um X. Na Figura 17c, é ilustrado o grafo obtido na terceira etapa. Pode-se observar que os vértices de um grupo estão conectados entre si e também aos vértices dos grupos vizinhos. Por fim, a Figura 17d apresenta uma possível solução para o problema.

Figura 17 – Etapas do método de [Forma, Raviv e Tzur \(2015\)](#)

Fonte: Elaborada pelo autor.

A Tabela 2 resume as principais características dos trabalhos que utilizam o agrupamento como estratégia de resolução de problemas de roteamento de veículos. Na primeira coluna, são referenciados os artigos. A aplicação e a motivação para o uso desta estratégia são descritos, respectivamente, nas colunas 2 e 3. Também é destacado se os autores consideram janelas de tempo (JT) e o número de depósitos (ND) nas colunas 4 e 5. Na sexta coluna, é destacado se o método proposto está baseado na estratégia *cluster first, route second* (CF-RS) ou se o método permite que um mesmo veículo atenda a mais de um agrupamento (M-Clu).

3.2 CluVRP

O roteamento de veículos para grupos (*clustered vehicle routing problem*, em inglês) é uma extensão do PRV capacitado em que os clientes são previamente separados em grupos (*clusters*).

Quadro 2 – Resumo de artigos que utilizam agrupamento como estratégia de resolução.

Autores	Aplicação	Motivação	JT	ND	Método
Desrosiers, Dumas e Soumis (1988)	Transporte de pessoas com necessidades especiais	Reduzir a dimensão do problema	Sim	1	Heurística (CF-RS)
Kim, Kim e Sahoo (2006)	Coleta de resíduos sólidos considerando horário de almoço dos motoristas	Melhorar a capacidade das rotas e balancear horários de trabalho	Não	1	Heurística (M-Clu)
Dondo e Cerdá (2007)	-	Reduzir a dimensão do problema	Sim	M	Heurística (M-Clu)
Schuijbroek, Hampshire e Hoeve (2013)	Reposição de bicicletas	Reduzir a dimensão do problema	Não	1	Math-heurística (CF-RS)
Forma, Raviv e Tzur (2015)	Reposição de bicicletas	Reduzir a dimensão do problema	Não	1	Heurística (M-Clu)

Segundo Sörensen *et al.* (2008), os clientes podem ser separados por dois motivos. O primeiro motivo é a separação das encomendas por regiões geográficas que pode simplificar o processo de empacotamento das entregas nos veículos, em especial, para grandes empresas. O segundo motivo é devido ao conhecimento local dos motoristas que seriam designados a determinadas regiões.

Sevaux e Sörensen (2008) têm como motivação a resolução de problemas de grandes empresas no qual o carregamento dos veículos é realizado de forma ordenada por regiões. De acordo com os autores, os clientes são geralmente separados por regiões postais. Dado um depósito central e clientes separados em grupos que possuem um centroide, o objetivo é minimizar o custo das rotas de forma a passar por todos os centroides e, portanto, por todas as regiões, sem violar a capacidade dos veículos. Em seguida, para obter uma solução final do problema são calculadas as rotas internas de cada grupo.

Barthélemy *et al.* (2010) foram os primeiros a definir formalmente o CluVRP. Seja $\mathcal{N} = \{1, \dots, n\}$ um conjunto de clientes que devem ser atendidos e $\mathcal{C} = \{C_1, \dots, C_m\}$ uma partição dos clientes em m grupos. O objetivo do problema é minimizar os custos para atender todos os clientes de forma que toda vez que um veículo entrar em uma partição, esse apenas irá sair após atender a todos os clientes da partição. As Figuras 18a e 18b ilustram o problema como proposto por Barthélemy *et al.* (2010). Na figura, os clientes de uma mesma região são definidos pela mesma cor e mesmo formato. A Figura 18b mostra uma possível solução do problema considerando dois veículos.

Para abordar o problema, os autores modificam os custo de viagem entre dois clientes que estão em grupos diferentes adicionando um valor M suficientemente grande para forçar a restrição de entrada e saída dos grupos. Depois desse pré-processamento, uma solução inicial é obtida pela heurística de Clarke e Wright (1964). Em seguida, são utilizadas uma busca local e *Simulated Annealing* para obter rotas de melhor qualidade.

Pop, Kara e Marc (2012) foram os primeiros a modelar o CluVRP. Os autores desenvolveram um modelo com base no *Generalized Vehicle Routing Problem* (GVRP). O GVRP é um problema de

roteamento no qual todos os clientes estão divididos em grupos e o objetivo é visitar apenas um cliente em cada grupo. Para o CluVRP, ao invés de atender apenas um cliente em cada grupo, é necessário atender todos os clientes em um determinado grupo antes de deixá-lo. Nesse artigo não é apresentado nenhum resultado computacional com essa variação.

Battarra, Erdogan e Vigo (2014) são os primeiros a apresentarem resultados com métodos exatos para o CluVRP. Os autores tratam o problema no contexto de coleta de resíduo sólido e o formulam como um problema de programação linear inteira que utiliza a estrutura de grupos para diminuir a dimensão do problema. Para isto, é realizado um pré-processamento que retorna os vértices de entrada e saída de cada grupo e o menor caminho hamiltoniano entre esses dois vértices. Esse pré-processamento é realizado para que todos os grupos sejam tratados como dois vértices do grafo durante a resolução do método exato. Os métodos de *Branch-and-Cut* e *Branch-and-Cut-and-Price* foram propostos para resolver o problema utilizando os grupos como vértices. O *Branch-and-Cut* foi o método que obteve melhores resultados.

A Figura 19 ilustra o processo de resolução de Battarra, Erdogan e Vigo (2014). A Figura 19a representa os clientes separados em regiões. Na Figura 19b está ilustrado o pré-processamento realizado o qual determina o menor caminho hamiltoniano em cada grupo e os vértices de entrada e saída. A rota em vermelho da Figura 19c representa a solução gerada pelo modelo que passa pelos vértices de entrada e saída de cada grupo. A Figura 19d ilustra o que seria a solução final utilizando os caminhos hamiltonianos obtidos pelo pré-processamento.

O artigo mais recente que aborda o CluVRP é de Pop *et al.* (2018). Nesse artigo, os autores também propõem a solução do problema em duas etapas. Diferentemente de Battarra, Erdogan e Vigo (2014), os autores resolvem primeiro o roteamento entre os grupos utilizando um algoritmo genético determinando assim a sequência de grupos a serem atendidos. Depois de obter as rotas entre os grupos, é definido um problema de caixeiro viajante com todas as cidades de cada grupo em uma única rota alterando os custos das viagens entre grupos diferentes suficientemente grande para que todos os

Figura 18 – Um exemplo com 25 clientes para o CluVRP.

Fonte: Adaptada de Solomon (1987b).

Figura 19 – Etapas do método de Battarra, Erdogan e Vigo (2014)

Fonte: Adaptada de Solomon (1987b).

clientes em cada grupo sejam atendidos juntos. Para a resolução do caixeiro viajante de cada rota é utilizado o *software* Concorde (APPLEGATE *et al.*, 2001). A maneira para assegurar uma única entrada e saída de cada grupo atribuindo um custo elevado para as viagens entre clientes de grupos distintos é realizado de maneira semelhante em (BARTHÉLEMY *et al.*, 2010).

A Tabela 3 resume os artigos encontrados durante a pesquisa tratam o CluVRP. Poucos artigos foram encontrados e apenas em (BATTARRA; ERDOGAN; VIGO, 2014) e (POP *et al.*, 2018) resultados computacionais são apresentados. As colunas da tabela trazem as mesmas informações apresentadas na Tabela 2.

3.3 Considerações finais

O problema estudado foi inspirado em um problema real em que os clientes estão divididos em estados (grupos). Por isso, foram pesquisados artigos em que os clientes estão particionados. Notou-se que na literatura existem estudos que dividem clientes de forma regional, entretanto, para nenhuma das aplicações há custos reais para as viagens que cruzam as fronteiras entre regiões (grupos). Ao invés

Quadro 3 – Resumo dos artigos que abordam o CluVRP.

Autores	Aplicação	Motivação	JT	ND	Método
Sevaux e Sörensen (2008)	Grandes empresas que separam clientes por regiões	Facilitar o empacotamento dos pedidos de uma rota	Não	1	Heurística
Barthélemy <i>et al.</i> (2010)	Grandes empresas que separam clientes por regiões	Clientes agrupados por bairros	Não	1	Heurística
Pop, Kara e Marc (2012)	-	Clientes agrupados por bairros	Não	1	Modelo
Battarra, Erdogan e Vigo (2014)	Coleta de resíduo sólido	Diminuir a dimensão do problema	Não	1	Math-heurística
Pop <i>et al.</i> (2018)	-	Clientes agrupados por bairros	Não	1	Heurística

disso, os veículos são forçados a entrar e sair de cada grupo apenas uma vez. Os PRVs que possuem essa característica são chamados de CluVRP. Além disso, foram apresentados artigos que utilizam o agrupamento de veículos para reduzir a dimensão do problema.

PROBLEMA ESTUDADO

Este trabalho é inspirado no problema de uma distribuidora de produtos que opera em território nacional. Os clientes são numerosos e situam-se em diferentes estados. O problema pode ser representado como o grafo ilustrado na Figura 20, em que cada ponto (nó) representa um cliente. Apesar do grafo ser completo, as arestas entre cada par de nós não são apresentadas para que a figura seja melhor visualizada. Os segmentos de reta que separam os nós do grafo representam a separação dos clientes em estados, ou seja, os clientes da mesma cor e formato estão situados em um mesmo estado.

Figura 20 – Exemplo de clientes separados em estados.

Fonte: Elaborada pelo autor.

A separação dos clientes em estados é ressaltada devido à fiscalização realizada em postos fiscais localizados nas fronteiras estaduais. Essas fiscalizações são realizadas para evitar sonegações de impostos sobre produtos como determina o [Protocolo-ICMS-10/03 \(2003\)](#).

A auditoria realizada nos postos fiscais é obrigatória em alguns estados ([PROTOCOLO-ICMS-10/03, 2003](#)) e, neste caso, o veículo é parado e deve ser realizada a checagem dos documentos e da

carga dos veículos que passam pelas fronteiras. A checagem pode levar algumas horas e, portanto, é desejável não cruzar a fronteira diversas vezes em uma mesma entrega. A Figura 21 ilustra duas formas de atender os clientes que se localizam próximos a uma fronteira. Na rota apresentada na Figura 21a, o veículo passa pela fronteira seis vezes. Isto significa que o veículo seria auditado em postos fiscais seis vezes em uma mesma viagem. Nota-se que a rota ilustrada na Figura 21b atende os mesmos clientes que a anterior, entretanto o veículo passa pela fronteira apenas duas vezes e, portanto, a rota seria realizada em menos tempo.

Figura 21 – Rotas em fronteiras.

Fonte: Elaborada pelo autor.

O problema estudado pode ser tratado como um PRV com uma característica adicional, uma vez que os caminhos entre clientes de estados distintos têm o custo mais dispendioso. Durante a pesquisa, foi avaliado qual seria o impacto de inserir custos adicionais reais nas viagens entre clientes de estados distintos, estratégia semelhante foi utilizada por [Barthélemy et al. \(2010\)](#) para o CluVRP. Os autores propõem acrescentar custos suficientemente grandes para que seja garantido que todos os clientes de um estado sejam atendidos antes do veículo visitar outro estado. Neste trabalho, no entanto, é proposto acrescentar custos que buscam refletir o tempo de parada na fronteira entre estados. Dessa forma, são acrescentados aos custos de viagens interestaduais um valor fixo. Além disso, uma modelagem que considera que o tempo de cruzamento de fronteira é proporcional a carga do veículo é apresentada.

