

UNIVERSIDADE FEDERAL DE OURO PRETO  
DEPARTAMENTO DE COMPUTAÇÃO

Leonardo Luiz Freitas de Resende

**UMA HEURÍSTICA PARA O PROBLEMA DE  
ESTOQUE COM ROTEAMENTO DE  
MÚLTIPLOS VEÍCULOS**

Ouro Preto, MG  
2022

Leonardo Luiz Freitas de Resende

UNIVERSIDADE FEDERAL DE OURO PRETO  
DEPARTAMENTO DE COMPUTAÇÃO

Monografia II apresentada ao Curso de Ciência da Computação da Universidade Federal de Ouro Preto como parte dos requisitos necessários para a obtenção do grau de Bacharel em Ciência da Computação.

**Orientador:** Prof. Dr. Puca Huachi Vaz Penna

Ouro Preto, MG  
2022

## SISBIN - SISTEMA DE BIBLIOTECAS E INFORMAÇÃO

R433u Resende, Leonardo Luiz Freitas de.  
Uma heurística para o problema de estoque com roteamento de múltiplos veículos. [manuscrito] / Leonardo Luiz Freitas de Resende. - 2022.  
39 f.

Orientador: Prof. Dr. Puca Huachi Vaz Penna.  
Monografia (Bacharelado). Universidade Federal de Ouro Preto.  
Instituto de Ciências Exatas e Biológicas. Graduação em Ciência da Computação .

1. Pesquisa Operacional. 2. Estoque com Roteamento de Veículos. 3. Meta-heurísticas. I. Penna, Puca Huachi Vaz. II. Universidade Federal de Ouro Preto. III. Título.

CDU 004

Bibliotecário(a) Responsável: Luciana De Oliveira - SIAPE: 1.937.800



## FOLHA DE APROVAÇÃO

**Leonardo Luiz Freitas de Resende**

### UMA HEURÍSTICA PARA O PROBLEMA DE ESTOQUE COM ROTEAMENTO DE MÚLTIPLOS VEÍCULOS

Monografia apresentada ao Curso de Ciência da Computação da Universidade Federal de Ouro Preto como requisito parcial para obtenção do título de Bacharel em Ciência da Computação

Aprovada em 21 de Março de 2022.

#### Membros da banca

Puca Huachi Vaz Penna (Orientador) - Doutor - Universidade Federal de Ouro Preto  
Pedro Henrique Lopes Silva (Examinador) - Mestre - Universidade Federal de Ouro Preto  
André Luyde da Silva Souza (Examinador) - Mestre - Escola Politécnica de Minas Gerais

Puca Huachi Vaz Penna, Orientador do trabalho, aprovou a versão final e autorizou seu depósito na Biblioteca Digital de Trabalhos de Conclusão de Curso da UFOP em 21/03/2022.



Documento assinado eletronicamente por **Puca Huachi Vaz Penna, PROFESSOR DE MAGISTERIO SUPERIOR**, em 24/03/2022, às 15:56, conforme horário oficial de Brasília, com fundamento no art. 6º, § 1º, do [Decreto nº 8.539, de 8 de outubro de 2015](#).



A autenticidade deste documento pode ser conferida no site [http://sei.ufop.br/sei/controlador\\_externo.php?acao=documento\\_conferir&id\\_orgao\\_acesso\\_externo=0](http://sei.ufop.br/sei/controlador_externo.php?acao=documento_conferir&id_orgao_acesso_externo=0), informando o código verificador **0297982** e o código CRC **C77336F5**.

# Resumo

Este estudo apresenta uma abordagem para o Problema de Estoque e Roteamento de Múltiplos Veículos (PERMV). Para tratar o problema é apresentada uma Meta-Heurística, sendo ela *Iterated Local Search* - ILS, com o método de Busca Local *Randomized Variable Neighborhood Descent* (RVND). O método proposto utiliza três diferentes estruturas de vizinhança, sendo elas trocas intrarota, trocas inter-rotas e realocação de clientes inter-rotas, em conjunto com uma série de perturbações na solução corrente. O método proposto foi avaliado através de experimentos computacionais realizados em um conjunto de instâncias da literatura, encontrando soluções de qualidade e tempo de processamento aceitável. Apresentando melhores resultados para instâncias de alto custo (*High*), com um GAP médio de 10,21% e para instâncias de baixo custo (*Low*) o GAP médio foi de 15,47%.

**Palavras-chave:** Estoque com Roteamento de Veículos, Meta-Heurísticas, Pesquisa Operacional.

# Abstract

This study presents an approach to the Inventory and Routing Problem (IRP) with Multiples Vehicles. To address the problem, a Meta-Heuristic is presented, which is Iterated Local Search - ILS, with the Randomized Variable Neighborhood Descent Local Search method (RVND). The proposed method uses three different neighborhood structures, which are exchanges intra-route, inter-route exchanges and inter-route customer relocation, together with a series of of disturbances in the current solution. The proposed method was evaluated through computational experiments carried out in a set of instances of the literature, finding solutions of acceptable quality and processing time. Presenting better results for high-cost (*High*) instances, with an average GAP of 10.21% and for low-cost (*Low*) instances, the average GAP was 15.47%.

**Keywords:** Inventory Routing Problem, metaheuristics, Operational Research.

# Lista de Ilustrações

Figura 3.1 – Exemplo de solução . . . . .	16
Figura 3.2 – Exemplo de aplicação de vizinhança intrarrotas . . . . .	18
Figura 3.3 – Exemplo de aplicação de vizinhança inter-rota . . . . .	19
Figura 3.4 – Exemplo de aplicação de vizinhança de realocação de clientes inter-rotas . .	20
Figura 3.5 – Exemplo de execução do método de Perturbação . . . . .	21

# Lista de Tabelas

Tabela 2.1 – Classificação do PERV . . . . .	5
Tabela 2.2 – Trabalhos relacionados e características . . . . .	14
Tabela 4.1 – Comparativo de resultados utilizando 2 veículos . . . . .	23
Tabela 4.2 – Resultados completo instâncias de baixo custo com 2 veículos . . . . .	24
Tabela 4.3 – Resultados completo instâncias de alto custo com 2 veículos . . . . .	25

