



**UNIVERSIDADE FEDERAL DO CEARÁ**  
**CAMPUS RUSSAS**  
**CURSO DE GRADUAÇÃO EM ENGENHARIA DE SOFTWARE**

**MATHEUS DIÓGENES ANDRADE**

**UM ALGORITMO PARA O PROBLEMA DE ROTEIRIZAÇÃO DE VEÍCULOS COM  
FROTA HETEROGÊNEA**

**RUSSAS**

**2018**

MATHEUS DIÓGENES ANDRADE

UM ALGORITMO PARA O PROBLEMA DE ROTEIRIZAÇÃO DE VEÍCULOS COM  
FROTA HETEROGÊNEA

Trabalho de Conclusão de Curso apresentado ao Curso de Graduação em Engenharia de Software do Campus Russas da Universidade Federal do Ceará, como requisito parcial à obtenção do grau de bacharel em Engenharia de Software.

Orientador: Prof. Dr. Dmontier Pinheiro Aragão Junior

Co-Orientador: Prof. Ms. Ramom Santana Rebouças

RUSSAS

2018

Dados Internacionais de Catalogação na Publicação  
Universidade Federal do Ceará  
Biblioteca Universitária

Gerada automaticamente pelo módulo Catalog, mediante os dados fornecidos pelo(a) autor(a)

---

A568a Andrade, Matheus Diógenes.

UM ALGORITMO PARA O PROBLEMA DE ROTEIRIZAÇÃO DE VEÍCULOS COM  
FROTA HETEROGÊNEA / Matheus Diógenes Andrade. – 2018.  
52 f. : il. color.

Trabalho de Conclusão de Curso (graduação) – Universidade Federal do Ceará, Campus  
de Russas, Curso de Engenharia de Software, Russas, 2018.

Orientação: Prof. Dr. Dmontier Pinheiro Aragão Junior.

Coorientação: Prof. Me. Ramon Santana Rebouças.

1. Problema de Roteirização de Veículos. 2. Roteirização. 3. Frota Heterogênea. 4.  
Algoritmo VND. I. Título.

CDD 005.1

---

MATHEUS DIÓGENES ANDRADE

UM ALGORITMO PARA O PROBLEMA DE ROTEIRIZAÇÃO DE VEÍCULOS COM  
FROTA HETEROGÊNEA

Trabalho de Conclusão de Curso apresentado ao  
Curso de Graduação em Engenharia de Software  
do Campus Russas da Universidade Federal do  
Ceará, como requisito parcial à obtenção do  
grau de bacharel em Engenharia de Software.

Aprovada em:

BANCA EXAMINADORA

---

Prof. Dr. Dmontier Pinheiro Aragão Junior (Orientador)  
Universidade Federal do Ceará (UFC)

---

Prof. Ms. Ramom Santana Rebouças (Co-Orientador)  
Universidade Federal do Ceará (UFC)

---

Prof. Ms. Tatiane Fernandes Figueiredo  
Universidade Federal do Ceará (UFC)

A Deus. Aos meus pais e a toda minha família. Aos professores Dmontier Pinheiro Aragão Junior e Ramom Santana Rebouças. A Universidade Federal do Ceará (UFC). Ao curso de Engenharia de Software, e aos colegas que convivi nesses anos.

## **AGRADECIMENTOS**

Agradeço a Deus, aos meus pais (dona Maria Diógenes e o senhor Francisco Gomes), por terem me dado o dom da vida e um berço de educação, amor e carinho, a minha família, ao professor orientador Dmontier Pinheiro Aragão Junior, ao professor coorientador Ramom Santana Rebouças, à Universidade Federal do Ceará campus Russas, ao curso de Engenharia de Software, e aos colegas que convivi nesses anos.

"O tempo que passa faz o trabalho de separar as nuvens de poeira de nossos pensamentos. Não significa necessariamente que sabemos mais, mas sim que as coisas ficam ordinariamente mais claras."(DIAS, 2014)

## RESUMO

No processo de distribuição é necessário fazer a entrega de bens e serviços para clientes dispersos geograficamente, nesse processo encontra-se o *Vehicle Routing Problem* (VRP). O VRP, ou Problema de Roteirização de Veículos, é o nome de uma classe de problemas para definir a sequência de visita a clientes dispersos geograficamente com um conjunto finito de veículos a partir de um depósito comum. Para resolver este problema e analisar os resultados obtidos, foi desenvolvido um algoritmo utilizando a meta-heurística *Variable Neighborhood Descent* (VND), ou Descida em Vizinhança Variável, o qual foi aplicado em instâncias conhecidas na literatura e realizado um *benchmarking* com outros algoritmos. Esse problema aplica-se na prática em coleta de peças automobilísticas, coleta de lixo industrial, coleta de lixo residencial, limpeza de ruas, e entre outras situações. Os VRPs receberam muita atenção nos últimos anos devido a sua aplicabilidade e sua importância econômica na determinação de estratégias eficientes, com o objetivo de reduzir os custos operacionais. Os resultados obtidos com o algoritmo proposto foram próximos dos algoritmos estudados no *benchmarking* realizado. Contudo, superando alguns destes algoritmos em apenas uma das instâncias das oito instâncias utilizadas.

**Palavras-chave:** Problema de Roteirização de Veículos. Roteirização. Frota Heterogênea. Algoritmo VND.

## ABSTRACT

In logistic distribution process is necessary deliver goods and services to geographically dispersed customers, in this process is found the *Vehicle Routing Problem (VRP)*. The *Vehicle Routing Problem (VRP)*, is the name of a problems class to define a sequence of visits to customers geographically dispersed with a finite set of vehicles from a common depot. To solve this problem, a algorithm was developed using the *Variable Neighborhood Descent (VND)* metaheuristic, comparing the results with some literature instances. The problem applies in practice on auto parts collection, industrial trash collection, residential trash collection, street cleaning, and other situations. The VRPs received many attention in the lasts years due to applicability and the economic importance in efficient strategies determination, with the objective of reduce the operational costs. The results of proposed algorithm were competitive to the algorithms studied. However, overcoming some of these algorithms in only one instance of the eight instances used.

**Keywords:** VRP. Routing. HVRP. VND.

## LISTA DE ILUSTRAÇÕES

Figura 1 – Metodologia . . . . .	19
Figura 2 – Funcionamento do 2-Opt . . . . .	27
Figura 3 – Funcionamento do 3-Opt . . . . .	27
Figura 4 – Funcionamento do SC . . . . .	29
Figura 5 – Funcionamento do SE . . . . .	29
Figura 6 – Funcionamento do SR . . . . .	30

## LISTA DE TABELAS

Tabela 1 – Resultado dos algoritmos . . . . .	44
Tabela 2 – Resultado dos algoritmos VND e VNS Ascendente . . . . .	45

## LISTA DE QUADROS

Quadro 1 – Descrição de algumas variantes do HVRP . . . . .	22
---	----

## LISTA DE ALGORITMOS

Algoritmo 1 – Construção de roteiros . . . . .	35
Algoritmo 2 – <i>Variable Neighborhood Descent</i> (VND) . . . . .	36
Algoritmo 3 – Aplica Buscas Locais . . . . .	37
Algoritmo 4 – Usar Mais Veículos . . . . .	38
Algoritmo 5 – VNS Ascendente . . . . .	39
Algoritmo 6 – Clarke e Wright HVRP . . . . .	40
Algoritmo 7 – VND Pre Climb . . . . .	40
Algoritmo 8 – Climb . . . . .	41
Algoritmo 9 – VND Pos Climb . . . . .	42
Algoritmo 10 – Construir Pior Caso . . . . .	43

## LISTA DE ABREVIATURAS E SIGLAS

AM	<i>Adaptative Memory</i>
CPU	<i>Central Process Unit</i>
CVRP	<i>Capacitated VRP</i>
FSM	<i>Fleet Size and Mix</i>
FSMD	<i>Fleet Size and Mix VRP with Vehicle Dependent Routing Costs</i>
FSMF	<i>Unlimited Fleet with only Fixed Costs</i>
FSMFD	<i>Fleet Size and Mix VRP with Fixed Costs and Vehicle Dependent Routing Costs</i>
FSMFV	<i>Unlimited Fleet with Fixed and Variable Costs</i>
FSMV	<i>Unlimited Fleet with only Variable Costs</i>
FSMVRP	<i>Fleet Size and Mix Vehicle Routing Problem</i>
GAMS	<i>General Algebraic Modeling System</i>
GENIUS	<i>GENI (Generalized Insertion Procedure) and US (Unstringing e Stringing)</i>
GEROCA	<i>Generalized Route Construction Algorithm</i>
GT	<i>Giant Tour</i>
HFFVRP	<i>Heterogeneous Fixed Fleet Vehicle Routing Problem</i>
HFVRP	<i>Heterogeneous Fleet VRP</i>
HVRP	<i>Heterogeneous VRP</i>
HVRPD	<i>HVRP with vehicle Dependent routing costs</i>
HVRPFD	<i>HVRP with Fixed costs and vehicle Depent routing costs</i>
HVRPFV	<i>Limited Fleet with Fixed and Variable Costs</i>
HVRPV	<i>Limited Fleet with Variable Costs but without Fixed Costs</i>
ILS	<i>Iterated Local Search</i>
MD-FSMVRP-TW	<i>Multi-Depot Fleet Size Mix Vehicle Routing Problem with Time Window</i>
MFVRP	<i>Mix Fleet VRP</i>
SC	<i>String Cross</i>
SE	<i>String Exchange</i>
SM	<i>String Mix</i>
SR	<i>String Relocation</i>
TS	<i>Tabu Search</i>
TSP	<i>Traveling-Salesman Problem</i>
VND	<i>Variable Neighborhood Descent</i>

VNS	<i>Variable Neighborhood Search</i>
VRP	<i>Vehicle Routing Problem</i>
VRPSPD	<i>VRP with Simultaneous Pickup and Delivery</i>
VRPTW	<i>VRP with Time Windows</i>

## LISTA DE SÍMBOLOS

$G$  Grafo direcionado

$V$  Conjunto de vértices

$A$  Conjunto de arcos

$V'$  Conjunto de vértices sem o depósito

$M$  Conjunto dos tipos dos veículos

$NV$  Número de veículos da frota

$m_u$  Quantidade de veículos disponíveis do tipo  $u$

$Q_u$  Capacidade do veículo do tipo  $u$

$f_u$  Custo fixo do veículo de tipo  $u$

$c_{ij}$  Custo do arco do vértice  $i$  para o vértice  $j$

$q_i$  Demanda do vértice  $i$

$r_u$  Custo variável do veículo de tipo  $u$

$x_{ij}^v$  Variável binária de três índices

$d_i$  Variável inteira que serve como parte da restrição de quebra de sub roteiros

## SUMÁRIO

<b>1</b>	<b>INTRODUÇÃO</b> . . . . .	<b>16</b>
<b>1.1</b>	<b>Justificativa</b> . . . . .	<b>17</b>
<b>1.2</b>	<b>Escopo do trabalho</b> . . . . .	<b>18</b>
<b>1.3</b>	<b>Objetivos</b> . . . . .	<b>18</b>
<b>1.3.1</b>	<i>Gerais</i> . . . . .	<b>18</b>
<b>1.3.2</b>	<i>Específicos</i> . . . . .	<b>18</b>
<b>1.4</b>	<b>Metodologia</b> . . . . .	<b>18</b>
<b>1.5</b>	<b>Organização do trabalho</b> . . . . .	<b>19</b>
<b>2</b>	<b>FUNDAMENTAÇÃO TEÓRICA</b> . . . . .	<b>20</b>
<b>2.1</b>	<b>Formulação do problema</b> . . . . .	<b>22</b>
<b>2.2</b>	<b>Heurísticas relacionadas</b> . . . . .	<b>25</b>
<b>2.2.1</b>	<i>Heurísticas de construção</i> . . . . .	<b>25</b>
<b>2.2.2</b>	<i>Heurísticas de melhoria</i> . . . . .	<b>26</b>
<b>2.3</b>	<b>Meta-heurísticas relacionadas</b> . . . . .	<b>30</b>
<b>3</b>	<b>ALGORITMO PROPOSTO</b> . . . . .	<b>35</b>
<b>4</b>	<b>RESULTADOS</b> . . . . .	<b>44</b>
<b>5</b>	<b>CONCLUSÕES E TRABALHOS FUTUROS</b> . . . . .	<b>46</b>
<b>5.1</b>	<b>Conclusões</b> . . . . .	<b>46</b>
<b>5.2</b>	<b>Trabalhos futuros</b> . . . . .	<b>46</b>
	<b>REFERÊNCIAS</b> . . . . .	<b>47</b>

## 1 INTRODUÇÃO

Dentre os principais processos da logística, destaca-se o processo de gerenciamento efetivo do fluxo total de bens, desde a aquisição da matéria-prima até o fim da distribuição dos produtos acabados para o cliente final (GOLDEN, 1975). É durante esse processo que acontece a distribuição de produtos para clientes dispersos geograficamente.

Christofides (1976) diz que o *Vehicle Routing Problem* (VRP), ou Problema de Roteirização de Veículos, é um nome dado para toda uma classe de problemas envolvendo a visita a clientes por veículos. No VRP, um conjunto de veículos localizados no depósito, é otimamente roteirizado para atender clientes com demandas conhecidas de acordo com as restrições dos veículos (BALDACCI *et al.*, 2008). O VRP tem um papel central nos campos de distribuição física e logística, preocupando-se com a roteirização dos veículos para minimizar a distância viajada (ou tempo necessário, custo incorrido, dentre outros objetivos almejados).