4.1 Modelos matemáticos

Esta seção é dividida em duas partes. Na primeira, são definidas as características do modelo considerando os custos de viagens interestaduais fixos. Na seção seguinte, é apresentado um modelo considerando esses custos proporcionais a carga do veículo no momento que cruzar a fronteira.

4.1.1 PRVC com Custos de Fronteira Fixos (PRVCFF)

Esta primeira abordagem de solução trata o problema como um PRVC em que os clientes estão divididos em estados e existem custos adicionais de fronteira, que neste trabalho é chamado de PRVC com Custos de Fronteira Fixos (PRVCFF). O problema é definido por um grafo como descrito no Capítulo 2. A característica adicionada ao PRVC clássico é o tempo de viagem entre dois vértices i e j que é dado por T_{ij} , em que as arestas entre vértices que estão em estados distintos representam não apenas o tempo de viagem, mas também o tempo adicional das auditorias realizadas nos postos fiscais. A Figura 22 mostra esta diferença considerada no trabalho. A Figura 22a ilustra um exemplo com nove clientes separados em três estados. A Figura 22b ilustra as viagens considerando as auditorias fiscais, por isso, as viagens interestaduais (representadas em vermelho) possuem uma distância maior quando comparada com a Figura 22a. As viagens entre estados diferentes, representados pelas arestas em vermelho, têm seu custo elevado por um valor fixo. Novamente, nesta figura não estão representadas todas as arestas para que seja ela melhor visualizada.

Figura 22 – Exemplo de arestas com custos adicionais.

(a) Viagens sem considerar as auditorias fiscais.

(b) Viagens considerando as auditorias fiscais.

Fonte: Elaborada pelo autor.

Em seguida, é apresentada a primeira modelagem proposta. Para tanto, foi utilizada a formulação de dois índices por fluxo de veículos (IRNICH; TOTH; VIGO, 2014). As variáveis x_{ij} assumem valor 1 se um veículo realiza a viagem do cliente i para o cliente j e 0 caso contrário. As variáveis u_i são reais e representam a quantidade de carga no veículo após visitar o cliente i .

$$\min \sum_{i \in \mathcal{N} \cup \{0\}} \sum_{j \in \mathcal{N} \cup \{0\}} \bar{T}_{ij} x_{ij} \quad (4.1)$$

$$\text{s.a.} \quad \sum_{j \in \mathcal{N} \cup \{0\}} x_{ij} = 1, \quad \forall i \in \mathcal{N} \cup \{0\} \quad (4.2)$$

$$\sum_{i \in \mathcal{N} \cup \{0\}} x_{ij} = 1, \quad \forall j \in \mathcal{N} \cup \{0\} \quad (4.3)$$

$$\sum_{j \in \mathcal{N}} x_{0j} \leq K, \quad (4.4)$$

$$u_i - d_j - M_1(x_{ij} - 1) \geq u_j, \quad \forall i, j \in \mathcal{N} \quad (4.5)$$

$$0 \leq u_i \leq Q, \quad \forall i \in \mathcal{N} \cup \{0\} \quad (4.6)$$

$$x_{ij} \in \{0, 1\}, \quad \forall i, j \in \mathcal{N} \cup \{0\} \quad (4.7)$$

A função objetivo (4.1) visa minimizar o tempo de viagem de todos os veículos. O parâmetro $\bar{T}_{ij}$ é o tempo de viagem do cliente i para o cliente j acrescido do custo de fronteira para clientes de estados diferentes. As restrições (4.2) e (4.3) garantem o fluxo dos veículos entre os vértices do grafo, isto é, todos os vértices são conectados a dois outros vértices em uma rota. A restrição (4.4) restringe o número máximo de veículos a serem utilizados. As restrições (4.5) eliminam subciclos no grafo e garantem o sequenciamento de carga dos veículos. M_1 é um parâmetro que corresponde a um valor suficientemente grande. Esse tipo de formulação para eliminação de subciclos é conhecida como MTZ e foi proposto por [Miller, Tucker e Zemlin \(1960\)](#). As restrições (4.6) garantem a não violação da capacidade dos veículos (Q). Por fim, as restrições (4.7) definem as variáveis x_{ij} como binárias.

4.1.2 PRVC com Custos de Fronteira Proporcionais (PRVCFP)

Este modelo tem como objetivo considerar o custo de passagem nas fronteiras proporcional ao volume de carga do veículo no momento em que ele cruza a fronteira, em que neste trabalho é chamado de PRVC com Custos de Fronteira Proporcionais (PRVCFP). Essa característica deve ser considerada quando os veículos são inspecionados no momento em que é parado em um posto fiscal. Neste modelo, são usadas as variáveis de decisão anteriores e as variáveis f_{ij} que contabilizam o tempo que o veículo fica parado na fronteira quando este muda de estado.

Além dos parâmetros anteriormente definidos, neste modelo, C é um parâmetro utilizado para converter a unidade de carga em cada veículo em tempo de passagem na fronteira. Por fim, M_2 é o parâmetro utilizado como *big M* para as restrições que acrescentam custos a viagens interestaduais.

$$\min \quad \sum_{i \in \mathcal{N}} \sum_{j \in \mathcal{N}} T_{ij} x_{ij} + \sum_{i \in \mathcal{N}} \sum_{j \in \mathcal{N}} C f_{ij} \quad (4.8)$$

$$\text{s.a.} \quad (4.2) - (4.6)$$

$$M_2(x_{ij} - 1) + u_i \leq f_{ij} \quad \forall i \in E_1, \forall j \in E_2, \\ \forall E_1, E_2 \in \mathcal{E}, \text{ se } E_1 \neq E_2 \quad (4.9)$$

$$f_{ij} \geq 0, \quad \forall i, j \in \mathcal{N} \quad (4.10)$$

$$0 \leq u_i \leq Q, \quad \forall i \in \mathcal{N} \cup \{0\} \quad (4.11)$$

$$x_{ij} \in \{0, 1\}, \quad \forall i, j \in \mathcal{N} \quad (4.12)$$

A função objetivo (4.8) visa minimizar o tempo de viagem de todos os veículos. Na primeira parcela é minimizado o tempo de viagem, enquanto a segunda parcela da função objetivo contabiliza o custo de atravessar a fronteira que é proporcional à carga do veículo. As restrições (4.2) - (4.6) são as mesmas do modelo anterior. As restrições (4.9) são desconsideradas se a variável $x_{ij} = 0$. Se $x_{ij} = 1$, ou seja, se cliente está trocando de estado, então é contabilizada a carga do veículo ao chegar no vértice j pela variável u_i . Esse valor é atribuído à variável f_{ij} . As restrições (4.10) e (4.12) definem o domínio das variáveis.

4.1.3 Desigualdade triangular

Nesta seção, é mostrado que a desigualdade triangular se mantém para o problema estudado. Sejam i, j e k quaisquer clientes de um PRVC, T_{ij} , T_{ik} e T_{kj} os tempos para viajar entre os clientes i, j e k . Inicialmente, é sabido que:

$$T_{ij} \geq T_{ik} + T_{kj}$$

Agora suponha que i e j são clientes de estados diferentes. Logo, é adicionado um custo fixo CF a essa viagem. Entretanto, o cliente k pode estar no mesmo estado que i ou j . Em ambos os casos:

$$T_{ij} + CF \geq T_{ik} + T_{kj} + CF \rightarrow \bar{T}_{ij} \geq \bar{T}_{ik} + \bar{T}_{kj}$$

em que $\bar{T}_{ij} = T_{ij} + CF$ para qualquer i e j que estejam em estados diferentes e, portanto, a desigualdade triangular é válida para o PRVCFF.

Agora, será demonstrada a desigualdade triangular para o PRVCFP. Sejam i, j e k quaisquer clientes de um PRVC, T_{ij} , T_{ik} e T_{kj} os custos para viajar entre os clientes i, j e k , f_{ij} , f_{ik} e f_{kj} as cargas do veículo após atender os clientes i e j e antes de trocarem de estado, respectivamente e C um parâmetro para converter unidade de carga em unidade de tempo. Inicialmente, é sabido que:

$$T_{ij} \geq T_{ik} + T_{kj}$$

Agora, é suposto que i e j estão em estados diferentes. Dessa maneira, $f_{ij} \geq 0$ pela restrição 4.9. Sem perda de generalidade, pode ser suposto que k pertence ao mesmo estado que i . Dessa forma, é adicionada nessa viagem f_{kj} conforme a restrição 4.9. Assim:

$$T_{ij} + Cf_{ij} \geq T_{ik} + Tkj + Cf_{kj}$$

e, como $u_i \geq u_k \rightarrow Cf_{ij} \geq Cf_{kj}$, a desigualdade é válida para o PRVCFP.

4.2 Exemplo ilustrativo

A Figura 23 ilustra as soluções ótimas obtidas resolvendo uma instância adaptada de Solomon (1987a) utilizando o PRVC e o PRVCFP. Como pode-se observar, as soluções ótimas utilizam apenas um veículo. A Figura 23a ilustra a solução do PRVC clássico. Nessa solução, como não há custos de fronteira, os clientes que estão nos estados cinza (polígono) são visitados em dois momentos, ou seja, o veículo visita alguns deles e depois de atender a outros clientes (em estados diferentes) retorna para atender os clientes restantes do estado. O mesmo acontece para os clientes nos estados verde escuro (estrela de seis pontas), laranja (estrela de dez pontas) e azul claro (círculo maior). A Figura 23b ilustra uma solução ótima obtida utilizando o PRVCFP em que as auditorias fiscais são consideradas como custo fixo em cada viagem realizada entre estados diferentes. Vale destacar que os clientes de um mesmo estado são atendidos em conjunto, ou seja, o veículo uma vez que entre em um estado, atende a todos os seus clientes. Logo, como esperado, os custos de fronteira podem alterar as rotas ótimas.

Figura 23 – Comparação entre soluções obtidas pelo PRVC e PRVCFP.

Fonte: Elaborada pelo autor.

4.3 Experimentos Computacionais

Nesta seção, são apresentados os experimentos computacionais realizados utilizando um computador com processador i7-4790, 16 GB de memória RAM no sistema operacional Ubuntu 16.04 LTS. O *software* utilizado foi o GUROBI 7.0 com a interface em python 2.7 com o limite máximo de tempo de 60 minutos.

Os resultados computacionais, na Seção 4.4, são divididos em duas partes. Na primeira, são apresentados os resultados que foram obtidos para os modelos PRVC e PRVCFF. O modelo PRVC possui as mesmas restrições que o modelo PRVCFF, entretanto, a função objetivo não possui os custos acrescidos, ou seja, os parâmetros T_{ij} são apenas as distâncias euclidianas. Além disso, para todas as instâncias as distâncias entre os clientes foram aproximadas para o inteiro mais próximo, pois, de acordo com Uchoa *et al.* (2017), essa medida tem sido adotada na literatura. O custo acrescido foi de 45. Esse valor foi escolhido devido a testes computacionais preliminares que mostraram que a inserção desse custo alterariam as soluções obtidas. Por fim, o parâmetro M_1 utilizado foi a capacidade do veículo (Q). Esses resultados foram realizados para buscar avaliar a importância de ser considerada os custos de fronteira em um problema de PRVC.