# Lista de Abreviaturas e Siglas

ABML	Associação Brasileira de Movimentação e Logística
DECOM	Departamento de Computação
EDI	<i>Electronic Data Interchange</i>
ILS	<i>Iterated Local Search</i>
ML	<i>Maximum Level</i>
MIRP	<i>Multi-Vehicle Routing and Inventory Problem</i>
OU	<i>Order-up-to</i>
PVC	Problema do Caixeiro Viajante
PRV	Problema de Roteamento de Veículos
PERV	Problema de Estoque e Roteamento de Veículos
PERMV	Problema de Estoque e Roteamento de Veículos com Múltiplos Veículos
RVND	<i>Randomized Variable Neighborhood Descent</i>
SPEA2	<i>Strength Pareto Evolutionary Algorithm 2</i>
UFOP	Universidade Federal de Ouro Preto
VMI	<i>Vendor Managed Inventory - VMI</i>

# Sumário

<b>1</b>	<b>Introdução</b>	<b>1</b>
1.1	Justificativa	2
1.2	Objetivos Geral e Específicos	2
1.3	Organização do Trabalho	2
<b>2</b>	<b>Revisão Bibliográfica</b>	<b>4</b>
2.1	Fundamentação Teórica	4
2.1.1	Problema de Roteamento de Veículos (PRV)	4
2.1.2	Problema de Estoque e Roteamento de Veículos (PERV)	5
2.1.3	Problema de Estoque e Roteamento de Múltiplos Veículos (PERMV)	7
2.1.4	Modelagem Matemática	9
2.2	Trabalhos Relacionados	10
<b>3</b>	<b>Desenvolvimento</b>	<b>15</b>
3.1	Abordagem da Solução	15
3.1.1	Representação da solução	15
3.1.2	<i>Iterated Local Search</i> - ILS	16
3.1.3	<i>Randomized Variable Neighborhood Descent</i> (RVND)	17
3.1.4	Perturbação	20
<b>4</b>	<b>Resultados</b>	<b>22</b>
4.1	Cenário de Testes	22
4.2	Resultados Obtidos	23
<b>5</b>	<b>Conclusões</b>	<b>26</b>
	<b>Referências</b>	<b>27</b>

# 1 Introdução

Com o objetivo de suprir as principais exigências de um mercado global competitivo, tais como a venda de produtos com baixa vida útil, o aumento do nível de serviço, a melhoria na eficiência e a redução de custos, as empresas têm voltado sua atenção ao gerenciamento de suas cadeias de suprimentos. Tendo em vista que a inclusão de sistemas de gestão modernizados e a incessante demanda por processos mais eficientes tornaram-se pilares para o sucesso no meio empresarial, a logística ganha uma nova proporção, integrando as atividades ao longo da cadeia de valores, desde a compra de matéria-prima até a logística de entrega dos produtos, incluindo as fases de produção, embalagem, transporte e armazenamento. Sendo assim, o gerenciamento da cadeia de suprimentos surge como forte opção a este novo contexto de concorrência empresarial.

Segundo [Filho et al. \(2003\)](#) um estudo realizado pela Associação Brasileira de Movimentação e Logística (ABML), atualmente o custo estimado com logística em uma empresa é equivalente, em média, a 19% de seu faturamento total, transformando a logística em uma área de estudo promissora com o objetivo de minimizar custos ao longo da cadeia de suprimentos.

Com isso, surgiram algumas iniciativas visando elevar o ganho total da cadeia de suprimento, como o processo de reposição de estoque gerenciado pelo fornecedor (*Vendor Managed Inventory* - VMI). Sendo que o mesmo tem como objetivo a redução simultânea dos custos de dois componentes da cadeia de suprimentos, o fornecedor e os clientes. Portanto, o VMI é um processo onde o fornecedor é responsável por gerenciar os níveis de estoque de seus clientes, e escolher quando e quanto entregar de mercadorias atendendo suas demandas. Ao longo dos últimos anos, essa técnica tem sido amplamente utilizada, impulsionada pela redução dos custos do intercâmbio eletrônico de dados (*Electronic Data Interchange*– EDI) e com o grande avanço tecnológico, os quais possibilitam ao fornecedor utilizar equipamentos de ponta capazes de medir os níveis de estoque dos clientes e coletar informações em tempo real a partir de qualquer celular, tablet ou computador. Segundo [Belfiore et al. \(2006b\)](#), o VMI é amplamente utilizado em indústrias com campos de atuação distintos, como indústrias de bebidas, automobilísticas, petroquímica, varejo, lojas de departamento. Nesse cenário, ocorre o problema integrado de estoque e roteamento de veículos, apresentando como principal aspecto o processo de VMI.

O Problema de Estoque e Roteamento de Veículos (PERV) tem sua origem em três diferentes problemas da literatura, sendo eles gestão de estoque, Roteamento de Veículos e decisões de programação de entregas englobando estoque e roteamento. Para mais informações consultar [Belfiore et al. \(2006a\)](#), [Coelho et al. \(2013\)](#). Aos clientes, proporciona um aumento no nível de serviço, no quesito de disponibilidade do produto e um menor investimento em recursos de gerenciamento de estoque. E, aos fornecedores, apresenta uma melhor utilização dos recursos disponíveis, além de uma minimização dos custos de estoque e distribuição.

Neste estudo é proposto uma solução para o Problema de Estoque com Roteamento de Múltiplos Veículos, considerando horizonte de tempo finito, demanda determinística, estrutura Um-Para-Muitos (sendo um único depósito e vários clientes), política de estoque *Order-to-up* (OU), garante que ao realizar uma entrega a quantidade máxima de estoque deve ser satisfeita, frota homogênea e decisão de estoque Não-negativo.

## 1.1 Justificativa

Segundo Bell et al. (1983), a maioria dos problemas que envolvem roteamento de veículos combinados com distribuição de produtos, onde as decisões de aquisição são realizadas pelos clientes, podem ser classificados como sendo Problema de Estoque e Roteamento de Veículos (PERV). O problema pertence à classe NP-Difícil e não se conhece nenhum método exato capaz de resolver instâncias do PERV de tamanho razoável em um tempo computacional aceitável. Portanto, tal fato possibilita um vasto campo de pesquisa envolvendo meta-heurísticas com o objetivo de encontrar soluções de boa qualidade. Isso porque apresenta grande aplicabilidade em situações cotidianas, com diversas possibilidades e variações de aplicação.