O *Traveling-Salesman Problem* (TSP), ou Problema do Caixeiro Viajante, pode ser generalizado no VRP pela adição de algumas condições (DANTZIG; RAMSER, 1959). O leitor mais interessado em TSP pode consultar Dantzig *et al.* (1954) para mais detalhes. Laporte (1992) define o VRP como o problema de concepção de entrega ideal de um ou muitos depósitos para um número de cidades/clientes geograficamente dispersos, sujeitos a diferentes restrições. Já Bodin *et al.* (1981), afirmam que o VRP requer um conjunto de rotas de entrega partindo do depósito central para vários pontos de demanda, cada ponto tendo requisitos de serviço, com o objetivo de minimizar a distância total coberta pela frota. Observando a utilização de veículos com capacidade limitada, o VRP pode ser chamado de CVRP, ou Problema de Roteirização de Veículos Capacitados.

Ao considerar uma frota de veículos com capacidades (ou mesmo custos) diferentes, tem-se o *Heterogeneous VRP* (HVRP), ou Problema de Roteirização de Veículos Heterogêneos (GENDREAU *et al.*, 1999). Outra definição deste mesmo problema é chamado de HFVRP, ou de Problema de Roteirização de Veículos com Frota Heterogênea.

O VRP e HVRP são descritos de várias formas na literatura, como será mostrado no Capítulo 2. O HVRP abordado neste trabalho utiliza uma frota de veículos ilimitada de cada tipo de veículo.

## 1.1 Justificativa

No gerenciamento de distribuição, é necessário determinar uma combinação de custo mínimo de rotas dos veículos, de acordo com uma sequência de clientes, sujeitas a restrições, onde o problema geralmente estudado é onde todos os veículos são idênticos (GENDREAU *et al.*, 1999).

Dado que a maioria dos problemas de roteirização e escalonamento são *NP-Hard*, que é uma classe de problemas que são pelo menos tão difíceis quanto os problemas mais difíceis em *NP*, abordagens conhecidas para resolver esses problemas otimamente sofrem de um crescimento exponencial em carga computacional de acordo com o tamanho do problema. Quando tratados como problemas *NP-Hard*, frequentemente recorre-se à heurísticas ou procedimentos aproximados para obter solução próxima da solução ótima ao invés da solução ótima. Um algoritmo heurístico é um procedimento que usa a estrutura do problema em uma abordagem matemática (e usualmente intuitiva) para prover soluções próximas da solução ótima. Uma heurística é considerada efetiva se as soluções que provê são consistentemente próximas da solução ótima (BODIN *et al.*, 1981, tradução nossa).

Lenstra e Kan (1981) provaram que o *Fleet Size and Mix Vehicle Routing Problem* (FSMVRP), ou Problema de Roteirização de Veículos com Frota de Tamanho Misto, com custos fixos é NP-Difícil. A roteirização de veículos é uma área onde uma solução para um determinado tipo de problema, seus dados e condições, pode não ser adequada para outro problema similar (TEIXEIRA; CUNHA, 2002).

A alocação de veículos heterogêneos para clientes com diferentes demandas durante a roteirização possui uma complexidade maior do que a simples roteirização com frota homogênea que não considera restrições de diferentes capacidades dos veículos. Diferente da simples roteirização, em que não é necessário se preocupar com alocação de veículos em roteiros. Quando as capacidades dos veículos são heterogêneas, realiza-se verificações para estabelecer a quantidade de clientes atendidos por veículo de uma dada capacidade e qual veículo de determinada capacidade será usado para atender um conjunto de clientes.

O VRP tem sido considerado como um elemento significativo na tratamento logístico. Portanto, uma seleção apropriada de rotas de veículo é extremamente importante para melhorar os benefícios econômicos das operações logísticas. O VRP é bem conhecido por ser uma das tarefas difíceis da pesquisa operacional (MOHAMMED *et al.*, 2012, tradução nossa).

Nos poucos 50 anos em que o trabalho de Dantzig e Ramser (1959) surgiu, vários trabalhos na área tem explorado dramaticamente, e hoje, uma busca no Google *Scholar* com as

palavras *vehicle routing problem* resulta em mais de 21,700 registros (GOLDEN S. RAGHAVAN, ).

## 1.2 Escopo do trabalho

Esse trabalho propõe um algoritmo para o HVRP usando a meta-heurística *Variable Neighborhood Search* (VNS), ou Busca em Vizinhança Variável, com base em uma revisão da literatura relacionada ao HVRP, VRP e TSP, e apresentar um *benchmarking* do algoritmo com algumas instâncias da literatura.

## 1.3 Objetivos

### 1.3.1 Gerais

Propor um algoritmo para o *Heterogeneous VRP* (HVRP).

### 1.3.2 Específicos

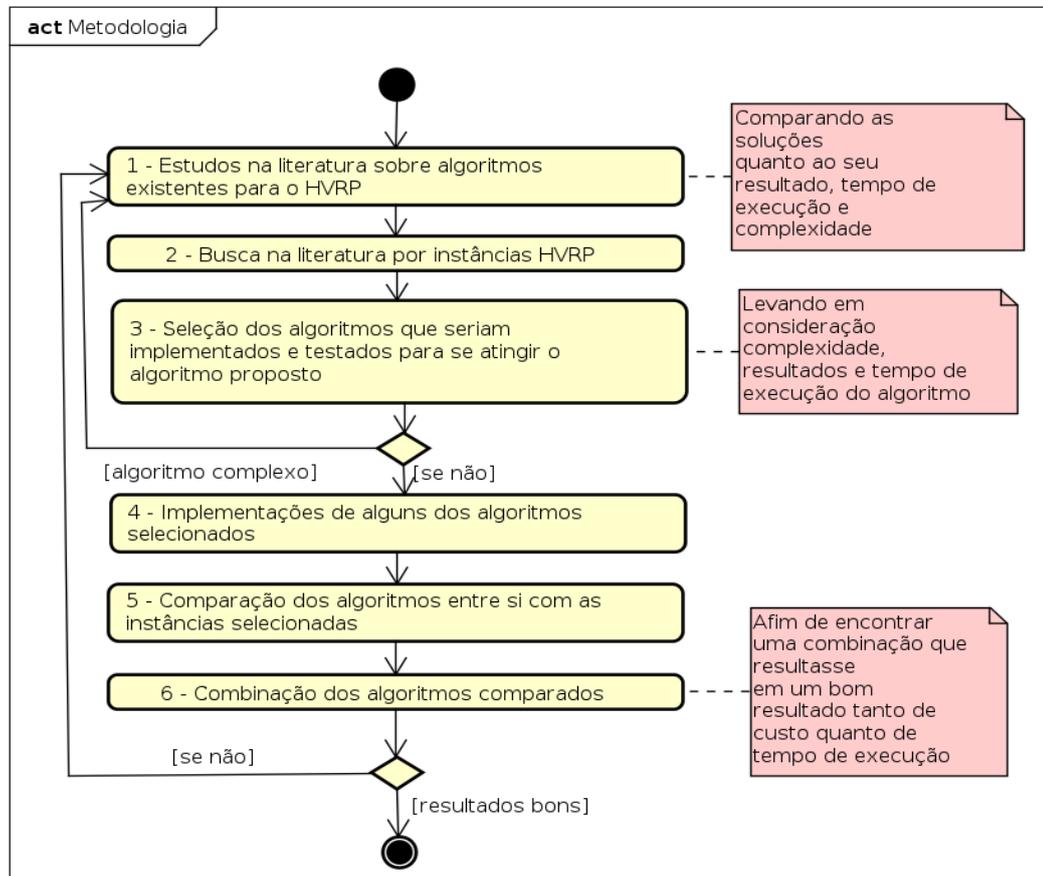
- Apresentar uma revisão do HVRP mostrando algumas definições e soluções usadas na literatura;
- Identificar os principais algoritmos relacionados ao problema;
- Desenvolver o algoritmo proposto nesse trabalho;
- Fazer um *benchmarking* do algoritmo com instâncias da literatura.

## 1.4 Metodologia

Para alcançar o algoritmo proposto neste trabalho, foram seguidos os passos da Figura 1. Na primeira etapa, foram realizados estudos de algoritmos para o HVRP a partir da literatura, comparando as soluções já existentes utilizando os critérios de resultado, tempo de execução e complexidade. Na etapa 2, foi feita uma busca por instâncias que serviram de comparativo para os testes. Na etapa 3, ocorreu uma seleção dos algoritmos a serem implementados e testados, se encontrado algum algoritmo com alta complexidade, então retorna-se para etapa 1, se não, executou-se a etapa 4. Na etapa 4 foi feita a implementação de alguns dos algoritmos selecionados nas etapas 1 e 3. Na etapa 5 comparou-se os resultados dos algoritmos implementados na etapa 3. Finalmente, na etapa 6, foram feitas combinações dos algoritmos comparados na etapa 5, com

o objetivo de encontrar uma combinação que produza boas soluções em tempo eficiente.

Figura 1 – Metodologia



Fonte: Elaborada pelo autor.

## 1.5 Organização do trabalho

O Capítulo 2 contém a fundamentação teórica, com uma descrição mais detalhada do HVRP, e das abordagens existentes para o problema em questão. O Capítulo 3 detalha o algoritmo implementado, descrevendo seu funcionamento. O Capítulo 4 mostra os resultados do algoritmo para as instâncias selecionadas, comparando com resultados de outros autores e descrevendo como e sobre quais condições o experimento foi realizado. Por fim o Capítulo 5 apresenta as conclusões do algoritmo e sugestões de trabalhos futuros.

## 2 FUNDAMENTAÇÃO TEÓRICA

As variantes do VRP geralmente são problemas de projetar um conjunto de rotas que iniciam e terminam no depósito central para um conjunto de veículos, onde cada rota passa por clientes com demandas conhecidas, cuja soma não deve exceder a capacidade do veículo atribuído a essa rota (GOLDEN, 1975).

Golden *et al.* (1984) afirmam que há três casos do VRP frequentemente aplicados na prática:

1. *Vehicle Routing Problem* (VRP): Envolve determinar o número de veículos, onde todos têm a mesma capacidade, considerando apenas o custo variável (da rota). O objetivo é minimizar o custo total das rotas, que é a distância total viajada pela frota de veículos. Esse problema tem recebido extensiva atenção em pesquisa e aplicação em distribuição de jornais, roteirização de ônibus escolar, coleta de lixo municipal, distribuição de óleo e gasolina, e despacho de caminhão em uma variedade de indústrias.
2. *The Fleet Size Problem* ou O Problema de Tamanho de Frota: A decisão de roteirização é frequentemente abordada por questões de tamanho da frota, que é o número de veículos necessários para satisfazer a demanda. Nesse caso, os custos fixos dos veículos e os custos variáveis da rota são considerados. Assume-se que os veículos da frota têm características de operação idênticas (por exemplo, custo e capacidade). O objetivo é determinar um tamanho de frota econômico.
3. *The Fleet Size and Mix Problem* ou O Problema de Tamanho de Frota Mista: Esse caso é uma generalização do anterior em que assume-se que todos os veículos podem ter custo e capacidade diferentes.

Os HVRPs podem ser divididos de acordo com diferentes recursos, incluindo a disponibilidade do veículo (quantidade limitada ou ilimitada de veículos) e os custos do veículo (fixos, mesmo custo para todos os veículos, ou variáveis, veículos com custos diferentes) (PENNA *et al.*, 2013).

Quando a frota é limitada, o número de veículos e sua capacidade são conhecidos de antemão, e as rotas da solução devem considerar essa disponibilidade. No caso de uma frota ilimitada, o número necessário de veículos para atender às demandas dos clientes é inicialmente desconhecido, e o problema deve determinar a composição da frota considerando o custo e a capacidade dos veículos (COELHO *et al.*, 2016, tradução nossa).

Derbel *et al.* (2017) dizem que, as duas seguintes características são consideradas

para classificar diferentes variações do problema na literatura: a limitação da frota e o tipo dos custos. Sendo que a maior parte dos trabalhos na literatura, como o de Derbel *et al.* (2017), abordam cinco variantes. As diferentes variantes que são nomeadas pelo uso de duas siglas: HVRP para problemas com número limitado de veículos de cada tipo e *Fleet Size and Mix* (FSM), ou Frota de Tamanho Mista, para variantes com o número de veículos ilimitado seguidas de: “F” para problemas com custo fixo e “V” para aqueles com custo variável.

- *Limited Fleet with Fixed and Variable Costs* (HVRPFV), ou Frota Limitada com Custos Variáveis e Fixos;
- *Limited Fleet with Variable Costs but without Fixed Costs* (HVRPV), ou Frota Limitada com Custos Variáveis mas sem Fixos;
- *Unlimited Fleet with Fixed and Variable Costs* (FSMFV), ou Frota Ilimitada com Custos Variáveis e Fixos;
- *Unlimited Fleet with only Fixed Costs* (FSMF), ou Frota Ilimitada apenas com Custos Fixos;
- *Unlimited Fleet with only Variable Costs* (FSMV), ou Frota Ilimitada apenas com Custos Variáveis.

MirHassani e Saadati (2014) citam duas variantes do HVRP com base em estudos da literatura:

Na primeira, supõe-se que o número de veículos de cada tipo é ilimitado e o objetivo é encontrar o conjunto ótimo de veículos. Esse problema é chamado de FSMVRP, que é usualmente adotado para decisões táticas onde a frota ainda não está paga e a seleção do número de veículos tem que ser realizada. No segundo tipo, há uma frota fixa de veículos, o que significa que a frota e o número de veículos de cada tipo é fixo. Esse problema é chamado de *Heterogeneous Fixed Fleet Vehicle Routing Problem* (HFFVRP), ou Problema de Roteirização de Veículos com Frota Heterogênea Fixa, que é usualmente aplicada em decisões operacionais, onde é necessário computar viagens e atribuir veículos para elas. O FSMVRP pode ser considerado como um HFFVRP particular, onde o número de veículos de cada tipo é igual ao número de clientes (MIRHASSANI; SAADATI, 2014, tradução nossa).