Na segunda parte, são comparados os resultados dos modelos PRVCFF e o PRVCFP. O valor utilizado para C na função objetivo (4.8) do modelo de PRVCFP é $\frac{45}{Q}$, pois é utilizado o custo fixo igual a 45 e Q é a capacidade do veículo, logo, C é proporcional a capacidade utilizada do veículo. Além disso, utiliza-se $M_2 = Q$ para as restrições (4.9).

4.3.1 Geração de Instâncias

As instâncias utilizadas foram os conjuntos A, B e P de Augerat *et al.* (1995). De acordo com Uchoa *et al.* (2017), essas instâncias fazem parte do conjunto de instâncias mais utilizado para avaliar métodos exatos para o PRVC. Para o nome de cada instância usa-se a notação “ T - nN - kK .vrp” em que T , N , K variam conforme a instância. O número de clientes e veículos mínimos a serem utilizados são N e K , respectivamente. T representa o tipo da instância em que altera a disposição dos clientes e suas demandas. Nas instâncias do tipo A, os clientes são distribuídos em um quadrado de tamanho 100 e as demandas são obtidas utilizando uma distribuição uniforme entre 1 e 30. Os clientes das instâncias do tipo B são gerados em um quadrado de tamanho 100 de forma a serem criados NC grupos de clientes em que $K \leq NC - 1$. Suas demandas são tomadas de uma distribuição uniforme entre 1 e 30 e, além disso, $N/10$ clientes foram escolhidos para terem sua demanda multiplicada por 3. Por fim, os clientes das instâncias do tipo P são modificações de algumas instâncias do conjunto A, B e E em que a capacidade dos veículos é alterada e os valores de K quando necessários.

As instâncias de Augerat *et al.* (1995) foram adaptadas para considerar estados. As fronteiras dos estados foram definidas paralelas aos eixos x e y da seguinte forma: foram tomadas as coordenadas dos clientes com as maiores e menores coordenadas x e y (x_{max}, x_{min} e y_{max}, y_{min} , respectivamente). Em seguida, são calculadas as diferenças $x_{dif} = x_{max} - x_{min}$ e $y_{dif} = y_{max} - y_{min}$. Assim, as fronteiras

são obtidas tendo como base:

$$d_x = x_{min} + Fx_{dif} \quad (4.13)$$

para as fronteiras do eixo x e

$$d_y = y_{min} + Fy_{dif} \quad (4.14)$$

para as fronteira do eixo y , em que F é a porcentagem de distância para criar a fronteira.

Dessa forma, foram definidas duas fronteiras no eixo x e as outras duas no eixo y . Duas dessas arestas não percorrem o espaço todo e terminam na primeira aresta que ela intercepta no eixo y . Foram criadas quatro arestas que dividem os clientes em no máximo 6 estados. A Figura 24 ilustra as fronteiras geradas. Na Figura 24a existem duas fronteiras paralelas ao eixo y (são a linha contínua verde). Ambas as linhas foram geradas utilizando a equação (4.14) em que P é igual a 15% para a linha mais a esquerda e P é igual a 40% na linha mais a direita. Diferentemente, da linha mais a direita, a linha com $P = 15\%$ não cruza todo o espaço. Ao invés disso, para ao encontrar a primeira fronteira paralela ao eixo x . Por outro lado, as fronteiras paralelas ao eixo x estão em tracejado laranja. Ambas foram criadas utilizando a equação (4.13). A que está mais abaixo possui o P igual a 30% e o P igual a 80% pertence a fronteira mais acima. Nessa fronteira mais ao norte da figura, não é percorrido o espaço todo. Ao invés disso, para ao encontrar a fronteira paralela ao eixo y . A Figura 24b ilustra como ficaria a instância final com todas as fronteiras.

Figura 24 – Geração de estados nas instâncias utilizadas.

Fonte: Elaborada pelo autor.

4.4 Resultados Computacionais

Os experimentos computacionais foram realizados para avaliar o impacto dos custos adicionais nas fronteiras em um PRVC. Dessa forma, para cada instância, foi avaliado o tempo de execução, a solução obtida e o GAP. O GAP neste trabalho é definido como:

$$GAP = \frac{Melhor_Sol - Lim_Inf}{Melhor_Sol}$$

em que *Melhor_Sol* é o custo da melhor solução encontrada e *Lim_Inf* é o custo da melhor expectativa de solução. Além disso, foi calculada a diferença relativa entre os custos de cada solução para cada instância *i* da seguinte forma:

$$Dif_Rel = \frac{Sol_PRVC_i - Sol_PRVCFF_i}{Sol_PRVCFF_i} \quad (4.15)$$

onde *Sol_PRVCFF_i* representa o valor da solução da instância *i* obtida pelo modelo PRVCFF. Por outro lado, *Sol_PRVC_i* representa o valor da solução da instância *i* obtida pelo PRVC adicionando-se ao valor da solução os custos de fronteira. Esse valor foi calculado para quantificar o impacto gerado pelos custos acrescidos nas fronteiras.

A Figura 25 mostra a diferença relativa das soluções obtidas para as instâncias do conjunto A, B e P em sequência para o PRVCFF e PRVC com custos de fronteira considerados após obter as soluções. Como esperado, na maior parte dos casos, as soluções de melhor qualidade são obtidas quando considerados os custos de fronteira *a priori*. Apenas para as instâncias A-n48-k7 e B-n57-k7 a solução obtida pelo PRVC é de melhor qualidade. Esse resultado foi obtido porque nenhum dos modelos conseguiu atingir soluções ótimas (GAP igual a zero). Nas instâncias B-n34-k5, P-n19-k2 e P-n20-k2 as soluções dos dois modelos foram as mesmas, por isso, a diferença relativa foi igual a zero.

Figura 25 – Diferença entre os custos das soluções para cada instância.

Para todas as instâncias o tempo máximo de 1 hora foi atingido, com exceção de duas instâncias para o PRVCFF e seis para o PRVC. Essas instâncias são as menores do conjunto P. As soluções que foram obtidas antes do tempo máximo para ambos os modelos foram P-n16-k8 e P-n22-k8, ou seja,

a solução obtida para esses casos foi a ótima. Para as instâncias P-n19-k2, P-n20-k2, P-n21-k2 e P-n22-k2, as soluções encontradas pelo modelo PRVC foram ótimas e, por isso, em um tempo inferior a 3600 segundos.

Na Tabela 1 são apresentadas as médias dos tempos de execução, os *GAPs* e diferença relativa como definido em (4.15) para todas as instâncias de cada conjunto na primeira, segunda e terceira linhas, respectivamente. As instâncias do tipo P possuem uma média menor para o tempo de execução devido às seis e às quatro instâncias em que o ótimo é atingido. As instâncias do tipo B são as que sofreram o menor aumento do *GAP* quando resolvidas considerando os custos de fronteira. Esse fator se deve ao fato que os clientes por já estarem agrupados demandam naturalmente viagens com menor número de cruzamento de fronteiras.

Tabela 1 – Resumo dos resultados obtidos.

	PRVC-A	PRVCFF-A	PRVC-B	PRVCFF-B	PRVC-P	PRVCFF-P
Tempo (s)	3600,01	3600,01	3603,29	3600,01	2926,62	3304,59
GAP (%)	43,70	49,53	66,43	69,17	22,90	35,82
Diferença relativa (%)	9,08		6,95		13,59	

Em seguida, são apresentados os resultados computacionais para o modelo proposto na Seção 4.1.2 (PRVCFP). Neste modelo considera-se que os custos de cruzar a fronteira são dependentes da carga do veículo ao passar pela mesma. O custo foi considerado dessa forma para uma representação mais fidedigna do tempo gasto pelos veículos nos postos fiscais.

As instâncias utilizadas foram as mesmas para o PRVCFF e as fronteira foram geradas da mesma forma. Para apresentar estes resultados também foi utilizada uma diferença relativa para comparar as soluções de ambos os modelos, PRVCFF e PRVCFP. Essa comparação é feita para avaliar o impacto causado pela alteração de considerar os custos fixos e variáveis. A diferença relativa foi calculada da seguinte forma:

$$Dif_Rel = \frac{Sol_PRVCFP_i - Sol_PRVCFF_i}{Sol_PRVCFF_i} \quad (4.16)$$

onde Sol_PRVCFF_i representa o custo da solução da instância i obtida pelo modelo PRVCFF. Por outro lado, Sol_PRVCFP_i representa o custo da solução da instância i obtida pelo PRVCFP. A Figura 26 ilustra a diferença relativa para cada instância do grupo A, B e P de Augerat *et al.* (1995). É possível notar que em todos os casos o modelo que considera a carga proporcional possui um custo menor.

Para seis instâncias o GAP das soluções do modelo PRVCFP foram maiores que o PRVCFF e apenas em duas o GAP das soluções foram as mesmas (que foram iguais a zero). Em todas as outras, os GAPs obtidos pelo modelo PRVCFP foram menores. Além disso, para as sete primeiras instâncias do grupo P, o GAP obtido foi menor que 10^{-4} enquanto que para o modelo PRVCFF duas dessas instâncias (P-n16-k8 e P-n22-k8) obtiveram um GAP igual a zero.

Por fim, é possível concluir que o modelo PRVCFP obteve melhores resultados tanto na qualidade da solução como nos custos das soluções obtidas. Entretanto, das 74 instâncias avaliadas em

Figura 26 – Diferença entre os modelos PRVCFP e PRVCFF.

todas o tempo de execução foi o tempo limite de 60 minutos, com exceção das sete instâncias que obtiveram um GAP menor que 10^{-4} .

Nas Tabelas 2, 3, 4 e 5 são apresentados os resultados obtidos de forma detalhada. Para nenhuma das instâncias dos Grupos A e B a solução ótima foi obtida no tempo limite (3.600s). Nas Tabelas 2 e 3, a primeira coluna corresponde à instância resolvida. Nas colunas seguintes para os modelos PRVC, PRVCFF e PRVCFP são apresentados: o custo da melhor solução obtida (Solução), o melhor limitante inferior (LI) e o GAP (em porcentagem). Na coluna da melhor solução obtida do PRVC são apresentados dois resultados. Primeiro é apresentado o custo da solução obtida pelo modelo PRVC e, em parênteses, é apresentado o custo da mesma solução com os custos de fronteira acrescidos. Ainda na coluna do PRVC, o GAP apresentado é obtido pelo modelo PRVC sem os custos de fronteira. Como para algumas as instâncias do Grupo P o ótimo foi atingido, nas Tabelas 4 e 5, além das informações anteriores, o tempo para resolver cada instância também é apresentado. Na coluna das soluções do modelo PRVCFP, as solução estão em decimais porque, apesar das distâncias entre clientes estarem arredondadas para o inteiro mais próximo, os custos para considerar as cargas são adicionados, tornando assim, o custo final da solução em um decimal.

4.5 Considerações Finais

Nesta seção foi apresentado um novo variante do PRVC em que os clientes são divididos em estados e são considerados custos reais de fronteira no momento em que os veículos cruzam as fronteiras. Esse problema foi abordado de duas maneiras. Na primeira, são considerados custos fixos quando o veículo troca de estado e, na segunda, os custos na fronteira são considerados em

relação a carga do veículo. Além disso, foram apresentadas e comparadas duas modelagens para cada abordagem. Como esperado, quando são considerados os custos proporcionais a carga na fronteira, as soluções obtidas são de melhores qualidades em todas as instâncias.