## 1.2 Objetivos Geral e Específicos

Este trabalho tem como principal objetivo realizar um estudo do Problema de Estoque e Roteamento de Múltiplos Veículos (PERMV) e propor a elaboração de uma meta-heurística consistente visando a redução simultânea dos custos de estoque e transporte. De um modo geral, os objetivos específicos são:

- Realização de um estudo sobre os métodos utilizados na literatura.
- Definição e implementação de uma meta-heurística para obtenção de soluções para o problema.
- Realização de experimentos computacionais e análise dos resultados obtidos.

## 1.3 Organização do Trabalho

O presente estudo encontra-se estruturado da seguinte forma: no Capítulo 2, será apresentada uma revisão bibliográfica sobre o PERV, com foco no problema abordado neste estudo. Ainda no Capítulo 2, são apresentados de forma resumida os principais trabalhos publicados sobre o PERMV. No Capítulo 3, é feita uma breve introdução sobre a classe de complexidade do problema abordado e em seguida é definido o escopo do método a ser utilizado. Além disso, são apresentados os parâmetros, variáveis e, por fim, a formulação completa do modelo matemático do problema tratado. No Capítulo 4, é apresentado o cenário de teste, instâncias utilizadas e,

finalmente, os resultados obtidos. Por fim, no Capítulo 5, são apresentadas as considerações finais a respeito do trabalho e as propostas para trabalhos futuros.

## 2 Revisão Bibliográfica

Este capítulo apresenta a revisão de literatura realizada para o desenvolvimento deste trabalho. Encontra-se organizado da seguinte forma: A Seção 2.1 apresenta a fundamentação teórica necessária para o entendimento deste trabalho e a Seção 2.2 contém os trabalhos relacionados ao tema estudado.

### 2.1 Fundamentação Teórica

Esta seção apresenta a fundamentação teórica e modelagem matemática do Problema de estoque e roteamento de múltiplos veículos, além de métodos disponíveis no estado da arte que serão utilizados no desenvolvimento desse trabalho.

Seção 2.1.1 apresenta uma introdução do PRV de acordo com [Larsen e Madsen \(2000\)](#). A Seção 2.1.2 apresenta uma introdução, fundamentação, classificação e as variações do PERV de acordo com [Almeida \(2014\)](#). A Seção 2.1.3 apresenta uma breve introdução ao PERMV e sua definição formal segundo [Almeida \(2014\)](#). A Seção 2.1.4 apresenta a modelagem matemática do PERMV proposta por [Coelho \(2013\)](#).

#### 2.1.1 Problema de Roteamento de Veículos (PRV)

De acordo com o estudo realizado por [Larsen e Madsen \(2000\)](#), o problema de roteamento de veículos mais conhecido e estudado da literatura é o problema do Caixeiro Viajante (PCV), que consiste basicamente em determinar a menor rota para percorrer um conjunto de cidades, visitando cada uma delas uma única vez, e retornar à cidade de origem. O problema tem como principal objetivo minimizar a distância total percorrida. O problema consiste em encontrar um ciclo hamiltoniano em um grafo, o problema é classificado como NP-Difícil, segundo [Karp \(1975\)](#), portanto, não se conhece nenhum algoritmo polinomial determinístico capaz de resolver o problema.

O PRV, é uma variante do PCV, resumindo-se em determinar rotas de veículos, sendo que uma rota é um ciclo que se inicia em um depósito, e então visita um subconjunto de clientes ou locais de demanda em uma determinada ordem e por fim retorna ao depósito de origem. Sendo que os locais de demanda devem ser visitados uma única vez e o total da demanda do conjunto de clientes não pode exceder a capacidade de transporte do veículo, segundo descrito por [Larsen e Madsen \(2000\)](#).

### 2.1.2 Problema de Estoque e Roteamento de Veículos (PERV)

Como descrito por Almeida (2014), o PERV engloba três diferentes problemas, sendo eles; gestão de estoque, roteamento de veículos e decisões de programação de entregas. De acordo com o estudo realizado por Shiguemoto et al. (2004), devido à competição global, empresas de diferentes setores privados ou estatais buscam melhorar seus produtos e diminuir os custos envolvidos ao longo da Cadeia de suprimentos.

Uma cadeia de suprimentos envolve desde a compra de matéria-prima até a logística de entrega dos produtos, incluindo as fases de produção, embalagem, transporte e armazenamento. A cadeia de suprimentos tem como objetivo maximizar o lucro total e não de etapas de forma isolada. Com isso, surgiram algumas iniciativas visando elevar o ganho total da cadeia de suprimento, como o processo de reposição de estoque gerenciado pelo fornecedor (*Vendor Managed Inventory-VMI*). Esse método de reposição tem como objetivo a redução simultânea dos custo de estoque (do cliente e fornecedor) e transporte, em que o fornecedor monitora o nível de estoque de seus clientes e toma decisões baseadas na demanda do cliente. Além disso, como o fornecedor gerencia a distribuição, é possível atender as demandas de forma conjunta, ou seja, promover um melhor planejamento do roteamento de entrega, possibilitando um melhor aproveitamento da capacidade de transporte.

Segundo Almeida (2014), o problema de estoque tem como objetivo a redução dos custos de estocagem (no fornecedor e clientes) de produtos, cumprindo as restrições, que, na maioria das vezes, dizem respeito à capacidade de estocagem e à forma de operação da empresa. A maioria dos problemas que englobam a distribuição de produtos podem ser classificados como PERV, sendo válidos para quando as decisões de quando e quanto entregar a cada cliente são feitas pelo fornecedor, através de um depósito central, com o objetivo de minimizar os custos totais de distribuição e estocagem.

Desde sua origem no trabalho de Bell et al. (1983), surgiram vários estudos com diversas variantes e diferentes abordagens do PERV. A Tabela 2.1 mostra a classificação (não necessariamente deve se limitar a essas) das variantes do PERV baseados em oito critérios, de acordo com Andersson et al. (2010). As variantes tratadas neste estudo estão em negrito na tabela 2.1.