Liu (2013) classifica o HVRP em duas variantes: *HVRP with Fixed costs and vehicle Depent routing costs* (HVRPFD), ou HVRP com custos Fixos e custos Dependentes da rota, e *HVRP with vehicle Dependent routing costs* (HVRPD), ou HVRP com custo Dependente da rota, que não considera o custo fixo.

Baldacci *et al.* (2010) classificam algumas variantes do HVRP, essa classificação encontra-se no Quadro 1.

Quadro 1 – Descrição de algumas variantes do HVRP

Variante	Tamanho da frota	Custos fixos	Custos da rota
HVRPFD	Limitada	Considerados	Dependente
HVRPD	Limitada	Não considerados	Dependente
FSMFD	Ilimitada	Considerados	Dependente
FSMD	Ilimitada	Não considerados	Dependente
FSMF	Ilimitada	Considerados	Independente

Fonte: Baldacci *et al.* (2010).

## 2.1 Formulação do problema

Nesse capítulo serão apresentados os elementos VRP para que posteriormente possamos descrever o HVRP. Gendreau *et al.* (1994) descrevem o VRP como um grafo direcionado  $G = (V, A)$ , onde o conjunto  $V = \{0, 1, \dots, n\}$  é composto por  $n + 1$  vértices, tal que o vértice 0 denota o depósito onde a frota com  $m$  veículos idênticos de capacidade  $Q$  está localizada. O conjunto  $V' = V \setminus \{0\}$  é composto pelos vértices restantes que representam os  $n$  clientes, onde cada cliente  $i \in V'$  tem uma demanda não negativa  $q_i$  e é visitado exatamente uma vez por exatamente um veículo. O conjunto  $A = \{(i, j) : i, j \in V, i \neq j\}$  é o conjunto de arcos, onde cada arco  $(i, j)$  representa o percurso do cliente  $i$  até o cliente  $j$ . As rotas começam e terminam no depósito e a demanda total em cada rota não pode exceder a capacidade  $Q$ . A frota de veículos é composta por  $NV$  veículos.

Bodin *et al.* (1981) descrevem um modelo de programação linear inteira para VRP, em que a variável  $x_{ij}^v$  tem valor 1 se o arco  $(i, j) \in A$  faz parte da solução ótima e é visitado pelo veículo  $v$ , e o valor 0 caso contrário.

$$\text{Minimize } \sum_{i=0}^n \sum_{j=0}^n \sum_{v=1}^{NV} c_{ij} x_{ij}^v \quad (2.1)$$

$$\text{Sujeito a } \sum_{i=0}^n \sum_{v=1}^{NV} x_{ij}^v = 1 \quad (j \in V') \quad (2.2)$$

$$\sum_{j=0}^n \sum_{v=1}^{NV} x_{ij}^v = 1 \quad (i \in V') \quad (2.3)$$

$$\sum_{i=0}^n x_{ip}^v - \sum_{j=0}^n x_{pj}^v = 0 \quad (v = 1, \dots, NV; p \in V) \quad (2.4)$$

$$\sum_{i=0}^n q_i \left( \sum_{j=0}^n x_{ij}^v \right) \leq Q \quad (v = 1, \dots, NV) \quad (2.5)$$

$$\sum_{i=1}^n x_{0i}^v \leq 1 \quad (v = 1, \dots, NV) \quad (2.6)$$

$$\sum_{i=1}^n x_{i0}^v \leq 1 \quad (v = 1, \dots, NV) \quad (2.7)$$

$$\sum_{i \in a} \sum_{j \in a} x_{ij}^v \leq |a| - 1 \quad (v = 1, \dots, NV; a \in P(V')) \quad (2.8)$$

$$x_{ij}^v \in \{0, 1\} \quad (i, j \in V; v = 1, \dots, NV) \quad (2.9)$$

O modelo é um índice duplo de fluxo de veículos que usa  $O(n^2)$  variáveis binárias  $x$ , para indicar se um veículo passa ou não pelo arco na solução ótima. A Função Objetivo 2.1 busca a minimização do custo total, onde  $n$  é o número de nós e  $NV$  o número de veículos. As Equações 2.2 e 2.3 dizem que para cada cliente, há exatamente um veículo percorrendo um arco que chega nele e exatamente um percorrendo um arco que parte dele. A Equação 2.4 diz que se um veículo atender um cliente, então o veículo deve partir desse cliente. Na Equação 2.5, está a representação da restrição de capacidade dos veículos. As Equação 2.6 e 2.7 garantem que a disponibilidade de veículos não é excedida. Por fim, a Equação 2.8 garante a ausência de subciclos na solução,  $P(V')$  é o conjunto das partes de  $V'$ .

O HVRP difere do VRP clássico na medida em que trata uma frota de veículos tendo várias capacidades, custos fixos e variáveis. Assim sendo, o HVRP envolve projetar um conjunto de rotas de veículos, cada uma começando e terminando no depósito central, para uma frota de veículos com diferentes capacidades (CHOI; TCHA, 2007, tradução nossa).

O HVRP segue os mesmos princípios do VRP, com exceção da questão dos veículos. De acordo com Subramanian *et al.* (2012), no HVRP o vértice 0 denota o depósito onde a frota de veículos com diferentes capacidades está localizada. Essa frota é composta por  $m$  diferentes

tipos de veículos. O conjunto  $M = \{1, \dots, m\}$  possui as seguintes características: Para cada  $u \in M$ , há uma quantidade de  $m_u$  veículos disponíveis, cada um com a capacidade  $Q_u$  e um custo fixo denotado por  $f_u$ . Finalmente, para cada arco  $(i, j) \in A$  há um custo  $c_{ij}^u = d_{ij}r_u$ , onde  $d_{ij}$  é a distância percorrida no arco  $(i, j)$  e  $r_u$  é o custo variável de viagem por unidade de distância para um veículo do tipo  $u$ .

Golden *et al.* (1984) descrevem outro modelo em programação matemática linear para o HVRP. A variável  $x_{ij}^u$  tem valor 1 se veículo de tipo  $u$  percorre o arco  $(i, j) \in A$  na solução ótima, e o valor 0 caso contrário, e a variável real  $d_i$  é a variável de fluxo de comodidade associado com o cliente  $i$ , que fornece a demanda total que um veículo tem atendido na sua rota depois de chegar no cliente  $i$  (a demanda do cliente  $i$  está incluída).

$$\text{Minimize } \sum_{u=1}^m f_u \sum_{j=1}^n x_{0j}^u + \sum_{u=1}^m \sum_{i=0}^n \sum_{j=0}^n c_{ij} x_{ij}^u \quad (2.10)$$

$$\text{Sujeito a } \sum_{u=1}^m \sum_{i=0}^n x_{ij}^u = 1 \quad (u \in M; j \in V') \quad (2.11)$$

$$\sum_{i=0}^n x_{ip}^u - \sum_{j=0}^n x_{pj}^u = 0 \quad (p \in V') \quad (2.12)$$

$$d_0 = 0 \quad (2.13)$$

$$d_j - d_i \geq (q_j + a_t) \sum_{u=1}^m x_{ij}^u - Q_t \quad (i \in V; j \in V') \quad (2.14)$$

$$d_j \geq \sum_{u=1}^m \sum_{i=0}^n a_u x_{ij}^u \quad (j \in V'; u \in M) \quad (2.15)$$

$$x_{ij}^u \in \{0, 1\} \quad (i \in V; j \in V'; u \in M) \quad (2.16)$$

Esse modelo é uma formulação compacta que assume que há um número infinito de cada tipo de veículo disponível. No termo  $\sum_{j=1}^n x_{0j}^u$  na Equação 2.10 está a representação do número de veículos do tipo  $u$  usados, e o termo  $\sum_{u=1}^m f_u \sum_{j=1}^n x_{0j}^u$  representa o custo fixo total. O termo  $\sum_{u=1}^m \sum_{i=0}^n \sum_{j=0}^n c_{ij} x_{ij}^u$  representa o custo total variável. As Equações em 2.11 e 2.12 garantem que cada cliente é visitado exatamente uma única vez e que um veículo que chega em um cliente deve partir dele. As Equações 2.13 e 2.14 definem a variável real  $d_i$  e servem como restrições de quebra de sub roteiros, o termo  $Q_u$  é a capacidade do veículo de tipo  $u$ , tal que  $(Q_1 < Q_2 < \dots < Q_t)$ , ou seja,  $t$  é o tipo de veículo com maior capacidade. No termo  $d_0 = 0$  da Equação 2.13, a demanda total atendida no depósito será sempre 0. Os próximos conjuntos de restrições garantem que as capacidades dos veículos não serão excedidas. A Equação 2.15 diz

que a demanda cumulativa em qualquer localização do cliente é limitada pela capacidade do veículo que está servindo aquele cliente.

## 2.2 Heurísticas relacionadas

Algoritmos heurísticos são métodos de tentativa e erro que não garantem encontrar a solução ótima, mas são projetados para encontrar soluções próximas da ótima em uma fração de tempo mais eficiente do tempo requerido pelo método ótimo (KARADIMAS *et al.*, 2008).

O termo heurística é usado para algoritmos que realizam busca por uma melhor solução num espaço reduzido de soluções. No entanto, esses algoritmos usualmente executam uma busca de maneira rápida e eficiente. Algumas vezes, esses algoritmos podem encontrar uma solução ótima, mas o algoritmo ainda é chamado de heurístico, por que não é garantido que uma solução ótima será sempre encontrada.

Laporte *et al.* (2000) revisam algumas das mais importantes famílias de heurísticas para VRP, apresentando heurísticas clássicas e alguns algoritmos baseados em *Tabu Search* (TS), ou Busca Tabu, além de apresentar comparativo dos resultados de teste das heurísticas para 14 instâncias. Nesse trabalho serão apresentados alguns algoritmos, classificados em heurísticas de construção e heurísticas de melhoria, abordados por Laporte *et al.* (2000).

### 2.2.1 Heurísticas de construção

- Heurística de ganhos ou economias ou Algoritmo de Clarke e Wright (CLARKE; WRIGHT, 1964): É um algoritmo de construção que parte do pior caso, onde há uma rota com apenas um cliente, e então a partir do conceito de ganhos, o algoritmo faz mesclas das rotas.
- Métodos de melhoria sequencial: Mole e Jameson (1976) desenvolveram um novo algoritmo de construção sequencial de rotas, que expande uma rota por vez, e logo após a construção, aplica o método de melhoria 3-Opt. A heurística de melhoria 3-Opt será apresentada na Subseção 2.2.2.
- Algoritmo de varredura: É um algoritmo de construção aplicado para instâncias planares de VRP. O agrupamento de clientes viável é construído pelo giro de um raio centralizado no depósito. Algumas implementações incluem uma fase de pós-otimização. Pelo conhecimento de Laporte *et al.* (2000) a primeira menção desse método foi encontrada no livro de Wren (1971) e no trabalho de Wren e Holliday (1972), mas o algoritmo de varredura é

comumente atribuído a Gillett e Miller (1974), que popularizaram o algoritmo.

- Algoritmo *cluster first-route second* ou agrupar primeiro e roteirizar depois: Bodin *et al.* (1981) descrevem essa estratégia da seguinte forma. Primeiro agrupa-se os nós e então projeta-se rotas econômicas sobre cada grupo como segundo passo. Alguns exemplos de aplicação dessa estratégia podem ser encontrados em Fisher e Jaikumar (1981), Bramel e Simchi-Levi (1995), Gillett e Johnson (1976) e Karp (1977).

Sinruk e Tangmo (2017) modificaram o algoritmo *Cluster First-Route Second*, para o HVRP em lojas de materiais de construção de varejo. Primeiro, os clientes são agrupados dentro de uma solução viável pela seleção de agrupamento de cliente. Segundo, o modelo matemático do problema é desenvolvido e resolvido otimamente usando o *software General Algebraic Modeling System (GAMS)*, ou Sistema de Modelagem Algébrica Geral, que é um sistema de modelagem para otimização matemática, como programação linear inteira mista.

- Algoritmos de *route first-cluster second*: Golden *et al.* (1984) diz que *Giant Tour (GT)*, ou Grande Roteiro, são exemplos de *route first-cluster second*. No GT há duas fases, na primeira, um método é utilizado para gerar um roteiro que visita todos os clientes. Esse roteiro será particionado em outros sub-roteiros, cada um satisfazendo as restrições do problema. Os roteiros são compostos de segmentos iguais aos do roteiro original, porém o fim e o começo de cada sub-roteiro é o depósito. Algumas abordagens desse algoritmo aplicam algoritmos de melhoria, tais como 3-Opt e 2-Opt para melhorar a solução após a divisão em sub-roteiros. As heurísticas de melhoria 2-Opt e 3-Opt serão apresentadas na Subseção 2.2.2.

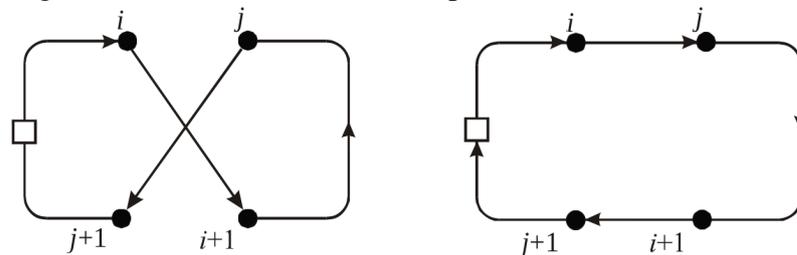
### 2.2.2 Heurísticas de melhoria

Nesse tipo de heurística, aplica-se algoritmos de melhoria em cada roteiro associado a um veículo. Nesse caso podem ser aplicadas heurísticas de melhoria TSP, separadamente, ou em muitas rotas ao mesmo tempo, sendo também necessário um procedimento para explorar estruturas multirrotas.