Tabela 2 – Resultados obtidos para as instâncias do Grupo A.

Nome	PRVC			PRVCFE			PRVCFP		
	Solução	LI	GAP	Solução	LI	GAP	Solução	LI	GAP
A-n32-k5.vrp	806 (1256)	481	40%	1222	780	36%	946,35	577,19	39%
A-n33-k5.vrp	665 (1205)	475	29%	1126	765	32%	791,75	593,69	25%
A-n33-k6.vrp	743 (1373)	544	27%	1291	867	33%	848,25	644,93	24%
A-n34-k5.vrp	785 (1370)	494	37%	1247	835	33%	948,65	611,04	36%
A-n36-k5.vrp	810 (1440)	529	35%	1342	730	46%	906,95	594,05	34%
A-n37-k5.vrp	671 (1121)	550	18%	1040	805	23%	732,75	624,51	15%
A-n37-k6.vrp	995 (1760)	594	40%	1621	819	49%	1143,70	640,94	44%
A-n38-k5.vrp	735 (1275)	511	30%	1226	784	36%	908,10	608,54	33%
A-n39-k5.vrp	822 (1452)	588	28%	1329	888	33%	951,20	674,08	29%
A-n39-k6.vrp	848 (1568)	579	32%	1403	865	38%	950,00	676,10	29%
A-n44-k7.vrp	951 (1761)	622	35%	1570	932	41%	1203,55	714,33	41%
A-n45-k6.vrp	999 (1809)	612	39%	1518	807	47%	1049,45	683,26	35%
A-n45-k7.vrp	1178 (1943)	546	54%	1793	888	50%	1340,75	680,08	49%
A-n46-k7.vrp	932 (1652)	581	38%	1501	813	46%	1122,65	698,19	38%
A-n48-k7.vrp	1111 (1876)	584	47%	1894	725	62%	1214,40	629,09	48%
A-n53-k7.vrp	1072 (1882)	582	46%	1702	869	49%	1141,10	704,32	38%
A-n54-k7.vrp	1206 (2061)	585	51%	2043	723	65%	1363,60	622,14	54%
A-n55-k9.vrp	1170 (2070)	598	49%	1894	857	55%	1304,15	686,78	47%
A-n60-k9.vrp	1451 (2531)	594	59%	2459	809	67%	1675,15	676,63	60%
A-n61-k9.vrp	1104 (2004)	560	49%	1825	803	56%	1231,65	658,57	47%
A-n62-k8.vrp	1386 (2196)	616	56%	2119	790	63%	1489,70	697,69	53%
A-n63-k10.vrp	1425 (2280)	635	55%	2118	807	62%	1524,30	689,95	55%
A-n63-k9.vrp	1774 (2764)	642	64%	2403	818	66%	1926,45	714,69	63%
A-n64-k9.vrp	1529 (2384)	610	60%	2035	781	62%	1663,70	669,89	60%
A-n65-k9.vrp	1301 (2291)	594	54%	2079	751	64%	1383,65	669,08	52%
A-n69-k9.vrp	1271 (2306)	692	46%	1997	824	59%	1443,55	783,50	46%
A-n80-k10.vrp	1970 (3005)	741	62%	2522	851	66%	2216,85	799,82	64%

Tabela 3 – Resultados obtidos para as instâncias do Grupo B.

Nome	PRVC			PRVCFF			PRVCFP		
	Solução	LI	GAP	Solução	LI	GAP	Solução	LI	GAP
B-n31-k5.vrp	675 (1350)	227	66%	1213	401	67%	734,15	283,23	61%
B-n34-k5.vrp	789 (1329)	283	64%	1329	454	66%	834,55	316,03	62%
B-n35-k5.vrp	981 (1476)	239	76%	1368	332	76%	1002,50	264,38	74%
B-n38-k6.vrp	819 (1404)	284	65%	1299	547	58%	922,95	354,31	62%
B-n39-k5.vrp	552 (1137)	245	56%	1115	391	65%	617,25	342,05	45%
B-n41-k6.vrp	841 (1561)	327	61%	1543	524	66%	910,30	375,35	59%
B-n43-k6.vrp	749 (1244)	358	52%	1243	601	52%	852,80	420,92	51%
B-n44-k7.vrp	940 (1840)	230	76%	1692	334	80%	986,35	261,49	73%
B-n45-k5.vrp	791 (1601)	411	48%	1387	743	46%	1006,80	434,68	57%
B-n45-k6.vrp	722 (1262)	281	61%	1109	496	55%	767,95	344,54	55%
B-n50-k7.vrp	741 (1371)	258	65%	1283	404	69%	786,70	284,07	64%
B-n50-k8.vrp	1353 (2478)	543	60%	2209	675	69%	1498,00	572,82	62%
B-n51-k7.vrp	1023 (1473)	338	67%	1472	425	71%	1086,95	391,70	64%
B-n52-k7.vrp	754 (1429)	247	67%	1390	433	69%	808,00	282,88	65%
B-n56-k7.vrp	765 (1395)	225	71%	1207	237	80%	784,75	230,88	71%
B-n57-k7.vrp	1148 (1778)	301	74%	1788	391	78%	1213,75	313,45	74%
B-n57-k9.vrp	1626 (2616)	341	79%	2563	452	82%	1776,30	360,08	80%
B-n63-k10.vrp	1578 (2613)	420	73%	2582	579	78%	1810,45	485,46	73%
B-n64-k9.vrp	895 (1480)	335	63%	1371	461	66%	961,70	376,91	61%
B-n66-k9.vrp	1361 (1856)	436	68%	1732	589	66%	1458,45	498,74	66%
B-n67-k10.vrp	1115 (2060)	374	66%	1779	459	74%	1240,35	391,08	68%
B-n68-k9.vrp	1323 (2223)	258	80%	2061	352	83%	1573,40	289,82	82%
B-n78-k10.vrp	1352 (2477)	414	69%	2200	567	74%	1596,20	488,87	69%

Tabela 4 – Resultados obtidos para as instâncias do Grupo P.

Nome	PRVC				PRVCFF				PRVCFP			
	Tempo	Solução	LI	GAP	Tempo	Solução	LI	GAP	Tempo	Solução	LI	GAP
P-n16-k8.vrp	2,07	450 (1215)	450	0%	1,86	1158	1158	0%	0,43	643,14	643,14	0%
P-n19-k2.vrp	726,55	212 (482)	212	0%	3600,01	482	403	16%	416,02	273,03	273,01	0%
P-n20-k2.vrp	2248,42	216 (486)	216	0%	3600,00	486	395	19%	426,56	275,34	275,34	0%
P-n21-k2.vrp	826,36	211 (481)	211	0%	3649,57	481	389	19%	1507,82	272,81	272,79	0%
P-n22-k2.vrp	1468,62	216 (486)	216	0%	3609,41	486	335	31%	2259,87	273,31	273,29	0%
P-n22-k8.vrp	166,86	590 (1220)	590	0%	49,30	1169	1169	0%	10,58	740,50	740,50	0%
P-n23-k8.vrp	3600,00	529 (1474)	506	4%	3600,00	1416	1198	15%	148,97	718,75	718,75	0%
P-n40-k5.vrp	3600,01	474 (969)	382	19%	3600,00	862	671	22%	3600,01	577,18	464,98	19%
P-n45-k5.vrp	3600,00	530 (1205)	414	22%	3600,00	916	658	28%	3600,01	608,40	494,17	19%
P-n50-k10.vrp	3600,02	725 (1535)	429	41%	3600,01	1504	717	52%	3600,01	890,70	523,61	41%
P-n50-k7.vrp	3600,00	575 (1295)	414	28%	3600,01	1144	631	45%	3600,02	701,40	483,39	31%
P-n50-k8.vrp	3600,00	650 (1550)	421	35%	3600,00	1210	649	46%	3600,01	735,12	509,46	31%

Tabela 5 – Resultados obtidos para as instâncias do Grupo P.

Nome	PRVC				PRVCFF				PRVCFP			
	Tempo	Solução	LI	GAP	Tempo	Solução	LI	GAP	Tempo	Solução	LI	GAP
P-n51-k10.vrp	3600,01	769 (1759)	451	41%	3600,00	1451	703	52%	3600,01	930,62	551,02	41%
P-n55-k10.vrp	3600,02	732 (1677)	429	41%	3600,00	1467	655	55%	3600,01	850,30	518,30	39%
P-n55-k15.vrp	3600,01	977 (2147)	533	45%	3600,01	1971	812	59%	3600,03	1152,93	645,94	44%
P-n55-k7.vrp	3600,01	578 (1253)	423	27%	3600,00	1113	637	43%	3600,01	720,29	500,57	31%
P-n55-k8.vrp	3600,00	605 (1280)	426	30%	3600,02	1168	646	45%	3600,01	793,09	505,54	36%
P-n60-k10.vrp	3600,01	787 (1597)	460	42%	3600,00	1476	698	53%	3600,02	945,00	558,32	41%
P-n60-k15.vrp	3600,01	992 (2117)	511	48%	3600,01	1900	775	59%	3600,01	1184,94	601,62	49%
P-n65-k10.vrp	3600,01	836 (2006)	511	39%	3600,01	1585	748	53%	3600,01	1020,50	594,69	42%
P-n70-k10.vrp	3600,00	874 (1909)	530	39%	3600,01	1608	763	53%	3600,02	1088,33	614,87	44%
P-n76-k4.vrp	3600,01	632 (1487)	538	15%	3600,02	1084	775	29%	3600,02	780,16	595,70	24%
P-n76-k5.vrp	3600,01	668 (1388)	539	19%	3600,01	1125	765	30%	3600,02	899,96	599,88	33%
P-n101-k4.vrp	3600,01	721 (1711)	628	13%	3600,01	1282	842	34%	3600,03	846,64	670,36	21%

MÉTODO HEURÍSTICO

Neste trabalho, uma metaheurística VND (*Variable Neighborhood Descent*) foi desenvolvida para resolver o Problema de Roteamento de Veículos Capacitado com Custo de Fronteiras Fixas (PRVCFF) e para o Problema de Roteamento de Veículos com Custo de Fronteira Proporcional (PRVCFP). A VND é uma variante da VNS (*Variable Neighborhood Search*) em que a troca de vizinhanças é realizada de forma determinística (HANSEN; MLADENOVIC, 2014). A VND foi escolhida como objeto de estudo devido aos bons resultados que tem apresentado para aplicações do PRVC (p.e. em (KYTÖJOKI *et al.*, 2007) e (CHEN; HUANG; DONG, 2010)) e a sua simplicidade de elaboração. Além disso, a VND possui grande flexibilidade para que seja aplicada a diferentes variantes do PRVC e tem permitido obter soluções em baixo tempo computacional. De acordo com Cordeau *et al.* (2002), simplicidade, qualidade, flexibilidade e tempo de resolução são fatores fundamentais para a obtenção de métodos heurísticos eficientes.

Neste capítulo, a princípio é descrita, em linhas gerais, a metaheurística VND. Em seguida, o método de resolução desenvolvido é detalhado. Posteriormente, é apresentada a diferença para o método considerando custos de fronteira fixos e proporcionais. Por último, os resultados dos experimentos computacionais para o PRVCFF e PRVCFP são reportados.