Tabela 2.1 – Classificação do PERV

Critério	Possíveis Opções		
	Tempo	<b>Finito</b>	Infinito
Demanda	<b>Determinística</b>	Estocástica	-
Estrutura	Um-Para-Um	<b>Um-Para-Muitos</b>	Muitos-Para-Muitos
Roteamento	Direto	<b>Múltiplos</b>	Contínuo
Política Estoque	<i>Maximum Level (ML)</i>	<b><i>Order-to-up (OU)</i></b>	-
Decisão de Estoque	Perda de venda	<i>Back-order</i>	<b>Não-negativo</b>
Tipo de Frota	<b>Homogênea</b>	Heterogênea	-
Tamanho da Frota	1 Veículo	<b>Múltiplos</b>	Irrestrito

Fonte: (ALMEIDA, 2014)

Na Tabela 2.1 são definidos os critérios básicos do PRV, como sendo:

- **Tempo:** Horizonte de tempo considerado pelo modelo, podendo ser finito ou infinito.
- **Demanda:** Tipo de demanda a ser utilizada no modelo.
  - **Determinística:** quando as demandas dos clientes são conhecidas a priori.
  - **Estocástica:** quando as demandas dos clientes não são conhecidas a priori.
- **Estrutura:** Como o número de clientes e fornecedores são variáveis, temos:
  - **Um-Para-Um:** Um fornecedor e apenas um único cliente.
  - **Um-Para-Muitos:** Um fornecedor e vários clientes.
  - **Muitos-Para-Muitos:** Vários fornecedores e vários clientes.
- **Roteamento:** Pode ser definido como sendo:
  - **Direto:** um único cliente é atendido por rota.
  - **Múltiplos:** mais de um cliente são atendidos por rota.
  - **Contínuo:** é quando não se tem um depósito central, utilizado por exemplo em aplicações marítimas.
- **Política de estoque:** São regras utilizadas para atender os clientes nas entregas, sendo as duas mais utilizadas:
  - **ML:** A quantidade entregue é flexível, porém respeitando as restrições de estoque de cada cliente.
  - **OU:** Sempre que um cliente é visitado, garante que a capacidade máxima de estoque seja satisfeita.
- **Decisão de estoque:** Determina a modelagem da gestão de estoque. podendo ser:
  - **Back-Order:** Permite que o estoque fique negativo e que a demanda seja atendida em um momento futuro.
  - **Perda de Vendas:** Perda de vendas em situações de demanda extra que não é utilizado *Back-Order*.
  - **Não-Negativo:** Não permite que o estoque fique negativo.
- **Tipo de frota:** Define a computação da frota de veículos, podendo ser homogênea ou heterogênea.
- **Tamanho de Frota:** Define a quantidade de veículos a serem considerados.

Tradicionalmente, fica a critério do fornecedor escolher a quantidade a ser entregue aos clientes, respeitando apenas as restrições de limite de estoques dos clientes e a capacidade de transporte. Essa política de entrega é conhecida como *Maximum Level* (ML). Foram propostas várias heurísticas, segundo os estudos de Archetti et al. (2012), Bertazzi et al. (2002) e Coelho (2013), além de um algoritmo exato proposto por Archetti et al. (2007), excepcionalmente para o caso de um único veículo. Porém, para evitar que o comportamento do VMI produza resultados inconvenientes para o fornecedor e clientes, como por exemplo, várias entregas consecutivas de pequenas quantidades em diferentes dias para um mesmo cliente e posteriormente o cliente ficar um longo prazo sem receber entregas, as soluções devem auxiliar os ganhos no nível de serviço para os clientes, além de diminuir custo de forma sistêmica. Para tal, Coelho (2013) propõe o conceito de consistência do PERV, adicionando 6 variações ao problema, sendo elas:

- **Quantidade consistente** : As entregas aos clientes devem respeitar as restrições de limite mínimo e máximo, eliminando grandes variações.
- **Taxa de ocupação dos veículos**: Os veículos serão alocados para entrega apenas se atingir a ocupação mínima.
- **Order-up-to (OU)** : Certifica-se que, quando houver entrega a um cliente, o limite máximo de estoque do cliente deve ser satisfeito.
- **Consistência do veículo/motorista**: Certifica-se que o mesmo veículo ou motorista atenda sempre os mesmos clientes.
- **Consistência parcial do veículo/motorista**: Os clientes podem ser atendidos por um veículo ou motorista diferente, porém é aplicada uma penalidade.
- **Espaçamento de visita**: Define um intervalo mínimo e máximo entre duas entregas consecutivas a um cliente.

### 2.1.3 Problema de Estoque e Roteamento de Múltiplos Veículos (PERMV)

Segundo Almeida (2014), o PERMV é uma variante do PERV com uma característica específica, a de múltiplos veículos, e tem como principal objetivo reduzir os custos de estoque e transporte de forma simultânea, tratando as demandas de clientes. O PERMV é essencialmente determinado como sendo um grafo  $G = \{(L, A)\}$ , sendo  $L$  o conjunto de vértices representando o depósito e os clientes, e o conjunto de arcos  $A$  define os caminhos entre clientes e depósito. O

vértice 0 foi utilizado para definir o depósito e os vértices de  $L' = L \setminus \{0\}$ , os clientes.

Considerando as seguintes definições :

$L =$  Representa o depósito e os clientes,  $L = \{0, \dots, n\}$ ;

$A = \{(i, j) : i, j \in L, i \neq j\} =$  Representa um caminho entre o depósito ou cliente  $i$  e o cliente ou depósito  $j$ ;

$T =$  Horizonte de planejamento,  $T = \{0, \dots, p\}$ ;

$V =$  Veículos disponíveis,  $V = \{0, \dots, v\}$ ;

Variáveis:

$Cv_v =$  Capacidade de transporte do veículo  $v$ ;

$Cr_v =$  Custo variável de transporte do veículo  $v$ ;

$Qf_p =$  Produção do depósito no período  $p$ ;

$Qc_{ip} =$  Consumo do cliente  $i$  no período  $p$ ;

$Emax_i =$  Valor máximo de depósito do local  $i$ ;

$Emin_i =$  Valor mínimo de depósito do local  $i$ ;

$Eo_i =$  Valor inicial de depósito do local  $i$ ;

$Ce_i =$  Custo unitário de estoque do local  $i$ ;

$D_{ij} =$  Distância entre o local  $i$  e o local  $j$ ;

$Cut_{vij} = D_{ij}.Cv_v =$  Custo de transporte entre o local  $i$  e o local  $j$ ;

Sendo que local, engloba depósito e clientes.