- Heurísticas de melhoria TSP: Nesse tipo de melhoria, podem ser feitas trocas dentro de um roteiro específico, pois as mudanças no roteiro não irão afetar as restrições do veículo associado para aquela rota. Nesse trabalho serão abordados o algoritmo de Lin (1965), o de Or (1976) e os de Ascheuer *et al.* (2001).

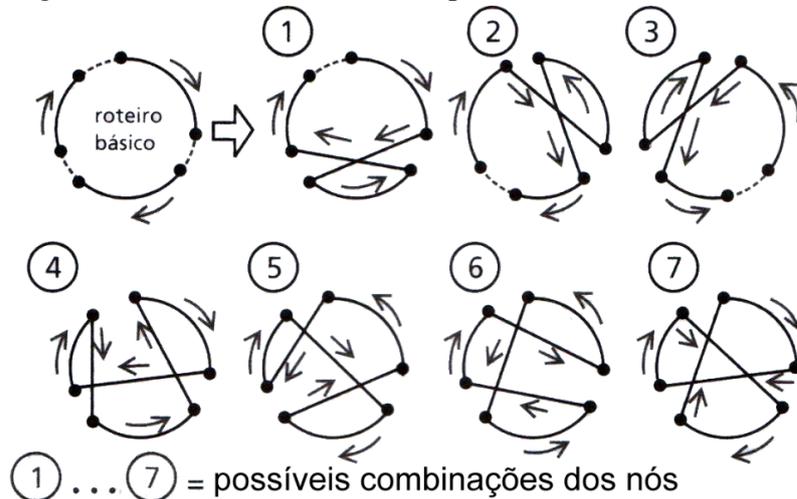
- $\lambda$ -Opt: A maior parte dos procedimentos de melhoria para TSP podem ser descritos por esse algoritmo de Lin (1965). Nesse algoritmo,  $\lambda$  arcos são removidos do roteiro e os  $\lambda$  arcos removidos são reconectados em todas as maneiras possíveis. Se qualquer alteração é identificada com o custo menor que a da solução antes de reconectar os arcos, então o novo roteiro passa a ser a solução atual. Checar o  $\lambda$ -Opt para uma solução pode ser alcançado em tempo  $O(n^\lambda)$ . Muitas modificações desse algoritmo já foram feitas, inclusive por Renaud *et al.* (1996). As Figuras 2 e 3 mostram o funcionamento do 2-Opt e do 3-Opt. Para mais detalhes sobre a melhoria 3-Opt, consulte Lin e Kernighan (1973).

Figura 2 – Funcionamento do 2-Opt



Fonte: Bräysy e Gendreau (2005).

Figura 3 – Funcionamento do 3-Opt



Fonte: Novaes (2007).

- Or-Opt: Desenvolvida por Or (1976), que consiste em remover dois ou três vértices de um roteiro e reconectá-los em outro local do roteiro.
- Heurística *Swap*: Dado um roteiro viável  $T = (v_0, v_1, \dots, v_{n+1})$  procura-se na sequência uma troca de dois nós adjacentes  $v_i$  e  $v_{i+1}$ ,  $i = 1, \dots, n - 1$  que resulta em uma solução viável com melhor custo. Esse procedimento é repetido até que não encontre

mais melhorias.

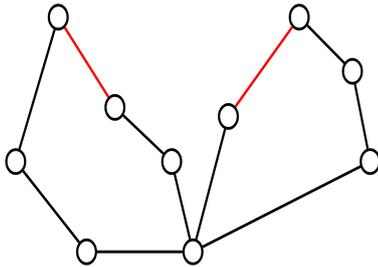
- Heurística *Two-Node-Exchange*: É uma generalização da Heurística *Swap* onde qualquer dois nós (não apenas adjacentes) na sequência são trocados. Se o resultado é um roteiro viável com melhor custo, então a troca é aceita. Esse procedimento é repetido até que nenhuma melhoria possa ser encontrada.
- Heurística *Node-Reinsertion*: Dado um roteiro viável  $(v_0, v_1, \dots, v_i, v_j, v_k, \dots, v_{n+1})$  e qualquer nó interior  $v_j$ , se constrói um roteiro parcial  $T = (v_0, v_1, \dots, v_i, v_k, \dots, v_{n+1})$  pela remoção de  $v_j$ . Então procura-se reinserir  $v_j$  na melhor posição de  $T$ , onde a nova sequência  $T$  é viável. Se o resultado é melhor,  $T$  é aceito como novo roteiro e o processo se repete até não encontrar mais melhorias.
- Heurística *Arc-Reinsertion*: Dado um roteiro viável  $(v_0, v_1, \dots, v_i, v_j, v_k, v_l, \dots, v_{n+1})$  é construído um roteiro parcial pela remoção de dois nós consecutivos  $v_j$  e  $v_k$ . Então procura-se reinserir o par  $(v_j, v_k)$  em qualquer posição de  $T$ , onde a nova sequência  $T$  é viável. Se o resultado possui melhor custo, então  $T$  é aceito, e o procedimento repete-se até não encontrar mais melhorias.
- Heurística *Arc-Reversal*: Dado um roteiro viável  $(v_0, \dots, v_j, v_k, \dots, v_l, v_m, \dots, v_{n+1})$  constrói-se um roteiro  $T = (v_0, \dots, v_j, v_l, \dots, v_k, v_m, \dots, v_{n+1})$  pela inversão do sub-caminho  $(v_k, \dots, v_l)$ , onde  $T$  é um roteiro viável. Se o resultado possui um custo melhor, então  $T$  é aceito como novo roteiro e repete-se até não conseguir melhorias.
- Heurística *Or-Exchange*: Dado um roteiro viável  $(v_0, v_1, \dots, v_i, \dots, v_j, \dots, v_{n+1})$  remove-se um sub-caminho  $(v_i, \dots, v_j)$  e se procura reinserir entre dois nós adjacentes  $v_l$  e  $v_{l+1}$ , onde a nova sequência  $T$  é viável. Se o resultado possui um custo melhor, então  $T$  é considerado como o novo roteiro e repete-se até que não haja mais melhorias. Ascheuer *et al.* (2001) restringem o tamanho de sub-caminhos para 5 nós. Mais informações sobre esta heurística podem ser encontradas em Or (1976) e Savelsbergh (1985).
- Heurísticas de melhoria multirotas: São melhorias aplicadas em diferentes rotas com diferentes veículos. Na literatura, há várias abordagens para este tipo de melhoria. Thompson e Psaraftis (1993) descrevem um método chamado "*b-cyclic, k-transfer*", onde, em uma permutação circular de  $b$  roteiros,  $k$  clientes de cada rota são transferidos para a próxima rota da permutação circular. Breedam (1994) classificou algumas operações de melhoria multi-rota, e mostrou a análise computacional sobre um número restrito de problemas. As

figuras 4, 5 e 6 de Toth e Vigo (2002) mostram o funcionamento de algumas melhorias multirotas.

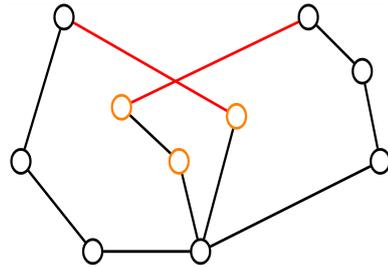
- *String Cross (SC)*, ou Cruzamento de *String*: Duas *strings* (ou cadeias) de vértices são trocadas pelo cruzamento de dois arcos de duas diferentes rotas. A Figura 4 explica seu funcionamento.

Figura 4 – Funcionamento do SC

(a) *String* de vértices selecionadas



(b) *String* de vértices trocadas entre roteiros

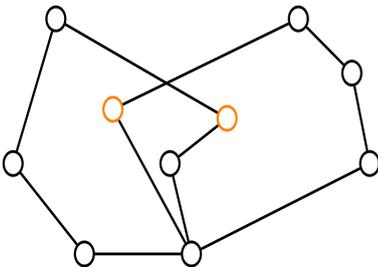


Fonte: Toth e Vigo (2002).

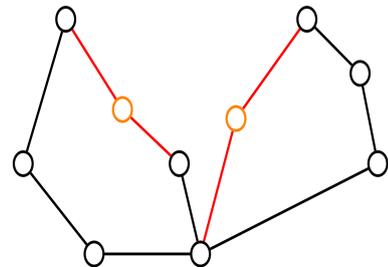
- *String Exchange (SE)*, ou Troca de *Strings*: Duas *strings* de  $k$  vértices são trocadas entre roteiros diferentes. A Figura 5 explica seu funcionamento.

Figura 5 – Funcionamento do SE

(a) *String* de vértices selecionadas



(b) *String* de vértices trocadas entre roteiros



Fonte: Toth e Vigo (2002).

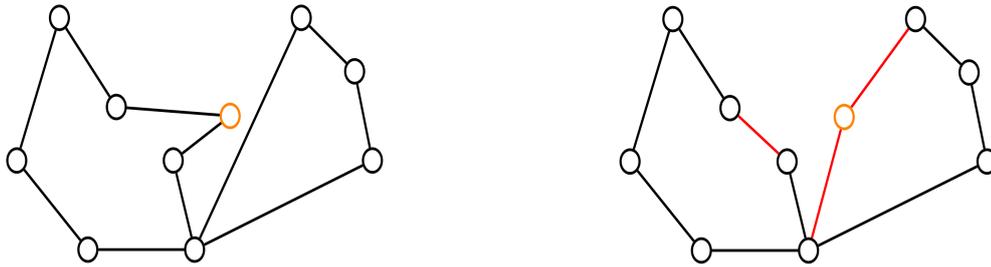
- *String Relocation (SR)*, ou Realocação de *Strings*: Uma *string* de  $k$  vértices é movida de uma rota para outra. A Figura 6 explica seu funcionamento.
- *String Mix (SM)*, ou Mistura de *Strings*: A melhor troca entre SE e SR é selecionada.

Silva *et al.* (2015) apresentam algumas buscas locais para VRP, dentre elas, encontra-se uma busca local chamada *Swap (2, 1)*, que troca dois nós adjacentes  $i$  e  $j$  de um roteiro  $r_1$  por um nó  $k$  de outro roteiro  $r_2$ , ao inserir os nós  $i$  e  $j$  em  $r_2$ , pode ser feita na ordem  $(i, j)$  ou  $(j, i)$ .

Hertz e Mittaz (2001) definem busca local como procedimento iterativo que tem o objetivo de encontrar uma solução  $s$  minimizando a função objetivo  $f$  sobre um conjunto  $S$  de todas as soluções viáveis. Hertz e Mittaz (2001) dizem que o processo iterativo começa de uma

Figura 6 – Funcionamento do SR

(a) Vértices para serem movidos foram selecionados (b) Vértices são movidos



Fonte: Toth e Vigo (2002).

solução inicial em  $s \in S$ , e que a próxima solução é escolhida na vizinhança  $N(S) \subseteq S$ , onde  $N(S)$  é o conjunto de soluções possíveis de se obter com a busca local. Desse modo, um vizinho  $s'$  em  $N(S)$  é obtido pela aplicação da busca local a partir de  $s$ .

Já Tarantilis e Kiranoudis (2007) usaram o algoritmo de *Adaptative Memory* (AM), ou Memória Adaptativa, junto com *Generalized Route Construction Algorithm* (GEROCA), ou Algoritmo de Construção de Rota Generalizada, para o HFFVRP em dois casos reais, um no setor de laticínios e o outro no setor de construção. Os autores conseguiram reduzir o custo total de distância da distribuição e a quantidade de veículos usados.

### 2.3 Meta-heurísticas relacionadas

Meta-heurísticas são uma classe de métodos aproximados que tem sido desenvolvida desde o início de 1980, e são projetadas para atacar a complexidade dos problemas de otimização onde heurísticas clássicas e métodos de otimização têm falhado em ser efetivas e eficientes. A meta-heurística é formalmente definida como um processo de geração iterativa que guia uma heurística subordinada pela combinação inteligente de diferentes conceitos para explorar o espaço de busca, aprendendo estratégias que são usadas para estruturar informação visando encontrar eficientemente soluções próximas da ótima (LAPORTE; OSMAN, 1995 apud OSMAN; LAPORTE, 1996) e (OSMAN; KELLY, 1996 apud OSMAN; LAPORTE, 1996).

- Baseados em TS: A *Tabu Search* (TS), ou Busca Tabu, inicia de uma solução inicial  $x_1$ , e para cada iteração  $t$  de  $x_t$ , move-se para uma solução melhor  $x_{t+1}$ , até que o critério de parada é satisfeito. Se  $f(x)$  denota o custo de  $x$ , então  $f(x_{t+1})$  não necessariamente é menor que  $f(x_t)$ . Alguns autores aplicaram TS no VRP em seus trabalhos, como Gendreau *et al.* (1994), Taillard (1993), Xu e Kelly (1996), Wassan e Osman (2002) e Lee *et al.* (2008).

Gendreau *et al.* (1994) aplicam a TS em sua heurística *TABUROUTE* para o VRP. O algoritmo *TABUROUTE* considera uma sequência de soluções adjacentes obtidas pela repetida remoção de um vértice de uma rota e a reinserção desse mesmo vértice em outra rota. Durante a execução do algoritmo *TABUROUTE*, soluções inviáveis são permitidas. Já Taillard (1993) apresenta dois métodos de partição que aceleram métodos de busca iterativa aplicados ao VRP incluindo problemas com grande número de veículos, utilizando uma simples implementação da TS como um método de busca iterativa.