5.1 Variable Neighborhood Descent

A *Variable Neighborhood Descent* consiste em buscar uma solução ótima alternando vizinhanças de forma sistemática para sair de regiões onde são encontrados mínimos locais (GENDREAU; POTVIN, 2010). Como citado, a troca de vizinhanças de uma VND é realizada de forma determinística. O Algoritmo 3 resume a metaheurística VND.

O número de vizinhanças utilizadas (n), um vetor de vizinhanças (vizinhanças) e uma solução factível (Sol) são parâmetros iniciais do método. O custo da melhor solução encontrada (custo_Melhor_Sol) é definido na linha 2. Em seguida, a busca por uma solução de melhor qualidade é iniciada (linhas 4 a 13). Antes, no entanto, uma variável auxiliar para identificar a vizinhança utilizada

Algoritmo 3 – VND

```

1: procedimento VND( $n$ , vizinhanças, Sol)
2:   Custo_Melhor_Sol  $\leftarrow$  Custo(Sol)
3:    $k \leftarrow 1$ 
4:   enquanto  $k < n$  faça
5:     Sol  $\leftarrow$  Movimento(Sol, vizinhança[k])
6:     se Custo(Sol) < Custo_Melhor_Sol então
7:       Melhor_Sol  $\leftarrow$  Sol
8:       Custo_Melhor_Sol  $\leftarrow$  Custo(Melhor_Sol)
9:        $k \leftarrow 1$ 
10:    senão
11:       $k \leftarrow k + 1$ 
12:    fim se
13:  fim enquanto
14:  retorna Melhor_Sol
15: fim procedimento

```

é inicializada (linha 3). Na linha 5, é encontrado um vizinho da solução atual (Sol). Essa busca pode ser realizada, basicamente, de duas formas: melhor vizinho ou primeiro vizinho melhor. Na busca pelo melhor vizinho, chamada de máxima descida para problemas de minimização, é procurada a melhor solução vizinha da atual em relação a k -ésima vizinhança do vetor de vizinhanças. Na busca pelo primeiro vizinho melhor, chamada de descida para problemas de minimização, assim que uma solução vizinha de melhor qualidade em relação a k -ésima vizinhança é encontrada a busca é terminada. Entre as linhas 6 e 12, a melhor solução obtida é avaliada e a troca de vizinhança é realizada. Caso a solução encontrada na linha 5 seja de melhor qualidade, a melhor solução é atualizada (linhas 7 e 8) e a vizinhança é reinicializada (linha 9). Caso contrário, a vizinhança é alterada para a seguinte do vetor de vizinhanças (linha 11).

5.2 VND com múltiplos inícios

Neste trabalho, foi implementada uma VND com múltiplos inícios (VND-MI) para resolver o problema de roteamento de veículos estudado. De acordo com Brønmo *et al.* (2007), não existem muitos trabalhos em roteamento de veículos que utilizam múltiplos inícios. Trabalhos que utilizam essa estratégia fazem uso de reinicializações de soluções iniciais para varrer um espaço maior durante a busca. O algoritmo desenvolvido consiste em gerar uma solução inicial, aplicar a VND partindo desta solução, ao seu término, perturbar a solução inicial e aplicar a VND novamente, este ciclo se repete por um número determinado de iterações. A melhor solução obtida é retornada. O Algoritmo 4 resume o método desenvolvido.

Algoritmo 4 – VND com múltiplos inícios - VND_MI

```

1: procedimento VND-MI( $n$ , vizinhanças)
2:    $Sol \leftarrow Sol\_inicial(Dados)$ 
3:    $n\_rotas \leftarrow Num\_Rotas(Sol)$ 
4:    $Custo\_Melhor\_Sol \leftarrow Custo(Sol)$ 
5:    $i \leftarrow 1$ 
6:   enquanto  $i < n\_rotas$  faça
7:      $Sol \leftarrow VND(n, vizinhanças, Sol)$ 
8:     se  $custo(Sol) < Custo\_Melhor\_Sol$  então
9:        $Melhor\_Sol \leftarrow Sol$ 
10:       $Custo\_Melhor\_Sol \leftarrow Custo(Melhor\_Sol)$ 
11:    fim se
12:     $Sol \leftarrow Perturba\_Solução(Sol\_Inicial, i)$ 
13:     $i \leftarrow i + 1$ 
14:  fim enquanto
15:  retorna  $Melhor\_Sol$ 
16: fim procedimento

```

Os dados de entrada do método são: o número de vizinhanças (n) e o vetor de vizinhanças a ser percorrido (vizinhanças). Na linha 2, é gerada uma solução inicial (Sol). Em seguida, nas linhas 3 e 4, são inicializados os valores do número de rotas da solução inicial (n_rotas) e seu custo ($Custo_Melhor_Sol$), respectivamente. Na linha 5, o número de iterações do método (i) é inicializado. Na linha 7, é aplicada a VND apresentada no Algoritmo 3. Entre as linhas 8 e 11, é avaliada a qualidade da solução encontrada e, se uma solução melhor é obtida, ocorre uma atualização da melhor solução ($Melhor_Sol$) e seu custo ($Custo_Melhor_Sol$). Por fim, na linha 12, é chamada a função $Perturba_Solução()$ para que uma nova solução inicial para a VND seja gerada. Ao final, o método irá retornar a melhor solução encontrada. Cada etapa do método é descrita a seguir.

5.2.1 Solução Inicial

Como solução inicial para a VND com múltiplos inícios foram avaliadas duas heurísticas construtivas: o método de economias (CLARKE; WRIGHT, 1964) e a Inserção do Mais Distante (JOHNSON; MCGEOCH, 2007). Ambos estão detalhados na Seção 2.1.1

5.2.2 VND - Movimento

A função $VND()$ é detalhada no Algoritmo 3. O movimento utilizado ($Movimento()$, linha 5 do Algoritmo 3) busca pelo melhor vizinho da k -ésima vizinhança, ou seja, a busca na vizinhança é realizada como máxima descida. O vetor de vizinhanças utilizado possui as seguintes vizinhanças: 2-opt inter-rotas, 1-1 *exchange*, 1-0 *exchange*, 2-0 *exchange*, Or-opt e por último 2-opt intra-rotas. O vetor de vizinhanças foi criado com essas vizinhanças nessa ordem e, portanto, toda busca realizada pela VND foi realizada nessa sequência. Essa sequência foi utilizada por apresentar os melhores resultados em relação às outras.

5.2.3 Perturba_Solução

A VND desenvolvida utiliza uma rotina de múltiplos inícios para diversificar a busca por soluções. A geração de novos inícios tem como base a perturbação da solução inicial obtida utilizando uma das duas heurísticas construtivas descritas anteriormente. Para gerar um novo início, uma rota da solução da inicial é destruída e, para cada cliente dessa rota, uma viagem com início e fim no depósito é criada. A Figura 27 ilustra esse procedimento. Na Figura 27a está ilustrada uma solução inicial gerada pela Inserção do Mais Distante. A Figura 27b mostra a etapa de destruição de uma rota. Nesse caso, a rota $(0, 3, 4, 2, 0)$ foi destruída. Por fim, a Figura 27c apresenta a reconstrução da solução adicionando uma rota para cada um dos clientes da rota destruída. Dessa forma, as rotas que foram construídas são $(0, 3, 0)$, $(0, 4, 0)$ e $(0, 2, 0)$. Essa nova solução com as rotas reconstruídas é a nova solução inicial da VND.

Figura 27 – Geração de múltiplas soluções.

(a) Solução gerada pela Inserção do Mais Distante. (b) Primeira etapa da geração de múltiplas soluções.

(c) Segunda etapa da geração de múltiplas soluções.

O Algoritmo 5 resume o pseudocódigo da rotina de perturbação da solução. Inicialmente, todas as rotas são ordenadas em relação ao custo de maneira decrescente (linha 2). Em seguida, a i -ésima rota é removida da solução inicial (linhas 3-4). Por fim, no *loop* nas linhas 5-7, cada cliente j da rota removida é adicionado como uma nova rota iniciando no depósito, viajando até o cliente j e terminando no depósito.

5.2.4 Aplicação da heurística

A heurística como descrita pelo algoritmo 5.2 foi aplicada para o PRVCFF e o PRVCFP. Foi necessária uma adaptação da heurística para que esta contemplasse os dois problemas estudados. A diferença consiste no cálculo dos custos das rotas. As gerações de fronteira foram realizadas da mesma maneira apresentada na Seção 4.3.1.

Para o cálculo dos custos do PRVCFF, o custo fixo de 45 foi acrescido em todas as viagens interestaduais de cada instância. Para o cálculo dos custos do PRVCFP nenhum custo foi acrescido a

Algoritmo 5 – Algoritmo da Perturba_Solução.

```

procedimento PERTUBA_SOLUÇÃO(Sol, i)
  Sol ← Ordena_Rotas(Sol)
  rota_removida ← Sol[i]
  Sol ← Remove(Sol, rota_removida)
  para cada cliente j em rota_removida faça
    nova_rota ← (0, j, 0)
    Sol ← Adiciona_Rota(Sol, nova_rota)
  fim para
  retorna Sol
fim procedimento

```

priori, uma vez que esse depende da carga do veículo no momento em que se cruza a fronteira, ou seja, é dado por:

$$C_{i,j} = \frac{45}{Q}(u_i) \quad \forall i \in E_1, \forall j \in E_2, \text{ se } E_1 \neq E_2 \quad (5.1)$$

em que $C_{i,j}$ representa o custo da viagem entre clientes localizados em estados diferentes. Q representa a carga total do veículo e u_i representa a carga do veículo no momento que troca de estado e E_1 e E_2 são estados distintos.

5.3 Resultados computacionais

Nesta seção, são apresentados os resultados computacionais obtidos para as heurísticas desenvolvidas durante a pesquisa para o PRVCFF e o PRVCFP. Primeiramente, foi avaliada a qualidade das soluções obtidas em relação a resultados obtidos na literatura e a importância da função Perturba_Solução(). Para tanto, o método também foi aplicado sem a rotina Perturba_Solução() que consiste na VND descrita no Algoritmo 3. Na terceira etapa, os melhores resultados são comparados com os resultados obtidos utilizando o modelo matemático PRVCFF e o modelo PRVCFP apresentado na Seção 4.1 e, por fim, os dois métodos são comparados entre si para avaliar o impacto de considerar os custos de fronteira proporcional a carga do veículo no momento de cruzar a fronteira.

5.3.1 Análise de Resultados - Etapa 1

Nesta seção, é avaliada a qualidade das soluções obtidas pelo método heurístico quando comparadas com as soluções de melhor qualidade reportadas na literatura. Na Tabela 6, são resumidas as diferenças relativas médias percentuais dos custos das soluções obtidas em relação aos melhores resultados da literatura para cada instância dos conjuntos A, B e P (colunas 2-4). Para cada instância a diferença percentual é dada por:

$$Diferenca_Rel_1 = 100 \frac{(Sol_Heur - Sol_Lit)}{Sol_Lit} \quad (5.2)$$

em que *Sol_Heur* representa a solução obtida pela heurística e *Sol_Lit* a melhor solução apresentada em (UCHOA *et al.*, 2017). Para realizar estes cálculos, apenas as soluções com a mesma quantidade de rotas das soluções de Augerat *et al.* (1995) foram utilizadas.