E o planejamento de fornecimento dos clientes tem como restrições:

1. Ao término de um período  $p$ , os limites de estoque mínimo e máximo devem ser válidos.
2. Cada viagem deve iniciar e terminar na origem (depósito).
3. O limite de transporte do veículo deve ser respeitado.
4. Os clientes podem receber no máximo uma visita por período.

A resolução do problema abordado estabelece quais clientes vão ser atendidos em cada período, qual veículo vai ser alocado para cada rota, o quanto deve ser entregue em cada visita aos clientes e qual a ordem de atendimento aos clientes em cada roteiro. Foi definido que, no trabalho em questão, a  $Qf_p$  (quantidade produzida) no período  $p$  fica disponível de forma imediata e pode ser entregue aos clientes no mesmo período de sua produção e as entregas recebidas  $x_{vip}$  pelo cliente  $i$  no período  $p$  podem ser usadas para suprir a demanda do cliente no mesmo período de recebimento.

### 2.1.4 Modelagem Matemática

O modelo matemático a seguir foi proposto por Coelho (2013) como sendo uma amplificação dos modelos de Bertazzi et al. (2002) e Archetti et al. (2012) de um único veículo para múltiplos veículos, com limitação para instâncias muito grandes. O modelo em questão trabalha com as seguintes variáveis:

$$w_{vijp} = \begin{cases} 1, & \text{se o veículo } v \text{ viaja do local } i \text{ para o local } j \text{ no período } p \\ 0, & \text{caso contrário} \end{cases}$$

$$y_{vip} = \begin{cases} 1, & \text{se o veículo } v \text{ visita o cliente } i \text{ no período } p \\ 0, & \text{caso contrário} \end{cases}$$

$x_{vip} \in \mathbb{R}^+$  = representa a quantidade de produto entregue pelo veículo  $v$  no cliente  $i$  no período  $p$ .

$e_{ip} \in \mathbb{R}^+$  = define o estoque do local  $i$  no período  $p$ .

$z_{vip} \in \mathbb{R}^+$  = garante a eliminação de subciclos.

$$\text{Minimizar} \quad \sum_{i \in L} C e_i \cdot E o_i + \sum_{i \in L} \sum_{p \in T} C e_i \cdot e_{ip} + \sum_{v \in V} \sum_{i \in L} \sum_{j \in L} \sum_{p \in T} C u t_{vij} \cdot w_{vijp} \quad (2.1)$$

$$E m i n_i \leq e_{ip} \leq E m a x_i \quad \forall i \in L, p \in T \quad (2.2)$$

$$e_{00} = E o_0 + Q f_0 - \sum_{v \in V} \sum_{i \in L'} x_{vi0} \quad (2.3)$$

$$e_{0p} = e_{0,p-1} + Q f_p - \sum_{v \in V} \sum_{i \in L'} x_{vip} \quad \forall p \in T \setminus \{0\} \quad (2.4)$$

$$e_{i0} = E o_0 - Q c_{i0} + \sum_{v \in V} x_{vi0} \quad \forall i \in L' \quad (2.5)$$

$$e_{ip} = e_{i,p-1} - Q c_{ip} + \sum_{v \in V} x_{vip} \quad \forall i \in L', p \in T \setminus \{0\} \quad (2.6)$$

$$\sum_{v \in V} x_{vip} \leq E m a x_i - e_{i,p-1} \quad \{\forall (i \in L', p \in T)\} \quad (2.7)$$

$$\sum_{v \in V} x_{vip} \leq E m a x_i \cdot \sum_{v \in V} \sum_{j \in L} w_{vijp} \quad \{\forall (i \in L', p \in T)\} \quad (2.8)$$

$$\sum_{i \in L'} x_{vip} \leq C v_v \quad \{\forall (v \in V, p \in T)\} \quad (2.9)$$

$$x_{vip} \leq Emax_i \cdot y_{vip} \quad \{\forall (v \in V, i \in L', p \in T)\} \quad (2.10)$$

$$\sum_{j \in L} w_{vijp} = \sum_{j \in L} w_{vjip} = y_{vip} \quad \{\forall (v \in V, i \in L', p \in T)\} \quad (2.11)$$

$$\sum_{j \in L'} w_{v0jp} \leq 1 \quad \{\forall (v \in V, p \in T)\} \quad (2.12)$$

$$z_{vip} - z_{vjp} + C_{Vv} \cdot w_{vijp} \leq C_{Vv} - x_{vjp} \quad \{\forall (i \in L', j \in L', p \in T, v \in V)\} \quad (2.13)$$

$$x_{vip} \leq z_{vip} \leq C_{Vv} \quad \{\forall (i \in L', p \in T, v \in V)\} \quad (2.14)$$

Sendo que a função objetivo (2.1) é definida pela soma dos custos de transporte e estoque no fornecedor e cliente. As restrições (2.2) garantem que as restrições de limite de estoque mínimo e máximo sejam respeitadas, no cliente e fornecedor. As restrições (2.3) e (2.4) representam o estoque do fornecedor em um período  $p$ . As restrições (2.5) e (2.6) representam o estoque de clientes em um período  $p$ . As restrições (2.7) e (2.8) asseguram que a quantidade entregue a um cliente  $i$  no período  $p$  respeita as restrições de limite de estoque caso o cliente seja atendido, e zero caso contrário. As restrições (2.9) asseguram que os limites de carga do veículo não sejam ultrapassados. As restrições (2.10 - 2.14) definem as condições de roteamento. Por fim, a restrição (2.13) assegura a consistência de carga em todos os veículos e previne subciclos.