Um algoritmo HVRP foi desenvolvido por Gendreau *et al.* (1999), o algoritmo utiliza o método de construção *GENI (Generalized Insertion Procedure) and US (Unstringing e Stringing)* (GENIUS) de Gendreau *et al.* (1992), ou Inserção Generalizada e Amarrar e Desamarrar, com uma abordagem da meta-heurística TS e um procedimento AM.

Xu e Kelly (1996) desenvolveram uma nova abordagem de busca local baseada em um modelo de rede de fluxo que é usado para avaliar movimento de remoção e inserção de vários cliente simultaneamente, essa abordagem junto com um procedimento de troca de cliente direto é usado para o VRP. A TS é incorporada no procedimento de Xu e Kelly (1996) para superar o ótimo local, e estratégias de intensificação e diversificação são desenvolvidas para fornecer aprimoramentos efetivos para o básico algoritmo de TS.

Wassan e Osman (2002) desenvolveram novas variantes da meta-heurística TS para o *Mix Fleet VRP* (MFVRP), ou Problema de Roteirização de Veículos com Frota Mista, as variantes usam uma mistura de diferentes componentes, incluindo conceitos de TS reativa, vizinhanças variáveis, estruturas especiais de memória de dados e funções *hash*. O conceito reativo de Wassan e Osman (2002), é usado em uma nova abordagem para desencadear o interruptor entre simples movimentos para intensificação e movimentos mais complexos para diversificação da estratégia de busca. As estruturas de dados especiais de Wassan e Osman (2002), são novamente introduzidas para busca eficiente de vários espaços de vizinhança. A combinação de estrutura de dados e estratégia de balanceamento de Wassan e Osman (2002) entre intensificação e diversificação gera uma implementação robusta e eficiente, a qual é muito competitiva com outros algoritmos da literatura sobre um conjunto de instâncias para alguns resultados já conhecidos.

Lee *et al.* (2008) desenvolvem uma heurística eficiente para determinar a composição de uma frota de veículos e de rotas de viagem usando a TS e um conjunto de problemas de particionamento, na heurística soluções iniciais são encontradas usando um método

modificado de varredura, e sempre quando uma nova solução encontrada em uma iteração da TS é obtida, a alocação ótima de veículos é realizada para o conjunto de rotas, que são construídas a partir da solução atual pela criação de um roteiro gigante.

- VNS: Mladenović e Hansen (1997) definem o *Variable Neighborhood Search* (VNS) como uma meta-heurística simples e efetiva obtida por um procedimento de uma mudança sistemática da vizinhança com um algoritmo de busca local. Ao contrário da maioria das buscas locais, o VNS não segue uma trajetória, mas explora as vizinhanças mais distantes da solução atual, e toma a vizinhança como nova solução atual, se uma melhoria foi feita na solução atual. Definindo:
  - $N_k, (k = 1, \dots, k_{max})$  um conjunto finito de estruturas de vizinhança pré-selecionadas.
  - $N_k(x)$  o conjunto de soluções da  $k^{th}$  vizinhança.

Heurísticas de buscas locais usualmente usam apenas uma estrutura de vizinhança, ou seja,  $k_{max} = 1$ . Quando usada mais de uma, as seguintes perguntas devem ser respondidas.

1. Qual  $N_k$  deve ser usado e quantos deles?
2. Qual deve ser sua ordem na busca?
3. Qual estratégia de busca deve ser usada em mudança de vizinhança?

Uma resposta direta para essas questões é dada pelas regras do algoritmo VNS básico.

*Inicialização.*

1. Selecione um conjunto de estruturas de vizinhança  $N_k$ , que será usado para a busca;
2. Encontre a solução inicial  $x$ ;

*Passo principal.*

1. Conjunto  $k := 1$ ;
2. Até que  $k = k_{max}$  repita os seguintes passos:
  - a) Selecione uma solução  $x'$  aleatoriamente a partir de  $N_k(x)$ , ou seja  $x' \in N_k(x)$ ;
  - b) Aplique algum método de busca local em  $x'$  como solução. E  $x''$  será o local ótimo obtido por esse método de busca local para a solução  $x'$ ;
  - c) Se a solução obtida  $x''$  é melhor do que a atual, então  $x := x''$  e continua a busca com  $N_1(k := 1)$ , se não  $k := k + 1$ ;

O passo principal pode ser iterado com algum outro critério de parada, como por exemplo, número máximo de iterações, tempo máximo de *Central Process Unit* (CPU), ou Unidade de Processamento Central, permitido, ou tempo máximo de interação entre duas melhorias. O VNS básico é de fato um método de descida. Pois como mostrado no Passo 2c do

algoritmo, a solução encontrada só será mantida se tiver o seu custo menor que o da solução inicial. Assim, o algoritmo irá sempre buscará a solução com menor custo. Sem muito esforço adicional, o algoritmo pode ser transformado em um método de descida e subida, bastando alterar o passo 2b para: Se a solução obtida  $x''$  é melhor do que a atual, então  $x := x''$  e continua a busca com  $N_1(k := 1)$ , se não  $x := x'$ .

Subramanian *et al.* (2010) aplicaram o *Variable Neighborhood Descent* (VND), ou Descida em Vizinhança Variável, com uma abordagem paralela para resolver o *VRP with Simultaneous Pickup and Delivery* (VRPSPD), ou VRP com coleta e entrega simultânea. O seu algoritmo paralelo utiliza uma heurística multi-início, que consiste em uma VND integrada em um *framework Iterated Local Search* (ILS), ou Busca Local Iterada, obtendo bons resultados em comparação a outros resultados da literatura.

Na literatura, há uma grande variedade de soluções para o HVRP e suas variações. Bula *et al.* (2017) realizaram uma implementação da versão geral do *Variable Neighborhood Search* (VNS) para resolver o problema de HFVRP em transporte de materiais perigosos, integrando o algoritmo *Variable Neighborhood Descent* (VND) para busca local, um mecanismo de perturbação (sacudindo a vizinhança) e um procedimento de pós-otimização, que é aplicado para melhorar a qualidade da solução. O algoritmo foi executado para instâncias de até 100 nós.

- Baseado em algoritmo genético:

Guezouli e Abdelhamid (2017) propuseram um algoritmo baseado em algoritmo genético para resolver o problema de *Multi-Depot Fleet Size Mix Vehicle Routing Problem with Time Window* (MD-FSMVRP-TW) ou Problema de Roteirização de Veículos com Multi Depósitos com Frota Heterogênea e Janela de Tempo, e executou testes com instâncias de até 100 nós.

Lima *et al.* (2004) utilizaram um algoritmo genético com duas buscas locais, o algoritmo de GENIUS, que é uma busca local dividida nas duas seguintes fases: Procedimento de Inserção Generalizada e Amarrar e Desamarrar, para mais detalhes consultar (GENDREAU *et al.*, 1992), e o  $\lambda$ -interchange, para mais detalhes consultar Wassan e Osman (2002).

Liu (2013) propôs um algoritmo genético baseado na heurística de população para o HVRPD e para o HVRPFD, envolvendo buscas locais. O algoritmo foi executado com 8 instâncias de Golden *et al.* (1984), com 50, 75 e 100 nós, considerando ambos os casos HVRPD e HVRPFD para cada instância. Liu *et al.* (2009) projetaram e implementou um

algoritmo genético, comparando com vinte instâncias da literatura.

- Métodos exatos:

O primeiro algoritmo sofisticado de *branch and cut* baseado na formulação de dois índices para o CVRP foi proposto por Augerat *et al.* (1998) (BALDACCI *et al.*, 2010).

Augerat *et al.* (1998) apresentam um algoritmo de *branch and cut* para resolver o CVRP de forma ótima, o resultado do algoritmo de plano de corte foi aplicado em um conjunto de instâncias da literatura, onde os limites inferiores obtidos são melhores que os conhecidos previamente, e algumas estratégias de ramificação foram implementadas para desenvolver um algoritmo de *branch and cut* que foi capaz de resolver instâncias CVRP grandes, sendo que algumas dessas instâncias ainda não tinham sido resolvidas.

Outros métodos de *branch and cut* baseados na formulação de dois índices foram propostas por Ralphs *et al.* (2003) e Lysgaard *et al.* (2004) (BALDACCI *et al.*, 2010). Ralphs *et al.* (2003) descrevem uma metodologia de separação baseada em decomposição para as restrições de capacidade, que obtêm vantagem da habilidade de resolver pequenas instâncias TSP eficientemente. Lysgaard *et al.* (2004) apresentam um novo algoritmo de *branch and cut* para o CVRP, que usa uma variedade de planos de corte e apresenta resultados computacionais para um grande número de instâncias, mostrando que o novo algoritmo é competitivo, em particular, por resolver três instâncias de Augerat (1995) de forma ótima pela primeira vez.

Baldacci *et al.* (2004) descrevem uma nova formulação de programação inteira para o CVRP baseado em uma abordagem de rede de fluxo, descreve um novo algoritmo de *branch and cut* para solução ótima do CVRP, apresenta um limite inferior derivado do relaxamento de programação linear de uma nova formulação, apresenta uma comparação entre o novo limite inferior e os limites inferiores derivados de relaxamentos de programação linear de diferentes formulações CVRP propostas na literatura, com resultados computacionais em um conjunto de problemas teste derivados da literatura e de novos problemas gerados aleatoriamente.

Gendreau *et al.* (1999) afirmam que, até onde está ciente, não há algoritmo exato desenvolvido para o HVRP, mas muitos algoritmos aproximados foram desenvolvidos. Devido à complexidade do problema, não há algoritmo exato conhecido para este problema (LIMA *et al.*, 2004). Porém Baldacci e Mingozzi (2008) apresentam um método exato unificado para resolver o HVRP baseado na formulação de particionamento de conjunto.

### 3 ALGORITMO PROPOSTO

O algoritmo que será descrito nesse trabalho utiliza a meta-heurística VND para melhoria de roteiro, com buscas locais intra-roteiro e inter-roteiro e inserção de clientes de forma aleatória. O algoritmo considera que há um número ilimitado de veículos de cada tipo disponível.

O conjunto  $S$  de rotas apresentado como saída do algoritmo é definido como  $S = \{(u, SR) : u \in M, SR = (0, a, \dots, 0), a \in V'\}$ .

O Algoritmo 1 apresenta o passo a passo para a construção de roteiros.

---

#### **Algoritmo 1:** Construção de roteiros

---

**Entrada :**  $V$  um conjunto de nós,  $M$  um conjunto dos tipos de veículos e  $c_{ij} : i, j \in V$  a relação de custo entre nós

**Saída :** Conjunto  $S$  de rotas

```

1  $S \leftarrow \emptyset$ ;
2 for  $i \in V'$  do
3    $v \leftarrow$  um veículo de  $u \in M : Q(u) \geq q(i)$ ;
4    $coloque(S, (v, (0, i, 0)))$ ;
5    $S \leftarrow VND(S, M, IVnd, EVnd)$ ;
6 end
7 return  $S$ ;
```

---

Na linha 1, define-se o conjunto solução  $S$  como  $\emptyset$ . Na linha 2, são percorridos todos os clientes da entrada. Na linha 3, seleciona-se um veículo de um tipo de veículo de  $M$ , tal que esse veículo tenha capacidade de atender o cliente  $i$ . Na linha 4, será adicionada no conjunto solução  $S$  uma dupla relacionando um veículo e os clientes que são atendidos por esse veículo. Na linha 5 é executado o Algoritmo 2.

O procedimento VND descrito nesse trabalho utiliza as buscas locais na ordem apresentada:

- Intra-roteiro:
  1. *Exchange*
  2. Or-Opt
  3. 2-Opt
  4. 3-Opt
- Inter-roteiro:
  1. *String Cross (SC)*

2. *String Relocation* (SR) com 1-vértices, ou *Shift* (0, 1)
3. *String Relocation* (SR) com 2-vértices, ou *Shift* (0, 2)
4. *String Exchange* (SE) com 1-vértices, ou *Swap* (1, 1)
5. *Shift* (2, 1)
6. *3-cyclic 2-transfer*

As buscas locais produzem soluções factíveis. A sequência  $IVnd$  irá conter as buscas locais intra-roteiro e a sequência  $EVnd$  irá conter as buscas locais inter-roteiro, tal que  $EVnd, IVnd = (R_1, \dots, R_{max})$ . O Algoritmo 2 mostra o funcionamento do VND utilizado nesse trabalho.

---

**Algoritmo 2:** *Variable Neighborhood Descent* (VND)

---

**Entrada :**  $S$  um conjunto de rotas,  $M$  o conjunto de tipos de veículos,  $I$  uma sequência de buscas locais intra-roteiro e  $E$  uma sequência de buscas locais inter-roteiro

**Saída :** Conjunto  $S$  de rotas

```

1  $\beta \leftarrow 10;$ 
2 while true do
3    $X \leftarrow S;$ 
4   Aplica Buscas Locais( $I, S$ );
5   Usar Mais Veículos( $S, M, \beta$ );
6   Aplica Buscas Locais( $E, S$ );
7   if Custo de  $S = \text{custo de } X$  then
8     return  $S;$ 
9   end
10 end

```

---

Na linha 1, é definida uma variável  $\beta$  com o valor 10, essa variável será usada no Algoritmo 4. Na linha 3, é feita uma cópia da solução  $S$  em  $X$ . Nas linhas 4 e 6, executa-se o Algoritmo 3 para a sequência de buscas locais intra-roteiro  $I$  e inter-roteiro  $E$ . Na linha 5, é executado o Algoritmo 4. Na linha 7, é verificado se o custo da solução obtida com as linhas 3, 4 e 5 permanece igual, ou seja não houve melhorias, se sim o algoritmo encerra.