Na primeira e segunda linhas da Tabela 6, são apresentadas as diferenças médias das soluções obtidas pelas heurísticas construtivas Inserção do Mais Distante e Clarke e Wright (1964), respectivamente. Na terceira (FI-VND) e na quarta linhas (CW-VND), são apresentadas, respectivamente, as diferenças médias obtidas pela VND a partir da solução inicial obtida pela Inserção do Mais Distante e por Clarke e Wright (1964). Por fim, na quinta (FI-VND-MI) e na sexta linhas (CW-VND-MI), são apresentadas as diferenças médias para a heurística proposta com múltiplos inícios utilizando a Inserção do Mais Distante e Clarke e Wright (1964), respectivamente.

Na Tabela 6, é possível perceber que as soluções obtidas pela heurística construtiva de Clarke e Wright (1964) estão muito próximas da melhor solução conhecida, em média, a 5,18%. As instâncias de Augerat *et al.* (1995) foram propostas para o PRVC clássico e, de acordo com Cordeau *et al.* (2002), essa heurística construtiva apresenta bons resultados para problemas desse tipo. Apesar das soluções geradas pela Inserção do Mais Distante serem de pior qualidade, a VND desenvolvida melhora significativamente os resultados obtidos, por exemplo, para as instâncias do Grupo B, os valores médios de 125,53% são reduzidos para 9,76%, mostrando a eficiência da VND. Mesmo para as soluções de Clarke e Wright (1964), a VND obtém melhoria, nesse caso, em média de 4,28% nos três grupos. As linhas de VND-MI, como esperado, resultam nas melhores médias, melhorando de forma significativa os resultados obtidos. Os melhores resultados foram obtidos pela combinação da solução inicial de Clarke e Wright (1964) e a VND-MI (CW-VND-MI) para a maioria das instâncias.

Tabela 6 – Melhoria obtida com as heurísticas implementadas.

	A	B	P
Inserção do Mais Distante	114,34	125,53	83,13
Clarke e Wright (1964)	4,50	2,82	8,24
FI-VND	10,97	9,76	4,91
CW-VND	3,33	1,90	7,61
FI-VND-MI	3,47	2,44	1,16
CW-VND-MI	1,94	1,78	3,20

Os resultados obtidos pelas soluções das heurísticas CW-VND e CW-VND-MI comparados com os melhores resultados da literatura (UCHOA *et al.*, 2017) para as instâncias avaliadas são apresentados na Figura 28. Na Figura 28, no eixo *x* estão representadas todas as instâncias, em que as soluções obtidas pelas heurísticas possuem a mesma quantidade de veículos que a solução da literatura, e no eixo *y* sua diferença relativa. A diferença é calculada utilizando (5.2) em que *Sol_Heur* representa a solução obtida pela heurística (CW-VND ou CW-VND-MI) e *Sol_Lit* a melhor solução apresentada em (UCHOA *et al.*, 2017). Como ilustrado na figura, quando realizado o procedimento de múltiplos inícios, as soluções obtidas são melhores ou não há nenhuma alteração. Esse resultado era esperado uma vez que a rotina Perturba_Solução() abrange o espaço de busca da heurística em que múltiplos inícios não são permitidos. Pode-se observar que CW-VND-MI é uma heurística competitiva com os melhores resultados da literatura.

Figura 28 – Resultados obtidos para as heurísticas CW-VND e CW-VND-MI comparados com a literatura.

5.3.2 Análise de Resultados - Etapa 2

Nessa etapa, são apresentados os resultados considerando as instâncias em são considerados os custos de fronteiras fixos e proporcionais. As heurísticas avaliadas são as CW-VND-MI e CW-VND-MI-P. As heurísticas são comparadas com os resultados obtidos pelos modelos PRVCFF e PRVCFP apresentados na Seção 4.4. A comparação dos resultados está baseada na diferença relativa entre as soluções encontradas, definida da seguinte forma:

$$Diferenca_Rel_2 = \frac{(Sol_Heu - Sol_Mod)}{Sol_Mod} \quad (5.3)$$

em que Sol_Heu e Sol_Mod são, respectivamente, o custo da solução obtida pela heurística e o custo da solução obtida pelo *solver* resolvendo o modelo considerando os custos de fronteira.

A Figura 29 ilustra a diferença relativa para cada instância estudada entre a heurística CW-VND-MI e o modelo PRVCFF comparando assim a qualidade das soluções obtidas pela heurística e as soluções obtidas pelo *solver* considerando os custos de fronteira fixos. Em 55 instâncias, as heurísticas obtiveram soluções melhores que *solver*; em 5 instâncias a solução obtida foi a mesma; e, por fim, em 14 instâncias o *solver* obteve soluções melhores. Vale destacar, no entanto, que a diferença está abaixo de 0,05 para 69 instâncias e acima de 0,15 para apenas uma. Em relação ao tempo, apenas para a instância P-n101-k4 o tempo foi de 91,58 segundos, enquanto para todas as outras instâncias o tempo de execução da heurística foi menor que 25 segundos. Vale lembrar que o *solver* utilizou o tempo limite (3.600 s) para 69 instâncias, para as demais o tempo variou de 2,07 a 2248,42 segundos.

A Figura 30 ilustra a diferença relativa para cada instância estudada entre a heurística CW-VND-MI-P e o modelo PRVCFP comparando a qualidade das soluções obtidas pela heurística e pelo *solver* considerando os custos de fronteira proporcionais. Em apenas 2 instâncias, as heurísticas

Figura 29 – Comparação entre as soluções obtidas pela CW-VND-MI e pelo modelo PRVCFF.

obtiveram soluções melhores que o *solver*. Em relação ao tempo, apenas para a instância P-n101-k4 o tempo foi de 240,26 segundos, enquanto para todas as outras instâncias o tempo de execução da heurística foi menor que 51 segundos. O *solver* utilizou o tempo limite (3.600 s) para 67 instâncias, para as demais o tempo variou de 0,43 a 2259,87 segundos.

Figura 30 – Comparação entre as soluções obtidas pela CW-VND-MI-P e pelo modelo PRVCFP.

Por fim, a Figura 31 ilustra a diferença relativa entre as heurísticas CW-VND-MI e CW-VND-MI-P comparando a qualidade das soluções obtidas pelas heurísticas considerando os custos de

fronteira fixos e proporcionais. A diferença relativa foi calculada como apresentado em (5.3), alterando o *Sol_Mod* pelo valor da solução obtida pela heurística CW-VND-MI-P. Em todas as instâncias, as soluções obtidas pela heurística considerando os custos proporcionais foram de melhor qualidade. Isso ocorreu devido a flexibilidade oferecida pelo método utilizando considerando as cargas proporcionais. As viagens interestaduais considerando os custos proporcionais possuem seus custos de fronteira sendo reduzidos conforme o veículo viaja, diferente do método com os custos fixos de fronteira.

Figura 31 – Comparação entre as soluções obtidas pela CW-VND-MI e pela CW-VND-MI-P.

5.4 Considerações finais

Neste capítulo, foi analisada a qualidade dos métodos desenvolvidos. Duas heurísticas construtivas foram avaliadas: método das economias de [Clarke e Wright \(1964\)](#) e a Inserção do Mais Distante de ([JOHNSON; MCGEOCH, 2007](#)). Além disso, uma VND e uma função que gera múltiplos inícios foram desenvolvidas. A heurística que obteve o melhor resultado para as instâncias de [Augerat et al. \(1995\)](#) (conjuntos A, B e P) foi a VND com múltiplos inícios utilizando a construtiva de [Clarke e Wright \(1964\)](#). A VND desenvolvida apresentou bons resultados para ambas heurísticas construtivas, entretanto, utilizando a rotina de múltiplos inícios resultados melhores foram obtidos.

Essa heurística foi comparada com os *solvers* utilizando o modelo PRVCFF e PRVCFP. Para cada modelo, a maneira de calcular os custos na heurística proposta foi diferente, considerando os custos de fronteira fixos e proporcionais, respectivamente. Os resultados detalhados estão disponíveis no Apêndice 1.

CONCLUSÕES

O Problema de Roteamento de Veículos tem sido muito estudado nas últimas décadas. Neste trabalho, é proposta uma variante que não foi encontrada na literatura. Devido ao protocolo fiscal [Protocolo-ICMS-10/03 \(2003\)](#), os veículos que trocam de estado com alguma carga específica devem parar em um posto fiscal na fronteira e ter seus documentos e sua carga auditada. Durante a pesquisa, não foram encontrados trabalhos que consideram esses tipos de custos durante a resolução do problema.

Por isso, neste trabalho, é proposto uma variante em que essas características são consideradas de duas maneiras. Na primeira, custos fixos são acrescidos nas viagens em que ocorre mudança de estado e, na segunda, os custos acrescidos são considerados proporcionais à carga do veículo no momento em que ele cruza a fronteira. Para ambos os casos foram estudados modelos matemáticos. Instâncias baseadas na literatura foram resolvidas utilizando um *solver* comercial com tempo limite de 3600 segundos. Apesar de obtidas soluções factíveis para todas as instâncias, o tempo computacional utilizado foi máximo para quase todos os casos. Por isso, uma heurística foi desenvolvida.

A metaheurística desenvolvida foi a *Variable Neighborhood Search* (VNS), mais especificamente, a variante *Variable Neighborhood Descent* (VND). Essa foi a heurística escolhida devido aos bons resultados apresentados na literatura para o Problema de Roteamento de Veículos Capacitado (PRVC). A VND foi inserida em uma rotina de múltiplos inícios e como solução inicial foram utilizadas a heurística de economias de [Clarke e Wright \(1964\)](#) e a Inserção do Mais Distante de [\(JOHNSON; MCGEOCH, 2007\)](#). A heurística foi aplicada aos problemas com custos interestaduais fixos e proporcionais. Para as instâncias em que os custos de fronteira são fixos, a heurística se mostrou rápida e eficiente, rodando em média 7,78 segundos atingindo um GAP médio de 1,57% quando comparada as soluções obtidas resolvendo o problema utilizando um *solver* comercial. Considerando custos proporcionais, a heurística é igualmente rápida (tempo médio de 13,45 segundos), mas a qualidade dos resultados obtidos são um pouco inferiores (GAP médio de 16,99%).

Como trabalhos futuros é sugerido buscar uma adaptação da heurística para obter resultados de melhor qualidade, em especial, considerando custos proporcionais. Além disso, para se aproximar ainda mais da realidade, considerar que ao cruzar a fronteira de um estado, existe o custo fixo

representando a parada do veículo no local de fiscalização e um custo proporcional ao volume de carga do veículo nesse momento que representa a inspeção propriamente dita. O processod de inspeção não acontece necessariamente, logo seria interessante considerar também a probabilidade dos veículos serem realmente inspecionados.