## 2.2 Trabalhos Relacionados

Na abordagem de [Silveira et al. \(1992\)](#), o PERV foi modelado de três formas, sendo que as duas primeiras não trabalham com restrições sobre a duração das rotas e estão relacionadas com o PRV ou PCV, já a terceira abordagem inclui restrições sobre a duração total das rotas executadas por um veículo. A metodologia de resolução parte de uma das formulações apresentadas, de maneira que, nas duas primeiras, é necessário apenas simplificar a função objetivo obtendo um problema de atribuição generalizado e, posteriormente, os subproblemas resultantes são removidos, os quais podem ser PRVs ou PCVs, obtendo uma solução inicial. Na terceira formulação, foram relaxadas as restrições de disponibilidade de atendimento dos veículos, permitindo aplicar uma solução inicial obtida em uma das formulações anteriores. Então, aplicaram-se duas heurísticas com o objetivo de melhorar a solução inicial, sendo elas: busca Tabu adicionando penalidades e trocas simultâneas. Essas heurísticas propostas apresentaram grandes melhorias sobre as soluções iniciais. Entretanto, a busca tabu apresentou resultados superiores à heurística de trocas, porém com uma alta demanda em tempo de processamento.

[Archetti et al. \(2007\)](#) trabalhou com o PERV onde um produto deve ser enviado de um fornecedor para vários varejistas ao longo de um horizonte de tempo, onde cada varejista

seleciona um nível de estoque máximo. Nesse modelo, o fornecedor fica responsável por monitorar o inventário de cada varejista e escolher qual política de reposição deve ser adotada, garantindo que as restrições de estoque sejam respeitadas. Considerando as seguintes restrições: a cada visita a um determinado varejista, a quantidade entregue deve ser suficiente para atingir o nível máximo do depósito (Order-up-to level policy); o transporte do fornecedor aos varejistas é realizado por um veículo com limites de capacidade, sendo que o objetivo do problema é definir para cada instante discreto de tempo a quantidade a ser entregue a cada varejista e a rota a ser utilizada. Com isso, foi apresentado um modelo de programação linear inteiro misto com adicional de novas desigualdades válidas adicionais usadas para impulsionar o relaxamento linear do modelo, além de a implementação de um algoritmo *branch-and-cut* para uma resolução otimizada do mesmo. Além disso, foi realizada uma comparação entre a solução ótima do problema com a solução ótima de dois problemas obtidos com o relaxamento de diferentes maneiras da política determinística *order-up-to level*. A eficácia do método proposto foi comprovada através de testes com instâncias de até 50 clientes e horizonte de tempo igual a três onde seus resultados foram comparados com os resultados obtidos pelo método exato.

O trabalho proposto por Shiguemoto et al. (2008) refere-se à otimização de dois problemas conjuntos de uma seção de cadeia de suprimento. O primeiro é produção-distribuição ao longo de períodos de um horizonte de tempo finito. Uma planta com limitações de capacidade trabalha com vários produtos e uma frota homogênea de veículos é utilizada para suprir a demanda dos clientes, sendo que, em cada período, os problemas de produção e distribuição definem, respectivamente, o quanto deve ser processado de cada produto e a quantidade a ser entregue aos clientes utilizando a melhor rota. O objetivo dele foi minimizar os custos de produção e estoque na planta e nos clientes, além dos custos de distribuição. Para isso, foi utilizada a meta-heurística busca Tabu em conjunto com um procedimento de religamento de caminho, o qual permite soluções infactíveis durante a busca. Uma segunda abordagem envolve o problema estoque-produção com restrições de demanda estocástica de apenas um produto, especificada por uma distribuição discreta de probabilidades. Nessa abordagem, foi definido pelo fornecedor quanto e quando entregar aos clientes e as rotas de cada período, de modo a maximizar o rendimento de cenários semelhantes ao processo de tomada de decisão markoviano. Para a resolução do problema, foi utilizada uma heurística baseada em horizonte rolante, de modo que, em cada passo, o modelo definido em uma janela de tempo foi resolvido de forma ótima pelo resolvidor matemático CPLEX. A eficiência da solução proposta para o primeiro problema foi testada por meio de dois conjuntos de instâncias pequenas e grandes geradas com múltiplos produtos, apresentando resultados de alta qualidade. Porém, para as instâncias grandes foi necessário um alto tempo de processamento. E, para o segundo problema, foram utilizadas instâncias da literatura com apenas um produto, obtendo resultados satisfatórios para instâncias de porte médio se comparado a resultados obtidos por métodos do estado da arte.

No estudo realizado por Moin et al. (2011), o PERV foi abordado como sendo uma rede de distribuição *many-to-one*, consistindo em uma planta de montagem e muitos fornecedores

distintos, onde cada um fornece um produto diferente. Seguindo as restrições de horizonte de tempo finito, multi-períodos, multi-fornecedores e multi-produtos, onde uma frota de veículos homogêneos de capacidade limitada são utilizados para transportarem produtos dos fornecedores para atender a demanda determinada pela planta de montagem em cada período. Como a demanda de cada produto é determinista, foi utilizada uma formulação matemática do problema e utilizado o CPLEX para obter limites inferiores e superiores em uma quantidade de tempo finita, além de um algoritmo genético híbrido, baseado na alocação da primeira rota e segunda estratégia, que considera tanto inventário quanto custos de transporte. Além de um novo conjunto de operadores de *crossover* e mutação, foram adicionadas duas novas representações cromossômicas. Foram realizados diversos testes com diferentes variações de instâncias pequenas e médias, a fim de comprovar a eficiência do método proposto.

Em Azuma et al. (2011), foi apresentada uma proposta de solução para uma formulação mais convencional do PERV, considerando as seguintes restrições: um determinado produto é entregue a partir de um fornecedor atendendo vários clientes em um horizonte de tempo definido e o transporte dos produtos é feito por um veículo com capacidade limitada. Com o objetivo de atingir uma redução simultânea dos custos de transporte e estoque, foi definida uma abordagem multiobjetivo baseada no uso do algoritmo SPEA2 (*Strength Pareto Evolutionary Algorithm 2*), adicionando novidades na representação de soluções candidatas, nos operadores genéticos e de busca local. Tomando por base que a fronteira de Pareto estimada é composta de múltiplas soluções não-dominadas, caracterizando compromissos distintos entre os custos de estoque e transporte, a eficiência da heurística desenvolvida foi comprovada através dos resultados obtidos, considerando instâncias de médio porte da literatura e instâncias de grande porte geradas pelos autores.