No Algoritmo 2, pode-se inserir outra condição na linha 2 para evitar um espera indeterminada, por exemplo número máximo de iterações.

A aplicação de buscas locais na solução é feita pelo Algoritmo 3.

---

**Algoritmo 3:** Aplica Buscas Locais

---

**Entrada :**  $R$  uma sequência de buscas locais e  $S$  um conjunto de rotas

**Saída :** Conjunto  $S$  de rotas

```

1 for  $i \leftarrow 1, \dots, max$  do
2    $S' \leftarrow$  Aplicação de  $R_i$  em  $S$ ;
3   if Custo de  $S'$  < custo de  $S$  then
4      $S \leftarrow S'$ ;
5      $i \leftarrow 0$ ;
6   end
7 end
8 return  $S$ ;

```

---

O Algoritmo 3, aplica uma sequência de buscas locais sobre um conjunto de rotas, e para cada aplicação verifica se há uma melhoria com a aplicação da busca local, se sim, então a melhoria é mantida e a aplicação de busca local retorna para a primeira busca local, se não, então continua com a próxima busca local da sequência.

Na linha 1, percorre-se todas as buscas locais da sequência  $R$ . Na linha 2, é aplicado a busca local  $R_i$  na entrada  $S$ . Na linha 3, verifica-se se o custo obtido com a aplicação da busca local obteve melhor resultado, se sim,  $S$  recebe  $S'$  e serão percorrida todas as buscas locais novamente.

No Algoritmo 3, igualmente ao Algoritmo 2, pode-se inserir outra condição nas linhas 1 e/ou 3 para evitar uma espera indeterminada.

O procedimento de usar mais veículos é descrito no Algoritmo 4.

---

**Algoritmo 4:** Usar Mais Veículos

---

**Entrada :** Conjunto  $S$  de rotas, conjunto  $M$  de tipos de veículos e  $\beta$  a quantidade de veículos

**Saída :** Conjunto  $S$  de rotas

```

1 for  $u \in M$  do
2   | for  $k \leftarrow 1, \dots, \beta$  do
3   |   |  $v \leftarrow$  veículo do tipo  $u$ ;
4   |   |  $coloque(S, (v, (0,0)))$ ;
5   |   | end
6 end
7 return  $S$ ;

```

---

Esse procedimento, tem o objetivo de inserir no conjunto  $S$ ,  $\beta$  veículos de cada tipo em roteiros sem clientes para serem atendidos. Isso é feito para que as buscas locais Inter-roteiro adicionem clientes de outros roteiros nesses roteiros sem clientes.

Na linha 1, é percorrido de 1 até fator  $\beta$ , que representa a quantidade de veículos de cada tipo que serão adicionados no conjunto  $S$ . Na linha 2, percorre-se todos os tipos de veículos. Na linha 3, é obtido um veículo  $v$  do tipo  $u$ . Na linha 4, executa-se a função *coloque*, que insere no conjunto  $S$  a dupla com o veículo  $v$  com uma rota apenas com o depósito como origem e destino.

Outro algoritmo também foi implementado neste trabalho, esse algoritmo utiliza uma abordagem ascendente do VNS. Esse procedimento é descrito no Algoritmo 5.

---

**Algoritmo 5:** VNS Ascendente

---

**Entrada :**  $V$  um conjunto de vértices,  $M$  um conjunto dos tipos de veículos e  $c_{ij} : i, j \in V$  a relação de custo entre nós

**Saída :** Conjunto  $S$  de rotas

```

1  $T \leftarrow 5$ ;
2  $S \leftarrow$  Clarke e Wright HVRP( $V, M$ );
3  $S \leftarrow$  VND Pre Climb ( $S, M$ );
4  $S' \leftarrow$  Climb ( $S, M$ );
5  $S' \leftarrow$  VND Pos Climb ( $S', M$ );
6 if Custo de  $S'$  < Custo de  $S$  then
7   | return  $S'$ ;
8 end
9 return  $S$ ;

```

---

O Algoritmo 5 utiliza o algoritmo de Clarke e Wright adaptado para o HVRP para construção de rotas, um método VND com poucas buscas locais, com a intenção de encontrar o ótimo local, um método de subida baseado na meta-heurística VNS, com a intenção de explorar mais vizinhanças, e um método VND com mais buscas locais, com a intenção de encontrar o ótimo global. Porém essa abordagem não gerou resultados melhores do que o algoritmo baseado em VND. O Algoritmo 5 mostra o funcionamento.

Na linha 1, define-se o termo  $T$  que será usado no Algoritmo 8. Na linha 2, define-se o conjunto  $S$  de rotas geradas pelo Algoritmo 6. Na linha 3, atribui-se ao conjunto  $S$  o retorno do Algoritmo 7. Na linha 4, define-se o conjunto  $S'$  como retorno do Algoritmo 8. Na linha 5, atribui-se ao conjunto  $S'$  o retorno do Algoritmo 9. Na linha 6, verifica-se se o custo obtido com a aplicação dos Algoritmos 8 e 9 obteve melhor resultado, se sim, retorna  $S'$ , se não, então retorna  $S$ .

O procedimento Clarke e Wright HVRP é descrito no Algoritmo 6.

---

**Algoritmo 6:** Clarke e Wright HVRP

---

**Entrada :**  $V$  um conjunto de nós e conjunto  $M$  de tipos de veículos

**Saída :** Conjunto  $S$  de rotas

```

1  $S \leftarrow \emptyset$ ;
2  $M' \leftarrow$  Ordenação de  $M$ ;
3 for  $u \in M'$  do
4    $S' \leftarrow$  Construir Pior Caso ( $V, u, M'$ );
5    $G \leftarrow$  Calcular Ganhos ( $S'$ );
6    $S' \leftarrow$  Mesclas ( $S', G$ );
7   if Custo de  $S' <$  Custo de  $S$  OR  $S = \emptyset$  then
8      $S \leftarrow S'$ ;
9   end
10 end
11 return  $S$ ;

```

---

O Algoritmo 6 é uma adaptação do algoritmo de Clarke e Wright para a construção de rotas HVRP. O algoritmo busca escolher qual o melhor tipo de veículo para roteirizar usando a lógica original do algoritmo de Clarke e Wright proposto por Clarke e Wright (1964).

Na linha 1, define-se o conjunto  $S$  como  $\emptyset$ . Na linha 2, define-se  $M'$  como o conjunto  $M$  ordenado pela capacidade dos veículos de forma crescente. Na linha 3, percorre-se todos os tipos de veículos ordenados pela capacidade de forma crescente. Na linha 4, define-se  $S'$  como o retorno do Algoritmo 10 para o tipo de veículo  $u$ . Na linha 5, define-se  $G$  como o retorno do procedimento de cálculo de ganhos do algoritmo de Clarke e Wright. Na linha 6, atribui-se a variável  $S'$  o retorno do procedimento de mesclas do algoritmo de Clarke e Wright. Na linha 7, verifica-se se o custo de  $S'$  é menor do que o custo de  $S$ , ou se o conjunto  $S$  é vazio, se sim, então atribui-se a  $S$  o conjunto  $S'$ . Na linha 11, retorna-se o conjunto  $S$ .

O procedimento VND Pre Climb é descrito no Algoritmo 7.

---

**Algoritmo 7:** VND Pre Climb

---

**Entrada :** Conjunto  $S$  de rotas, conjunto  $M$  de tipos de veículos

**Saída :** Conjunto  $S$  de rotas

```

1 return  $VND(S, M, IRPre, ERPre)$ ;

```

---

O procedimento VND Pre Climb descrito nesse trabalho utiliza as buscas locais na ordem apresentada:

- Intra-roteiro:
  1. *Exchange*
  2. Or-Opt
- Inter-roteiro:
  1. *String Relocation* (SR) com 1-vértices, ou *Shift* (0, 1)
  2. *String Exchange* (SE) com 1-vértices, ou *Swap* (1, 1)

As buscas locais produzem soluções factíveis. A sequência  $I\text{Pre}$  irá conter as buscas locais intra-roteiro e a sequência  $E\text{Pre}$  irá conter as buscas locais inter-roteiro, tal que  $E\text{Pre}, I\text{Pre} = (R_1, \dots, R_{max})$ .

Na linha 1, retorna-se o procedimento VND com as sequências  $I\text{Pre}$  e  $E\text{Pre}$ .

O procedimento Climb é descrito no Algoritmo 8.

---

#### **Algoritmo 8:** Climb

---

**Entrada :** Conjunto  $S$  de rotas, conjunto  $M$  de tipos de veículos e número  $T$  o limite de vezes que uma solução pode ser piorada

**Saída :** Conjunto  $S$  de rotas

```

1 for  $t \leftarrow 1, \dots, T$  do
2   | for  $i \leftarrow 1, \dots, max$  do
3   |   |  $S \leftarrow$  Aplicação de  $Rc_i$  em  $S$ ;
4   |   end
5 end
6 return  $S$ ;

```

---

O procedimento Climb realiza uma degradação da solução  $S$  aumentando o seu custo, porém gerando soluções factíveis. O procedimento utiliza adaptações de algumas buscas locais para degradar a solução, as buscas locais retornam a primeira pior solução que encontram. As buscas locais adaptadas estão na ordem apresentada:

1. Primeiro Pior Or-Opt
2. Primeiro Pior *String Cross* (SC)
3. Primeiro Pior *Two-Node-Exchange*
4. Primeiro Pior *Shift* (2, 1)
5. Primeiro Pior *Arc-Reversal*
6. Primeiro Pior *String Relocation* (SR) com 2-vértices, ou *Shift* (0, 2)
7. Primeiro Pior *Node-Reinsertion*
8. Primeiro Pior *String Exchange* (SE) com 1-vértices, ou *Swap* (1, 1)

As adaptações das buscas locais produzem soluções factíveis. A sequência *BCLimb* irá conter as adaptações das buscas locais apresentadas, tal que  $BCLimb = (Rc_1, \dots, Rc_{max})$ .

Na linha 1, percorre-se todos os números de 1 até  $T$ . Na linha 2, percorre-se as buscas locais adaptadas todas as buscas locais de *BCLimb*. Na linha 3, aplica-se a busca local adaptada  $Rc_i$  no conjunto solução  $S$ . E na linha 6, retorna-se o conjunto  $S$ .

O procedimento VND Pos Climb é descrito no Algoritmo 9.

---

**Algoritmo 9:** VND Pos Climb

---

**Entrada :** Conjunto  $S$  de rotas, conjunto  $M$  de tipos de veículos

**Saída :** Conjunto  $S$  de rotas

1 **return**  $VND(S, M, IPos, EPos)$ ;

---

O procedimento VND Pos Climb é semelhante ao Pre Climb, a diferença é que o Pos Climb utiliza diferentes buscas locais para tentar alcançar o global ótimo. O procedimento VND Pos Climb descrito nesse trabalho utiliza as buscas locais na ordem apresentada:

- Intra-roteiro:
  1. *Swap*
  2. *Node-Reinsertion*
  3. *Arc-Reversal*
  4. *Arc-Reinsertion*
  5. *Two-Node-Exchange*
  6. Or-Opt
- Inter-roteiro:
  1. *String Relocation* (SR) com 2-vértices, ou *Shift* (0, 2)
  2. *Shift* (2, 1)
  3. *String Cross* (SC)
  4. *3-cyclic 2-transfer*

As buscas locais produzem soluções factíveis. A sequência *IPos* irá conter as buscas locais intra-roteiro e a sequência *EPos* irá conter as buscas locais inter-roteiro, tal que  $EPos, IPos = (R_1, \dots, R_{max})$ .

O procedimento Construir Pior Caso é descrito no Algoritmo 10.

---

**Algoritmo 10:** Construir Pior Caso

---

**Entrada :** Conjunto  $V$  de vértices, tipo  $u$  de veículo e sequência  $M'$  de tipos de veículos ordenados de forma crescente pela capacidade

**Saída :** Conjunto  $S$  de rotas

```

1  $S \leftarrow \emptyset$ ;
2 for  $i \in V'$  do
3    $v \leftarrow u$ ;
4   while  $q_i > Q_v$  do
5      $v \leftarrow$  Veículo após  $v$  na sequência  $M'$ ;
6   end
7    $coloque(S, (v, (0, i, 0)))$ ;
8 end

```

---

O Algoritmo 10 constrói a pior rota possível usando a lógica original do algoritmo de Clarke e Wright. O algoritmo tenta usar o veículo do tipo  $u$  em todos os clientes, porém, quando o veículo do tipo  $u$  não pode atender o cliente  $i$  por não ter capacidade de atender a demanda do cliente, então é selecionado o tipo de veículo após o tipo  $u$  na sequência  $M'$ .

Na linha 1, define-se o conjunto  $S$  como  $\emptyset$ . Na linha 2, percorre-se todos os nós de  $V'$ . Na linha 3, define-se a variável  $v$  como o tipo de veículo  $u$ . Na linha 4, verifica-se se o cliente  $i$  não pode ser atendido pelo tipo de veículo  $u$ , se sim, então a variável  $v$  recebe o próximo tipo de veículo da sequência de tipos  $M'$ . Na linha 7, adiciona-se no conjunto  $S$  uma rota apenas com um cliente, o cliente  $i$ .

## 4 RESULTADOS

O algoritmo VND proposto e o algoritmo VNS Ascendente foram aplicados em oito instâncias HVRP citadas no trabalho de Gendreau *et al.* (1999), que contém 50, 75 e 100 clientes. Cada instância possui veículos com capacidades e custos variáveis. Os custos que serão apresentados na Tabela 1 são calculados multiplicando o custo variável do veículo pela distância percorrida pelo veículo. As instâncias utilizam distâncias euclidianas e satisfazem a desigualdade triangular.