REFERÊNCIAS

- AMOUS, M.; TOUMI, S.; JARBOUI, B.; EDDALY, M. A variable neighborhood search algorithm for the capacitated vehicle routing problem. **Electronic Notes in Discrete Mathematics**, Elsevier, v. 58, p. 231–238, 2017. Citado nas páginas 41 e 42.
- APPLEGATE, D.; BIXBY, R.; CHVATAL, V.; COOK, W. **Concorde TSP Solver**. 2001. Disponível em: <<http://www.math.uwaterloo.ca/tsp/concorde/index.html>>. Citado na página 51.
- AUGERAT, P.; BELENGUER, J.; BENAVENT, E.; CORBERÁN, A.; NADDEF, D.; RINALDI, G. Computational results with a branch and cut code for the capacitated vehicle routing problem. **Research report RR949–M, ARTEMIS-IMAG, France**, 1995. Citado nas páginas 59, 62, 74 e 77.
- BARTHÉLEMY, T.; ROSSI, A.; SEVAUX, M.; SÖRENSEN, K. **Metaheuristic approach for the clustered VRP**. 2010. Citado nas páginas 49, 51, 52 e 54.
- BATTARRA, M.; ERDOGAN, G.; VIGO, D. Exact algorithms for the clustered vehicle routing problem. **Operations Research**, v. 62, n. 1, p. 58–71, 2014. ISSN 0030364X 15265463. Citado nas páginas 13, 29, 50, 51 e 52.
- BRAEKERS, K.; RAMAEKERS, K.; NIEUWENHUYSE, I. V. The vehicle routing problem: State of the art classification and review. **Computers & Industrial Engineering**, Elsevier, v. 99, p. 300–313, 2016. Citado na página 30.
- BRÄYSSY, O. A reactive variable neighborhood search for the vehicle-routing problem with time windows. **INFORMS Journal on Computing**, INFORMS, v. 15, n. 4, p. 347–368, 2003. Citado na página 37.
- BRØNMO, G.; CHRISTIANSEN, M.; FAGERHOLT, K.; NYGREEN, B. A multi-start local search heuristic for ship scheduling—a computational study. **Computers & Operations Research**, Elsevier, v. 34, n. 3, p. 900–917, 2007. Citado na página 70.
- CHEN, P.; HUANG, H. kuan; DONG, X. Y. Iterated variable neighborhood descent algorithm for the capacitated vehicle routing problem. **Expert Systems with Applications**, Elsevier Ltd, v. 37, n. 2, p. 1620–1627, 2010. ISSN 09574174. Disponível em: <<http://dx.doi.org/10.1016/j.eswa.2009.06.047>>. Citado nas páginas 40, 42 e 69.
- CLARKE, G.; WRIGHT, J. W. Scheduling of vehicles from a central depot to a number of delivery points. **Operations research**, INFORMS, v. 12, n. 4, p. 568–581, 1964. Citado nas páginas 13, 26, 32, 33, 40, 42, 49, 71, 74, 77, 79 e 85.
- CONFEDERAÇÃO NACIONAL DO TRANSPORTE. 2016. Disponível em: <<http://www.cnt.org.br/Imprensa/noticia/custo-logistico-consome-12-do-pib-do-brasil>>. Acesso em: 28 fev 2017. Citado na página 25.
- CORDEAU, J.-F.; GENDREAU, M.; LAPORTE, G.; POTVIN, J.-Y.; SEMET, F. A guide to vehicle routing heuristics. **Journal of the Operational Research society**, JSTOR, p. 512–522, 2002. Citado nas páginas 30, 31, 32, 69 e 74.

- DANTZIG, G. B.; RAMSER, J. H. The truck dispatching problem. **Management science**, Informs, v. 6, n. 1, p. 80–91, 1959. Citado nas páginas 25 e 29.
- DESROSIERS, J.; DUMAS, Y.; SOUMIS, F. A dynamic programming solution of the large-scale single-vehicle dial-a-ride problem with time windows. **American Journal of Mathematical and Management Sciences**, Taylor & Francis, v. 6, n. 3-4, p. 301–325, 1986. Citado na página 43.
- _____. The multiple vehicle dial-a-ride-problem. **Computer-Aided Transit Scheduling. Lecture Notes in Economics and Mathematical Systems**, v. 308, p. 15–27, 1988. Citado nas páginas 13, 43, 44, 45 e 49.
- DONDO, R.; CERDÁ, J. A cluster-based optimization approach for the multi-depot heterogeneous fleet vehicle routing problem with time windows. **European Journal of Operational Research**, v. 176, n. 3, p. 1478–1507, 2007. ISSN 03772217. Citado nas páginas 13, 45, 46, 47 e 49.
- EKSIOGLU, B.; VURAL, A. V.; REISMAN, A. The vehicle routing problem: A taxonomic review. **Computers & Industrial Engineering**, Elsevier, v. 57, n. 4, p. 1472–1483, 2009. Citado na página 25.
- FORMA, I. A.; RAVIV, T.; TZUR, M. A 3-step math heuristic for the static repositioning problem in bike-sharing systems. **Transportation Research Part B: Methodological**, v. 71, n. June, p. 230–247, 2015. ISSN 01912615. Citado nas páginas 13, 47, 48 e 49.
- GENDREAU, M.; POTVIN, J.-Y. **Handbook of metaheuristics**. [S.l.]: Springer, 2010. v. 2. Citado na página 69.
- GOLDEN, B. L.; WASIL, E. A. Or practice—computerized vehicle routing in the soft drink industry. **Operations research**, INFORMS, v. 35, n. 1, p. 6–17, 1987. Citado na página 25.
- HANSEN, P.; MLADENOVIĆ, N. Variable neighborhood search: Principles and applications. **European journal of operational research**, Elsevier, v. 130, n. 3, p. 449–467, 2001. Citado na página 38.
- _____. Variable neighborhood search. In: **Search methodologies**. [S.l.: s.n.], 2014. p. 313–337. Citado nas páginas 38 e 69.
- IRNICH, S.; TOTH, P.; VIGO, D. The family of vehicle routing problems. **Vehicle Routing: Problems, Methods, and Applications**, SIAM, v. 18, p. 1–36, 2014. Citado nas páginas 32 e 55.
- JOHNSON, D. S.; MCGEOCH, L. A. Experimental analysis of heuristics for the stsp. In: **The traveling salesman problem and its variations**. [S.l.]: Springer, 2007. p. 369–443. Citado nas páginas 26, 32, 33, 71, 77 e 79.
- KIM, B.-I.; KIM, S.; SAHOO, S. Waste collection vehicle routing problem with time windows. **Computers & Operations Research**, v. 33, n. 12, p. 3624–3642, 2006. ISSN 03050548. Citado nas páginas 13, 44, 45, 46, 47 e 49.
- KYTÖJOKI, J.; NUORTIO, T.; BRÄYSS, O.; GENDREAU, M. An efficient variable neighborhood search heuristic for very large scale vehicle routing problems. **Computers and Operations Research**, v. 34, n. 9, p. 2743–2757, 2007. ISSN 03050548. Citado nas páginas 40, 42 e 69.
- LAPORTE, G. Fifty years of vehicle routing. **Transportation Science**, v. 43, n. 4, p. 408–416, 2009. ISSN 0041-1655. Citado na página 30.

- LAWLER, E. L.; LENSTRA, J. K.; KAN, A. R.; SHMOYS, D. B. *et al.* **The traveling salesman problem: a guided tour of combinatorial optimization**. [S.l.]: Wiley New York, 1985. v. 3. Citado na página 34.
- LENSTRA, J. K.; KAN, A. Complexity of vehicle routing and scheduling problems. **Networks**, Wiley Online Library, v. 11, n. 2, p. 221–227, 1981. Citado na página 31.
- MILLER, C. E.; TUCKER, A. W.; ZEMLIN, R. A. Integer programming formulation of traveling salesman problems. **Journal of the ACM (JACM)**, ACM, v. 7, n. 4, p. 326–329, 1960. Citado na página 56.
- MLADENOVIC, N.; HANSEN, P. Variable neighborhood search. **Computers & operations research**, Elsevier, v. 24, n. 11, p. 1097–1100, 1997. Citado nas páginas 38 e 39.
- OR, I. Traveling salesman type combinatorial problems and their relation to the logistics of regional blood banking. 1977. Citado na página 37.
- OSMAN, I. H. Metastrategy simulated annealing and tabu search algorithms for the vehicle routing problem. **Annals of operations research**, Springer, v. 41, n. 4, p. 421–451, 1993. Citado na página 35.
- PARRAGH, S. N.; DOERNER, K. F.; HARTL, R. F. A survey on pickup and delivery problems. **Journal fur Betriebswirtschaft**, v. 58, n. 1, p. 21–51, 2008. ISSN 03449327. Citado na página 30.
- POP, P. C.; FUKSZ, L.; MARC, A. H.; SABO, C. A novel two-level optimization approach for clustered vehicle routing problem. **Computers & Industrial Engineering**, Elsevier, v. 115, n. February 2017, p. 304–318, 2018. ISSN 0360-8352. Disponível em: <<https://doi.org/10.1016/j.cie.2017.11.018>>. Citado nas páginas 50, 51 e 52.
- POP, P. C.; KARA, I.; MARC, A. H. New mathematical models of the generalized vehicle routing problem and extensions. **Applied Mathematical Modelling**, v. 36, n. 1, p. 97–107, 2012. ISSN 0307904X. Citado nas páginas 49 e 52.
- POTVIN, J.-Y.; KERVAHUT, T.; GARCIA, B.-L.; ROUSSEAU, J.-M. The vehicle routing problem with time windows part i: tabu search. **INFORMS Journal on Computing**, INFORMS, v. 8, n. 2, p. 158–164, 1996. Citado na página 35.
- PRINS, C. A grasp× evolutionary local search hybrid for the vehicle routing problem. **Bio-inspired algorithms for the vehicle routing problem**, Springer, p. 35–53, 2009. Citado nas páginas 40 e 42.
- PROTOCOLO-ICMS-10/03. 2003. Disponível em: <<http://app1.sefaz.mt.gov.br/Sistema/legislacao/legislacaotribut.nsf/07fa81bed2760c6b84256710004d3940/550f7efb88f30e1304256d0b004a41c5?OpenDocument>>. Acesso em: 21 fev 2017. Citado nas páginas 26, 53 e 79.
- SCHUIJBROEK, J.; HAMPSHIRE, R.; HOEVE, W.-J. van. Inventory rebalancing and vehicle routing in bike sharing systems. Cluvrp. 2013. Citado nas páginas 46 e 49.
- SEVAUX, M.; SÖRENSEN, K. Hamiltonian paths in large clustered routing problems. **Proceedings of the EU/MEeting 2008 workshop on Metaheuristics for Logistics and Vehicle Routing, EU/ME'08**, p. 4:1–4:7, 2008. Disponível em: <[./Publications/inp-sevaux-08.pdf](#)>. Citado nas páginas 49 e 52.
- SOLOMON, M. M. Algorithms for the vehicle routing and scheduling problems with time window constraints. **Operations research**, Informs, v. 35, n. 2, p. 254–265, 1987. Citado nas páginas 29 e 58.