No estudo de Solyali e Süral (2011), foi abordado o PERV, onde um fornecedor recebe uma determinada quantidade de um mesmo produto em cada período e o distribui a vários varejistas ao longo de um horizonte de tempo finito utilizando um veículo capacitado. De modo que, cada varejista recebe uma demanda dinâmica externa, além de ser gerenciado pela política determinística *Order-up-to level* de modo a garantir que o fornecedor entregue uma quantidade suficiente para preencher completamente o depósito em cada reabastecimento. O problema tem como objetivo decidir quando e em qual sequência visitar os varejistas, de maneira a minimizar os custos de estoque e roteamento em todo o sistema. Para tal, foram propostos um algoritmo *branch-and-cut* e uma heurística baseada em um *priori tour*, usando uma formulação forte. Ainda de acordo com os autores, esse estudo era o primeiro em sua categoria a considerar uma formulação forte para a parte de reabastecimento de estoque de PERVs. O método proposto teve sua eficiência comprovada ao se comparar os resultados obtidos com os resultados do estado da arte.

No estudo de Archetti et al. (2012), foi definida uma heurística híbrida baseada na busca Tabu com modelos *ad hoc* projetados de programação inteira mista, para o tratamento do PERV,

onde um fornecedor recebe demanda de um conjunto de clientes ao longo de um horizonte de tempo discreto, com restrições de capacidades de estoque para cada cliente sem causar situações de *stock-out*. Para tal, duas políticas se verificam, sendo elas *Order-up-to* e *maximum level policies*, com um único veículo contendo uma determinada capacidade, de modo que os custos de transportes são calculados de acordo com a distância percorrida e o custo de inventário se deve de acordo com o inventário do cliente e do fornecedor. A eficiência do mesmo foi comprovada através de instâncias da literatura.

No estudo realizado por [Coelho e Laporte \(2013\)](#), foi proposto uma solução exata para algumas variantes do problema de estoque com roteamento de veículos, utilizando um algoritmo de *Branch-and-cut*, com foco em multi-veículo considerando frota homogênea e heterogênea e opções de transbordo. Além de propor recursos adicionais ao modelo a fim de garantir uma melhor qualidade em nível de serviços e consistência do modelo. O Algoritmo proposto foi implementado em C++ e CPLEX e avaliado em instâncias da literatura, considerando diferentes configurações, variando número de período, veículos e clientes, onde foram encontrados novas soluções ótimas em algumas instâncias da literatura.

O trabalho de [Curdulino et al. \(2016\)](#), foi proposto uma solução para o problema de Estoque e roteamento de veículos com demanda estocástica aplicados a centros comerciais com um único centro de distribuição. Foi proposto uma variante do algoritmo de [Juan et al. \(2014\)](#) utilizando técnicas de Monte Carlo, sendo que o mesmo teve sua eficiência comprovada em instâncias da literatura apresentando uma pequena melhora em alguns casos considerando a união de políticas granulares com políticas globais.

A Tabela 2.2 descreve o problema solucionado por cada autor citado anteriormente e quais métodos foram aplicados.

Tabela 2.2 – Trabalhos relacionados e características

Abordagens Mono-Objetivo		
Autores	Problema	Solução Proposta
Silveira et al. (1992)	PERV	Busca Tabu com penalidades; Trocas simultâneas;
Archetti et al. (2007)	PERV	Programação linear inteiro misto; <i>Branch-and-cut</i> ;
Shiguemoto et al. (2008)	PERV	Busca Tabu com um método de religamento de caminho; Heurística baseada em horizonte rolante;
Solyali e Süral (2011)	PERV	<i>Branch-and-cut</i> ; Heurística baseada em <i>priori tour</i> usando formulação forte;
Moin et al. (2011)	PERV	Formulação matemática resolvida em CPLEX; Algoritmo genético híbrido;
Archetti et al. (2012)	PERV	Heurística híbrida baseada na busca Tabu com modelos <i>ad hoc</i> projetados de programação inteira mista;
Abordagens Multi-Objetivo		
Azuma et al. (2011)	PERV	SPEA2;

## 3 Desenvolvimento

Este capítulo apresenta a metodologia utilizada para o desenvolvimento desse trabalho. Encontra-se organizado da seguinte forma: a Seção 3.1 apresenta uma breve explicação sobre a classe de complexidade do problema abordado, o método a ser utilizado e, finalmente, define a formulação matemática para o problema.