Os algoritmos VND e VNS Ascendente foram executados em um computador *Desktop* com 3,5 GB de memória *RAM*, processador *Intel® Celeron(R) CPU N3060 @ 1.60GHz 2*, sistema operacional *Ubuntu 16.04 LTS 64 bits* e 103,5 GB de HD.

Para o *benchmarking* foram escolhidos os resultados dos algoritmos de Gendreau *et al.* (1999), Wassan e Osman (2002) e Lee *et al.* (2008), pois os trabalhos citados apresentam resultados de algoritmos HVRP com as instâncias de Gendreau *et al.* (1999).

A Tabela 1 apresenta os resultados do algoritmo VND proposto neste trabalho, Gendreau *et al.* (1999), Wassan e Osman (2002) e Lee *et al.* (2008). A coluna instância apresenta a numeração da instância utilizada por Gendreau *et al.* (1999), a coluna *n* diz a quantidade de clientes da instância, a coluna custo representa o custo da solução obtida e a coluna tempo, informa o tempo em segundos que o algoritmo levou para executar.

Tabela 1 – Resultado dos algoritmos

Instância	<i>n</i>	Gendreau <i>et al.</i> (1999)		Wassan e Osman (2002)		Lee <i>et al.</i> (2008)		Algoritmo VND	
		Custo	Tempo (s)	Custo	Tempo (s)	Custo	Tempo (s)	Custo	Tempo (s)
13	50	1491	626	1499	140	<b>1491</b>	142	1558	26
14	50	603	669	608	575	<b>603</b>	144	643	15
15	50	999	736	<b>999</b>	289	<b>999</b>	162	1086	13
16	50	1136	852	<b>1131</b>	246	<b>1131</b>	144	1209	17
17	75	1031	1453	1047	2156	1038	865	1102	94
18	75	1801	1487	1814	2469	<b>1800</b>	545	1955	93
19	100	1105	1681	<b>1100</b>	2553	<b>1105</b>	331	1194	374
20	100	1541	1706	<b>1530</b>	2910	<b>1531</b>	314	<b>1530</b>	305

Fonte: Elaborada pelo autor.

A Tabela 2 apresenta os resultados do algoritmo VND e do VNS Ascendente com as mesmas instâncias de Gendreau *et al.* (1999).

Tabela 2 – Resultado dos algoritmos VND e VNS Ascendente

Instância	$n$	Algoritmo VND		Algoritmo VNS Ascendente	
		Custo	Tempo (s)	Custo	Tempo (s)
13	50	1558	26	1581	3
14	50	643	15	<b>640</b>	0.3
15	50	1086	13	<b>1070</b>	0.2
16	50	1209	17	1231	0.2
17	75	1102	94	1127	1
18	75	1955	93	2036	4
19	100	1194	374	1237	1
20	100	<b>1530</b>	305	1708	2

Fonte: Elaborada pelo autor.

## 5 CONCLUSÕES E TRABALHOS FUTUROS

### 5.1 Conclusões

O algoritmo VND proposto apresentou um resultado melhor para uma das instâncias em comparação ao algoritmo de Gendreau *et al.* (1999) e Lee *et al.* (2008) e um resultado igual em comparação ao algoritmo de Wassan e Osman (2002), e os demais resultados próximos aos deles. Por mais que o algoritmo VND não tenha obtido resultados melhores para a maioria das instâncias, o algoritmo obteve bons resultados. Pois para todas as instâncias o algoritmo VND não apresentou um custo maior que 9% em comparação aos resultados de Gendreau *et al.* (1999), Wassan e Osman (2002) e Lee *et al.* (2008).

O algoritmo VNS Ascendente proposto apresentou bons resultados apenas para duas instâncias em relação aos resultados do algoritmo VND proposto. Porém para todas as outras instâncias, o algoritmo VNS Ascendente não apresentou bons resultados.

### 5.2 Trabalhos futuros

Como trabalhos futuros, pode ser feita uma proposta de um algoritmo para o *VRP with Time Windows* (VRPTW), ou VRP com Janela de Tempo. Para tanto, pode-se utilizar uma variação do próprio VND ou mesmo de outros métodos, como os observados em outros trabalhos: Busca Tabu, VNS, dentre outros algoritmos heurísticos e meta-heurísticos.

## REFERÊNCIAS

- ASCHEUER, N.; FISCHETTI, M.; GRÖTSCHHEL, M. Solving the asymmetric travelling salesman problem with time windows by branch-and-cut. **Mathematical Programming**, v. 90, n. 3, p. 475–506, May 2001. ISSN 1436-4646. Disponível em: <<https://doi.org/10.1007/PL00011432>>. Acesso em: 29 jun. 2018.
- AUGERAT, P. **Polyhedral approach of the vehicle routing problem**. Tese (Theses) — Institut National Polytechnique de Grenoble - INPG, jun. 1995. Disponível em: <<https://tel.archives-ouvertes.fr/tel-00005026>>. Acesso em: 29 jun. 2018.
- AUGERAT, P.; BELENGUER, J. M.; BENAVENT, E.; CORBERAN, A.; NADDEF, D.; RINALDI, G. **COMPUTATIONAL RESULTS WITH A BRANCH-AND-CUT CODE FOR THE CAPACITATED VEHICLE ROUTING PROBLEM**. Roma: ISTITUTO DI ANALISI DEI SISTEMI ED INFORMATICA, 1998.
- BALDACCI, R.; BATTARRA, M.; VIGO, D. Routing a heterogeneous fleet of vehicles. In: GOLDEN, B.; RAGHAVAN, S.; WASIL, E. (Ed.). **The Vehicle Routing Problem: Latest Advances and New Challenges**. Boston, MA: Springer US, 2008. p. 3–27. ISBN 978-0-387-77778-8. Disponível em: <[https://doi.org/10.1007/978-0-387-77778-8\\_1](https://doi.org/10.1007/978-0-387-77778-8_1)>. Acesso em: 29 jun. 2018.
- BALDACCI, R.; HADJICONSTANTINO, E.; MINGOZZI, A. An exact algorithm for the capacitated vehicle routing problem based on a two-commodity network flow formulation. **Operations Research**, v. 52, n. 5, p. 723–738, 2004. Disponível em: <<https://doi.org/10.1287/opre.1040.0111>>. Acesso em: 29 jun. 2018.
- BALDACCI, R.; MINGOZZI, A. A unified exact method for solving different classes of vehicle routing problems. **Mathematical Programming**, v. 120, n. 2, p. 347, Apr 2008. ISSN 1436-4646. Disponível em: <<https://doi.org/10.1007/s10107-008-0218-9>>. Acesso em: 29 jun. 2018.
- BALDACCI, R.; TOTH, P.; VIGO, D. Exact algorithms for routing problems under vehicle capacity constraints. **Annals of Operations Research**, v. 175, n. 1, p. 213–245, Mar 2010. ISSN 1572-9338. Disponível em: <<https://doi.org/10.1007/s10479-009-0650-0>>. Acesso em: 29 jun. 2018.
- BODIN, L.; GOLDEN, B.; ASSAD, A.; BALL, M. **The state of the art in the routing and scheduling of vehicles and crews**. Maryland, 1981.
- BRAMEL, J.; SIMCHI-LEVI, D. A location based heuristic for general routing problems. **Operations Research**, v. 43, n. 4, p. 649–660, 1995. Disponível em: <<https://doi.org/10.1287/opre.43.4.649>>. Acesso em: 29 jun. 2018.
- BREEDAM, V. **An Analysis of the Behavior of Heuristics for the Vehicle Routing Problem for a selection of problems with Vehicle-related, Customer-related, and Time-related Constraints**. Tese (Doutorado) — University of Antwerp, Antwerpen, 1994.
- BRÄYSY, O.; GENDREAU, M. Vehicle routing problem with time windows, part i: Route construction and local search algorithms. **Transportation Science**, v. 39, n. 1, p. 104–118, 2005. Disponível em: <<http://pubsonline.informs.org/doi/abs/10.1287/trsc.1030.0056>>. Acesso em: 29 jun. 2018.

BULA, G. A.; PRODHON, C.; GONZALEZ, F. A.; AFSAR, H. M.; VELASCO, N. Variable neighborhood search to solve the vehicle routing problem for hazardous materials transportation. **Journal of Hazardous Materials**, v. 324, n. Part B, p. 472 – 480, 2017. ISSN 0304-3894. Disponível em: <<http://www.sciencedirect.com/science/article/pii/S0304389416310196>>. Acesso em: 29 jun. 2018.

CHOI, E.; TCHA, D.-W. A column generation approach to the heterogeneous fleet vehicle routing problem. **Computers & Operations Research**, v. 34, n. 7, p. 2080 – 2095, 2007. ISSN 0305-0548. Disponível em: <<http://www.sciencedirect.com/science/article/pii/S0305054805002650>>. Acesso em: 29 jun. 2018.

CHRISTOFIDES, N. The vehicle routing problem. **Revue française d'automatique, informatique, recherche opérationnelle. Recherche opérationnelle**, EDP Sciences, v. 10, n. V1, p. 55–70, 1976.

CLARKE, G.; WRIGHT, J. W. Scheduling of vehicles from a central depot to a number of delivery points. **Operations Research**, v. 12, n. 4, p. 568–581, 1964. Disponível em: <<https://doi.org/10.1287/opre.12.4.568>>. Acesso em: 29 jun. 2018.

COELHO, V. N.; GRASAS, A.; RAMALHINHO, H.; COELHO, I. M.; SOUZA, M. J.; CRUZ, R. C. An ils-based algorithm to solve a large-scale real heterogeneous fleet vrp with multi-trips and docking constraints. **European Journal of Operational Research**, Elsevier, v. 250, n. 2, p. 367–376, 2016.

DANTZIG, G.; FULKERSON, R.; JOHNSON, S. Solution of a large-scale traveling-salesman problem. **Journal of the Operations Research Society of America**, v. 2, n. 4, p. 393–410, 1954. Disponível em: <<https://doi.org/10.1287/opre.2.4.393>>. Acesso em: 29 jun. 2018.

DANTZIG, G. B.; RAMSER, J. H. The truck dispatching problem. **Management Science**, v. 6, n. 1, p. 80–91, 1959. Disponível em: <<https://doi.org/10.1287/mnsc.6.1.80>>. Acesso em: 29 jun. 2018.

DERBEL, H.; JARBOUI, B.; BHIRI, R. A skewed general variable neighborhood search algorithm with fixed threshold for the heterogeneous fleet vehicle routing problem. **Annals of Operations Research**, Springer US, p. 1–30, Jul 2017. ISSN 1572-9338. Disponível em: <<https://doi.org/10.1007/s10479-017-2576-2>>. Acesso em: 29 jun. 2018.

DIAS, L. **Tempo que passa, maturidade que chega**. 2014. <<http://ultimato.com.br/sites/fatosecorrelatos/2014/05/14/tempo-que-passa-maturidade-que-chega/>>. Acesso em: 29 jun. 2018.

FISHER, M. L.; JAIKUMAR, R. A generalized assignment heuristic for vehicle routing. **Networks**, Wiley Subscription Services, Inc., A Wiley Company, v. 11, n. 2, p. 109–124, 1981. ISSN 1097-0037. Disponível em: <<http://dx.doi.org/10.1002/net.3230110205>>. Acesso em: 29 jun. 2018.

GENDREAU, M.; HERTZ, A.; LAPORTE, G. New insertion and postoptimization procedures for the traveling salesman problem. **Operations Research**, v. 40, n. 6, p. 1086–1094, 1992. Disponível em: <<https://doi.org/10.1287/opre.40.6.1086>>. Acesso em: 29 jun. 2018.

GENDREAU, M.; HERTZ, A.; LAPORTE, G. A tabu search heuristic for the vehicle routing problem. **Management Science**, v. 40, n. 10, p. 1276–1290, 1994. Disponível em: <<https://doi.org/10.1287/mnsc.40.10.1276>>. Acesso em: 29 jun. 2018.