- _____. Algorithms for the vehicle routing and scheduling problems with time window constraints. **Operations research**, Informs, v. 35, n. 2, p. 254–265, 1987. Citado nas páginas 30, 45, 50 e 51.
- SÖRENSEN, K.; BERGH, J. Van den; CATTRYSSE, D.; SEVAUX, M. A multiobjective distribution problem for parcel delivery at tnt. In: **Invited talk at the international workshop on vehicle routing in practice, VIP**. [S.l.: s.n.], 2008. v. 8. Citado na página 49.
- SUBRAMANIAN, A.; DRUMMOND, L. M. d. A.; BENTES, C.; OCHI, L. S.; FARIAS, R. A parallel heuristic for the vehicle routing problem with simultaneous pickup and delivery. **Computers & Operations Research**, Elsevier, v. 37, n. 11, p. 1899–1911, 2010. Citado nas páginas 41 e 42.
- SUBRAMANIAN, A.; UCHOA, E.; OCHI, L. S. A hybrid algorithm for a class of vehicle routing problems. **Computers and Operations Research**, Elsevier, v. 40, n. 10, p. 2519–2531, 2013. ISSN 03050548. Disponível em: <<http://dx.doi.org/10.1016/j.cor.2013.01.013>>. Citado na página 41.
- THACH, C.; JIAN, N.; MINMEI, S.; LI, H.; MILLER, S. W.; ZHANG, L. Approximating the spanning star forest problem and its applications to genomic sequence alignment. Citeseer, 2008. Citado na página 47.
- TOTH, P.; VIGO, D. **Vehicle routing: problems, methods, and applications**. [S.l.]: SIAM, 2014. Citado nas páginas 25, 29 e 30.
- UCHOA, E.; PECIN, D.; PESSOA, A.; POGGI, M.; VIDAL, T.; SUBRAMANIAN, A. New benchmark instances for the capacitated vehicle routing problem. **European Journal of Operational Research**, Elsevier, v. 257, n. 3, p. 845–858, 2017. Citado nas páginas 41, 59 e 74.
- VOUDOURIS, C.; TSANG, E. Guided local search. **Handbook of metaheuristics**, Springer, p. 321–361, 2010. Citado na página 40.

RESULTADOS VND

Nas Tabelas 7, 8 e 9 estão as informações obtidas pela heurística e pelo modelo para cada grupo de instância A, B e P, respectivamente. Na primeira coluna, estão os nomes de cada instância dos grupos A, B e P. A segunda, terceira e quarta colunas apresentam os resultados obtidos pela VND implementada (VND), VND com múltiplos inícios (VND-MI) e a VND com múltiplos inícios e com os custos fixos de fronteira (VND-MI-F). Nas colunas seguintes, estão os tempos e os resultados da heurística utilizando VND com múltiplos inícios e com custos de fronteira proporcionais (VND-MI-P). Em todos os casos a heurística construtiva utilizada foi a [Clarke e Wright \(1964\)](#). Na última coluna estão os custos obtidos pelo modelo PRVCCP.

Tabela 7 – Resultados detalhados obtidos pela heurística desenvolvida e o modelo PRVCFF para as instâncias do grupo A.

Nome	VND		VND-MI		VND-MI-F		VND-MI-P		PRVCFF
	Tempo	Solução	Tempo	Solução	Tempo	Solução	Tempo	Solução	Solução
A-n32-k5.vrp	0,26	829	0,67	829	0,94	1222	0,69	1031,89	946,35
A-n33-k5.vrp	0,47	698	0,90	680	1,28	1128	0,32	936,200	791,75
A-n33-k6.vrp	0,24	774	1,03	774	1,37	1314	0,67	1037,84	848,25
A-n34-k5.vrp	0,20	792	0,62	790	1,48	1239	0,65	1012,35	948,65
A-n36-k5.vrp	0,24	805	1,44	805	1,94	1333	0,37	1082,39	906,95
A-n37-k5.vrp	0,68	691	1,61	669	1,70	1038	0,46	934,70	732,75
A-n37-k6.vrp	0,97	975	1,38	975	1,71	1585	0,62	1302,55	1143,70
A-n38-k5.vrp	1,04	749	1,69	749	1,78	1265	0,81	1004,90	908,10
A-n39-k5.vrp	0,43	893	1,26	823	2,21	1329	0,69	1210,64	951,20
A-n39-k6.vrp	0,91	855	1,84	843	3,08	1402	0,74	1189,24	950,00
A-n44-k7.vrp	1,07	966	2,69	959	3,88	1548	0,92	1280,64	1203,55
A-n45-k6.vrp	1,19	983	2,28	978	3,76	1516	1,76	1239,85	1049,45
A-n45-k7.vrp	1,03	1197	2,87	1172	3,13	1788	0,71	1592,24	1340,75
A-n46-k7.vrp	2,07	922	2,87	920	3,83	1486	0,87	1242,59	1122,65
A-n48-k7.vrp	1,57	1097	3,75	1097	3,57	1831	1,84	1423,90	1214,40
A-n53-k7.vrp	1,91	1058	4,26	1033	4,38	1686	1,99	1373,05	1141,10
A-n54-k7.vrp	4,97	1181	6,53	1172	5,78	1880	1,65	1517,00	1363,60
A-n55-k9.vrp	1,94	1109	5,83	1088	6,02	1909	2,01	1499,59	1304,15
A-n60-k9.vrp	3,45	1404	9,32	1365	8,81	2375	1,74	1871,09	1675,15
A-n61-k9.vrp	2,97	1088	7,94	1072	13,72	1798	2,56	1470,80	1231,65
A-n62-k8.vrp	2,27	1339	6,76	1333	10,30	2000	1,92	1673,00	1489,70
A-n63-k10.vrp	3,67	1341	9,08	1325	14,93	1967	2,47	1651,20	1524,30
A-n63-k9.vrp	12,57	1642	12,71	1640	17,15	2366	2,79	2075,44	1926,45
A-n64-k9.vrp	2,91	1472	9,12	1453	13,27	1951	2,73	1764,29	1663,70
A-n65-k9.vrp	3,88	1256	8,19	1233	14,56	2039	2,71	1657,85	1383,65
A-n69-k9.vrp	5,80	1181	12,97	1181	15,89	1903	2,67	1633,19	1443,55
A-n80-k10.vrp	5,90	1829	22,92	1821	24,98	2401	3,24	2191,30	2216,85

Tabela 8 – Resultados detalhados obtidos pela heurística desenvolvida e o modelo PRVCFF para as instâncias do grupo B.

Nome	VND		VND-MI		VND-MI-F		VND-MI-P		PRVCFF
	Tempo	Solução	Tempo	Solução	Tempo	Solução	Tempo	Solução	Solução
B-n31-k5.vrp	0,37	672	0,65	672	1,05	1216	0,33	933,40	734,15
B-n34-k5.vrp	0,39	788	1,00	788	1,04	1328	0,29	1040,14	834,55
B-n35-k5.vrp	0,54	971	0,85	968	1,58	1373	0,38	1136,85	1002,50
B-n38-k6.vrp	1,15	817	1,45	815	2,01	1285	1,28	1042,90	922,95
B-n39-k5.vrp	0,43	563	1,20	562	2,20	1106	0,49	853,450	617,25
B-n41-k6.vrp	0,96	892	2,33	843	3,27	1529	1,43	1211,55	910,30
B-n43-k6.vrp	0,68	773	2,38	755	2,75	1240	0,84	1049,85	852,80
B-n44-k7.vrp	1,12	933	2,30	917	2,56	1613	0,99	1273,20	986,35
B-n45-k5.vrp	0,36	751	1,44	751	3,26	1342	1,25	1027,80	1006,80
B-n45-k6.vrp	0,51	715	2,14	715	2,54	1064	0,86	868,64	767,95
B-n50-k7.vrp	0,84	743	2,63	743	3,50	1262	1,15	971,60	786,70
B-n50-k8.vrp	1,91	1352	4,33	1334	4,84	2168	2,08	1775,59	1498,00
B-n51-k7.vrp	2,05	1019	4,39	1018	3,71	1468	0,78	1242,64	1086,95
B-n52-k7.vrp	3,15	756	5,20	756	7,39	1346	2,66	1031,45	808,00
B-n56-k7.vrp	3,74	723	6,46	722	7,20	1204	2,94	964,59	784,75
B-n57-k7.vrp	5,71	1147	6,65	1145	7,20	1775	1,74	1435,15	1213,75
B-n57-k9.vrp	3,47	1637	7,67	1623	11,59	2522	1,86	2066,19	1776,30
B-n63-k10.vrp	3,51	1579	9,49	1567	12,62	2497	3,10	2032,40	1810,45
B-n64-k9.vrp	4,08	914	9,21	914	11,40	1362	2,50	1126,69	961,70
B-n66-k9.vrp	10,08	1400	13,20	1362	18,12	1670	2,29	1517,60	1458,45
B-n67-k10.vrp	5,29	1085	12,33	1083	15,40	1868	3,46	1456,74	1240,35
B-n68-k9.vrp	4,12	1306	11,20	1298	12,96	2023	3,30	1602,34	1573,40
B-n78-k10.vrp	5,11	1254	16,76	1253	22,32	2139	6,22	1669,14	1596,20

Tabela 9 – Resultados detalhados obtidos pela heurística desenvolvida e o modelo PRVCFF para as instâncias do grupo P.

Nome	VND		VND-MI		VND-MI-F		VND-MI-P		PRVCFF
	Tempo	Solução	Tempo	Solução	Tempo	Solução	Tempo	Solução	Solução
P-n16-k8.vrp	0,09	472	0,09	461	0,12	1237	0,04	935.00	643,14
P-n19-k2.vrp	0,04	234	0,07	234	0,15	501	0,08	427.09	273,03
P-n20-k2.vrp	0,02	234	0,03	234	0,16	499	0,11	432.09	275,34
P-n21-k2.vrp	0,03	236	0,06	211	0,19	481	0,10	388.78	272,81
P-n22-k2.vrp	0,03	240	0,07	216	0,16	486	0,09	401.21	273,31
P-n22-k8.vrp	0,11	590	0,16	590	0,22	1169	0,12	948.50	740,50
P-n23-k8.vrp	0,12	537	0,19	537	0,31	1483	0,20	1066.12	718,75
P-n40-k5.vrp	1,14	486	1,85	486	2,73	850	0,96	670.28	577,18
P-n45-k5.vrp	0,61	528	1,70	528	3,16	913	2,40	717.39	608,40
P-n50-k10.vrp	0,89	726	2,90	716	6,58	1471	1,09	1122.14	890,70
P-n50-k7.vrp	0,90	584	3,59	575	5,34	1129	1,16	824.79	701,40
P-n50-k8.vrp	0,92	642	3,16	640	6,26	1214	1,71	978.37	735,12
P-n51-k10.vrp	2,02	768	3,98	763	6,34	1414	1,72	1113.31	930,62
P-n55-k10.vrp	1,05	734	5,36	711	7,50	1412	1,62	1073.30	850,30
P-n55-k15.vrp	1,37	986	6,08	965	9,03	1972	0,98	1561.92	1152,93
P-n55-k7.vrp	3,00	611	5,97	596	6,65	1131	1,63	884.52	720,29
P-n55-k8.vrp	2,50	607	5,39	605	6,06	1159	1,76	933.50	793,09
P-n60-k10.vrp	3,01	789	7,84	782	10,12	1431	1,16	1157.37	945,00
P-n60-k15.vrp	6,37	1007	9,65	989	10,53	1891	2,59	1547.62	1184,94
P-n65-k10.vrp	2,59	833	9,32	820	13,33	1504	1,92	1231.73	1020,50
P-n70-k10.vrp	2,69	878	10,99	870	14,20	1536	2,80	1253.66	1088,33
P-n76-k4.vrp	3,61	665	12,95	647	21,43	1025	13,2	841.57	780,16
P-n76-k5.vrp	6,32	660	17,95	660	21,54	1068	6,42	920.71	899,96
P-n101-k4.vrp	21,94	701	40,56	700	91,58	1107	48,35	901.03	846,64