- GENDREAU, M.; LAPORTE, G.; MUSARAGANYI, C.; TAILLARD, É. D. A tabu search heuristic for the heterogeneous fleet vehicle routing problem. **Computers & Operations Research**, Elsevier, v. 26, n. 12, p. 1153–1173, 1999.
- GILLETT, B. E.; JOHNSON, J. G. Multi-terminal vehicle-dispatch algorithm. **Omega**, v. 4, n. 6, p. 711 – 718, 1976. ISSN 0305-0483. Disponível em: <<http://www.sciencedirect.com/science/article/pii/0305048376900979>>. Acesso em: 29 jun. 2018.
- GILLETT, B. E.; MILLER, L. R. A heuristic algorithm for the vehicle-dispatch problem. **Operations Research**, v. 22, n. 2, p. 340–349, 1974. Disponível em: <<https://doi.org/10.1287/opre.22.2.34>>. Acesso em: 29 jun. 2018.
- GOLDEN, B.; ASSAD, A.; LEVY, L.; GHEYSENS, F. The fleet size and mix vehicle routing problem. **Computers & Operations Research**, Elsevier, v. 11, n. 1, p. 49–66, 1984.
- GOLDEN, B. L. **Vehicle routing problems: formulations and heuristic solution techniques**. Cambridge, 1975. Disponível em: <<http://www.dtic.mil/dtic/tr/fulltext/u2/a013639.pdf>>. Acesso em: 29 jun. 2018.
- GOLDEN S. RAGHAVAN, E. W. B. **The vehicle routing problem: latest advances and new challenges**. [S.l.]: Springer, Boston, MA. v. 43. ISSN 1387-666X. ISBN 978-0-387-77778-8.
- GUEZOULI, L.; ABDELHAMID, S. A multi-objective optimization of multi-depot fleet size and mix vehicle routing problem with time window. In: IEEE. **Systems and Control (ICSC), 2017 6th International Conference on**. Batna, 2017. p. 328–333.
- HERTZ, A.; MITTAZ, M. A variable neighborhood descent algorithm for the undirected capacitated arc routing problem. **Transportation Science**, v. 35, n. 4, p. 425–434, 2001. Disponível em: <<https://doi.org/10.1287/trsc.35.4.425.10431>>. Acesso em: 29 jun. 2018.
- KARADIMAS, N. V.; DOUKAS, N.; KOLOKATHI, M.; DEFTERAIOU, G. Routing optimization heuristics algorithms for urban solid waste transportation management. **W. Trans. on Comp.**, World Scientific and Engineering Academy and Society (WSEAS), Stevens Point, Wisconsin, USA, v. 7, n. 12, p. 2022–2031, dez. 2008. ISSN 1109-2750. Disponível em: <<http://dl.acm.org/citation.cfm?id=1486811.1486826>>. Acesso em: 29 jun. 2018.
- KARP, R. M. Probabilistic analysis of partitioning algorithms for the traveling-salesman problem in the plane. **Mathematics of Operations Research**, v. 2, n. 3, p. 209–224, 1977. Disponível em: <<https://doi.org/10.1287/moor.2.3.209>>. Acesso em: 29 jun. 2018.
- LAPORTE, G. The vehicle routing problem: An overview of exact and approximate algorithms. **European journal of operational research**, Elsevier, v. 59, n. 3, p. 345–358, 1992.
- LAPORTE, G.; GENDREAU, M.; POTVIN, J.-Y.; SEMET, F. Classical and modern heuristics for the vehicle routing problem. **International Transactions in Operational Research**, Blackwell Publishing Ltd, v. 7, n. 4-5, p. 285–300, 2000. ISSN 1475-3995. Disponível em: <<http://dx.doi.org/10.1111/j.1475-3995.2000.tb00200.x>>. Acesso em: 29 jun. 2018.
- LAPORTE, G.; OSMAN, I. H. Routing problems: A bibliography. **Annals of Operations Research**, v. 61, n. 1, p. 227–262, Dec 1995. ISSN 1572-9338. Disponível em: <<https://doi.org/10.1007/BF02098290>>. Acesso em: 29 jun. 2018.

- LEE, Y. H.; KIM, J. I.; KANG, K. H.; KIM, K. H. A heuristic for vehicle fleet mix problem using tabu search and set partitioning. **Journal of the Operational Research Society**, v. 59, n. 6, p. 833–841, Jun 2008. ISSN 1476-9360. Disponível em: <<https://doi.org/10.1057/palgrave.jors.2602421>>. Acesso em: 29 jun. 2018.
- LENSTRA, J. K.; KAN, A. H. G. R. Complexity of vehicle routing and scheduling problems. **Networks**, v. 11, n. 2, p. 221–227, 1981. Disponível em: <<https://onlinelibrary.wiley.com/doi/abs/10.1002/net.3230110211>>. Acesso em: 29 jun. 2018.
- LIMA, C.; GOLDBARG, M.; GOLDBARG, E. A memetic algorithm for the heterogeneous fleet vehicle routing problem. **Electronic Notes in Discrete Mathematics**, v. 18, n. Supplement C, p. 171 – 176, 2004. ISSN 1571-0653. Disponível em: <<http://www.sciencedirect.com/science/article/pii/S1571065304010844>>. Acesso em: 29 jun. 2018.
- LIN, S. Computer solutions of the traveling salesman problem. **Bell System Technical Journal**, Blackwell Publishing Ltd, v. 44, n. 10, p. 2245–2269, 1965. ISSN 1538-7305. Disponível em: <<http://dx.doi.org/10.1002/j.1538-7305.1965.tb04146.x>>. Acesso em: 29 jun. 2018.
- LIN, S.; KERNIGHAN, B. W. An effective heuristic algorithm for the traveling-salesman problem. **Operations Research**, INFORMS, v. 21, n. 2, p. 498–516, 1973. ISSN 0030364X, 15265463. Disponível em: <<http://www.jstor.org/stable/169020>>. Acesso em: 29 jun. 2018.
- LIU, S. A hybrid population heuristic for the heterogeneous vehicle routing problems. **Transportation Research Part E: Logistics and Transportation Review**, v. 54, n. Supplement C, p. 67 – 78, 2013. ISSN 1366-5545. Disponível em: <<http://www.sciencedirect.com/science/article/pii/S1366554513000604>>. Acesso em: 29 jun. 2018.
- LIU, S.; HUANG, W.; MA, H. An effective genetic algorithm for the fleet size and mix vehicle routing problems. **Transportation Research Part E: Logistics and Transportation Review**, v. 45, n. 3, p. 434 – 445, 2009. ISSN 1366-5545. Disponível em: <<http://www.sciencedirect.com/science/article/pii/S1366554508001324>>. Acesso em: 29 jun. 2018.
- LYSGAARD, J.; LETCHFORD, A. N.; EGGLESE, R. W. A new branch-and-cut algorithm for the capacitated vehicle routing problem. **Mathematical Programming**, v. 100, n. 2, p. 423–445, Jun 2004. ISSN 1436-4646. Disponível em: <<https://doi.org/10.1007/s10107-003-0481-8>>. Acesso em: 29 jun. 2018.
- MIRHASSANI, S.; SAADATI, Z. A reactive adaptive memory metaheuristic for solving hffvrp. **Production & Manufacturing Research**, Taylor & Francis, v. 2, n. 1, p. 805–830, 2014.
- MLADENOVIĆ, N.; HANSEN, P. Variable neighborhood search. **Computers & Operations Research**, v. 24, n. 11, p. 1097 – 1100, 1997. ISSN 0305-0548. Disponível em: <<http://www.sciencedirect.com/science/article/pii/S0305054897000312>>. Acesso em: 29 jun. 2018.
- MOHAMMED, M. A.; AHMAD, M. S.; MOSTAFA, S. A. Using genetic algorithm in implementing capacitated vehicle routing problem. In: **2012 International Conference on Computer Information Science (ICIS)**. [S.l.: s.n.], 2012. v. 1, p. 257–262.
- MOLE, R. H.; JAMESON, S. R. A sequential route-building algorithm employing a generalised savings criterion. **Journal of the Operational Research Society**, v. 27, n. 2, p. 503–511, Jun

1976. ISSN 1476-9360. Disponível em: <<https://doi.org/10.1057/jors.1976.95>>. Acesso em: 29 jun. 2018.

NOVAES, A. G. 10 - roteirização de veículos. In: NOVAES, A. G. (Ed.). **Logística e Gerenciamento Da Cadeia De Distribuição**. Rio de Janeiro: Elsevier Editora Ltda., 2007. p. 303 – 330. ISBN 978-85-352-2415-3. Disponível em: <<http://www.sciencedirect.com/science/article/pii/B9788535224153500103>>. Acesso em: 29 jun. 2018.

OR, I. **Traveling salesman-type combinatorial problems and their relation to the logistics of regional blood banking**. Tese (Doutorado) — Department of Industrial Engineering and Management Sciences, Northwestern University, Evanston, 1976.

OSMAN, I. H.; KELLY, J. P. Meta-heuristics: An overview. In: OSMAN, I. H.; KELLY, J. P. (Ed.). **Meta-Heuristics: Theory and Applications**. Boston, MA: Springer US, 1996. p. 1–21. ISBN 978-1-4613-1361-8. Disponível em: <[https://doi.org/10.1007/978-1-4613-1361-8\\_1](https://doi.org/10.1007/978-1-4613-1361-8_1)>. Acesso em: 29 jun. 2018.

OSMAN, I. H.; LAPORTE, G. Metaheuristics: A bibliography. **Annals of Operations Research**, v. 63, n. 5, p. 511–623, Oct 1996. ISSN 1572-9338. Disponível em: <<https://doi.org/10.1007/BF02125421>>. Acesso em: 29 jun. 2018.

PENNA, P. H. V.; SUBRAMANIAN, A.; OCHI, L. S. An iterated local search heuristic for the heterogeneous fleet vehicle routing problem. **Journal of Heuristics**, Springer, v. 19, p. 201–232, 2013. ISSN 1572-9397. Disponível em: <<https://doi.org/10.1007/s10732-011-9186-y>>. Acesso em: 29 jun. 2018.

RALPHS, T.; KOPMAN, L.; PULLEYBLANK, W.; TROTTER, L. On the capacitated vehicle routing problem. **Mathematical Programming**, v. 94, n. 2, p. 343–359, Jan 2003. ISSN 1436-4646. Disponível em: <<https://doi.org/10.1007/s10107-002-0323-0>>. Acesso em: 29 jun. 2018.

RENAUD, J.; BOCTOR, F. F.; LAPORTE, G. A fast composite heuristic for the symmetric traveling salesman problem. **INFORMS Journal on Computing**, v. 8, n. 2, p. 134–143, 1996. Disponível em: <<https://doi.org/10.1287/ijoc.8.2.134>>. Acesso em: 29 jun. 2018.

SAVELSBERGH, M. W. P. Local search in routing problems with time windows. **Annals of Operations Research**, v. 4, n. 1, p. 285–305, Dec 1985. ISSN 1572-9338. Disponível em: <<https://doi.org/10.1007/BF02022044>>. Acesso em: 29 jun. 2018.

SILVA, M. M.; SUBRAMANIAN, A.; OCHI, L. S. An iterated local search heuristic for the split delivery vehicle routing problem. **Computers & Operations Research**, v. 53, n. Supplement C, p. 234 – 249, 2015. ISSN 0305-0548. Disponível em: <<http://www.sciencedirect.com/science/article/pii/S0305054814002159>>. Acesso em: 29 jun. 2018.

SINRUK, P.; TANGMO, Y. Heterogeneous fleet vehicle routing problem for retail construction material stores. In: IEEE. **Industrial Engineering and Applications (ICIEA), 2017 4th International Conference on**. Nagoya, 2017. p. 234–238. Disponível em: <<http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=7939213&isnumber=7939161>>. Acesso em: 29 jun. 2018.

- SUBRAMANIAN, A.; DRUMMOND, L.; BENTES, C.; OCHI, L.; FARIAS, R. A parallel heuristic for the vehicle routing problem with simultaneous pickup and delivery. **Computers & Operations Research**, v. 37, n. 11, p. 1899 – 1911, 2010. ISSN 0305-0548. Metaheuristics for Logistics and Vehicle Routing. Disponível em: <<http://www.sciencedirect.com/science/article/pii/S0305054809002779>>. Acesso em: 29 jun. 2018.
- SUBRAMANIAN, A.; PENNA, P. H. V.; UCHOA, E.; OCHI, L. S. A hybrid algorithm for the heterogeneous fleet vehicle routing problem. **European Journal of Operational Research**, Elsevier, v. 221, n. 2, p. 285–295, 2012.
- TAILLARD, É. Parallel iterative search methods for vehicle routing problems. **Networks**, Wiley Subscription Services, Inc., A Wiley Company, v. 23, n. 8, p. 661–673, 1993. ISSN 1097-0037. Disponível em: <<http://dx.doi.org/10.1002/net.3230230804>>. Acesso em: 29 jun. 2018.
- TARANTILIS, C.; KIRANOUDIS, C. A flexible adaptive memory-based algorithm for real-life transportation operations: Two case studies from dairy and construction sector. **European Journal of Operational Research**, v. 179, n. 3, p. 806 – 822, 2007. ISSN 0377-2217. Disponível em: <<http://www.sciencedirect.com/science/article/pii/S0377221705007460>>. Acesso em: 29 jun. 2018.
- TEIXEIRA, R. G.; CUNHA, C. B. da. Heurísticas para o problema de dimensionamento e roteirização de uma frota heterogênea utilizando o algoritmo out-of-kilter. **Transportes**, ANPET - Associação Nacional de Pesquisa e Ensino em Transportes, v. 10, n. 2, p. 9–30, 2002.
- THOMPSON, P. M.; PSARAFTIS, H. N. Cyclic transfer algorithm for multivehicle routing and scheduling problems. **Operations Research**, v. 41, n. 5, p. 935–946, 1993. Disponível em: <<https://doi.org/10.1287/opre.41.5.935>>. Acesso em: 29 jun. 2018.
- TOTH, P.; VIGO, D. **The Vehicle Routing Problem**. Society for Industrial and Applied Mathematics, 2002. Disponível em: <<http://epubs.siam.org/doi/abs/10.1137/1.9780898718515>>. Acesso em: 29 jun. 2018.
- WASSAN, N. A.; OSMAN, I. H. Tabu search variants for the mix fleet vehicle routing problem. **The Journal of the Operational Research Society**, Palgrave Macmillan Journals, v. 53, n. 7, p. 768–782, 2002. ISSN 01605682, 14769360. Disponível em: <<http://www.jstor.org/stable/822764>>. Acesso em: 29 jun. 2018.
- WREN, A. **Computers in transport planning and operation**. Riverdene Business Park: Transport and Road Research Laboratory (TRRL), 1971.
- WREN, A.; HOLLIDAY, A. Computer scheduling of vehicles from one or more depots to a number of delivery points. **Journal of the Operational Research Society**, v. 23, n. 3, p. 333–344, Sep 1972. ISSN 1476-9360. Disponível em: <<https://doi.org/10.1057/jors.1972.53>>. Acesso em: 29 jun. 2018.
- XU, J.; KELLY, J. P. A network flow-based tabu search heuristic for the vehicle routing problem. **Transportation Science**, v. 30, n. 4, p. 379–393, 1996. Disponível em: <<https://doi.org/10.1287/trsc.30.4.379>>. Acesso em: 29 jun. 2018.