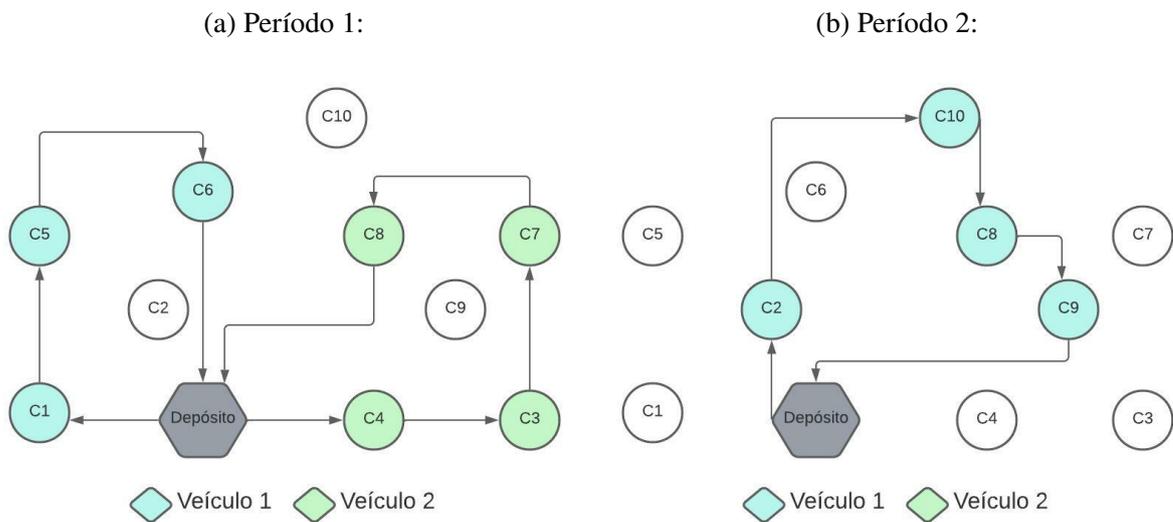
### 3.1 Abordagem da Solução

Como descrito anteriormente, o PERMV é classificado como sendo NP-difícil, pois o mesmo é uma variação do problema de roteamento de veículo capacitado. Portanto, o método apresentado na Seção 2.1.4 é capaz de resolver apenas instâncias pequenas. Por tal motivo, é proposto um método baseado em uma meta-heurística, que tem como objetivo definir a quantidade a ser entregue pelo veículo  $v$  para o cliente  $i$  no período  $p$  (representado pela variável  $x_{vip}$ ), utilizando como base o modelo descrito anteriormente na Seção 2.1.4. O método proposto é dividido em duas fases, sendo que na primeira fase é definido em qual período, veículo e a quantidade a ser entregue a cada um dos clientes, utilizando a política de estoque *Order-to-up* (OU), na segunda fase é realizado o roteamento de veículos utilizando a meta-heurística *Iterated Local Search* (ILS) segundo Lourenço et al. (2001), aplicando o procedimento de Busca local *Randomized Variable Neighborhood Descent* (RVND), definido em Souza (2019), com 3 diferentes estruturas de vizinhança, sendo elas, trocas intrarrotas, inter-rotas e realocação de clientes inter-rotas, não sendo permitido alterar o período e a quantidade a ser entregue a cada um dos clientes.

#### 3.1.1 Representação da solução

Considerou-se a Figura 3.1 como sendo o resultado da fase roteamento de veículos, a qual utiliza a meta-heurística *ILS*, visto que, todas as rotas começam e terminam no depósito, seguindo uma ordem de visita a cada cliente, como indicado no gráfico. A quantidade a ser entregue a cada um dos clientes é definida pela política de estoque *Order-to-up* (OU), que garante que sempre que um determinado cliente é atendido, a capacidade máxima de estoque do mesmo deve ser satisfeita.

Figura 3.1 – Exemplo de solução



A Figura 3.1 apresenta um exemplo de solução considerando 2 veículos e 2 períodos, sendo a Figura 3.1a representa a solução no período 1, onde o Veículo 1 atende os clientes C1, C5 e C6 e o Veículo 2 atende os clientes C4, C3, C7 e C8. A Figura 3.1b representa a solução no período 2 que utiliza somente um veículo para atender os clientes. Neste caso, o Veículo 1 atende os clientes C2, C10, C8, C9. Observe que o cliente C8 é atendido nos dois períodos.

### 3.1.2 Iterated Local Search - ILS

Esta seção apresenta de maneira resumida o algoritmo da fase de roteamento, sendo que o pseudo-código do algoritmo proposto é apresentado abaixo. A meta-heurística recebe uma solução inicial gerada utilizando um método construtivo e então aplica o método da descida (*Randomized Variable Neighborhood Descent (RVND)*) com a estratégia *Best Improvement*, que utiliza as três diferentes estruturas de vizinhança selecionadas de forma aleatória. O laço principal do *ILS* tenta melhorar a solução atual aplicando o método de busca local *RVND* em conjunto com uma série de perturbações na solução corrente. O parâmetro *vezesNivel* define quando aumentar o nível de perturbação, ou seja aumenta o número de movimentos de trocas intrarrotas depois de algumas tentativas sem sucesso, como demonstrado no exemplo da Seção 3.1.4. O parâmetro *MaxIterILS* representa o número máximo de iterações consecutivas sem melhora.

No algoritmo 1 é apresentado um pseudocódigo do algoritmo *ILS* proposto. Nas linhas 1 e 2 do pseudocódigo são inicializadas as variáveis auxiliares. A Linha 3 constrói uma solução inicial gulosa, considerando o cliente mais próximo. A linha 4 aplica o método de busca local *RVND* sobre a solução gulosa encontrada na linha anterior. A iteração do algoritmo que é representado das linhas 5 a 13 termina somente quando o critério de parada é satisfeito, que corresponde ao número máximo de iterações do algoritmo. A linha 6 consiste na fase de perturbação da solução corrente, enquanto a 7 é a fase de busca local na solução corrente. A Linha 8 incrementa o

contador de iterações da heurística. Na Linha 9 verifica se obteve melhora da solução corrente, caso não tenha melhora, na linha 10 aumenta o nível de perturbação, caso tenha melhora, na linha 12 atualiza a solução corrente.

Para mais informações sobre o *ILS*, consultar Lourenço et al. (2001).

---

**Algoritmo 1:** Iterated Local Search - ILS

---

```

1  $it \leftarrow 1$ ;
2  $vezesNivel \leftarrow 1$ ;
3  $Sol_0 \leftarrow ConstroiSolucaoInicial()$ ;
4  $Sol^* \leftarrow RVND(Sol_0)$ ;
5 while  $it \leq MaxIterILS$  do
6    $Sol' \leftarrow Pertubacao(Sol^*, vezesNivel)$ ;
7    $Sol'' \leftarrow RVND(Sol')$ ;
8    $it \leftarrow it + 1$ ;
9   if  $Sol^* \leq Sol''$  then
10     $vezesNivel \leftarrow vezesNivel + 1$ ;
11  else
12     $Sol^* \leftarrow Sol''$ ;
13  end
14 end

```

---

### 3.1.3 Randomized Variable Neighborhood Descent (RVND)

A busca local é um método que tenta melhorar a qualidade da solução corrente por meio de buscas no conjunto de soluções do problema. O método de busca local *RVND* percorre o espaço de soluções realizando trocas de estruturas de vizinhança. O método foi adaptado para selecionar, de forma aleatória, uma estrutura de vizinhança, aplicando-a sobre a solução corrente, utilizando a estratégia *Best Improvement*. Caso encontre melhora, atualiza a solução corrente e sorteia uma nova estrutura de vizinhança. Caso contrário, bloqueia a estrutura de vizinhança que foi executada e sorteia uma nova. Se a nova estrutura de vizinhança melhorar a solução corrente, as estruturas que se encontram bloqueadas ficam disponíveis novamente e o processo se repete. O algoritmo termina após utilizar todas as estruturas de vizinhança sem conseguir melhorar a solução corrente, ou seja, quando todas as estruturas de vizinhança estiverem bloqueadas ou atingir o número máximo de iterações permitidas.

No algoritmo 2 é apresentado um pseudocódigo do algoritmo *RVND*. Nas linhas 1 e 2 do pseudocódigo são inicializadas as variáveis auxiliares. A iteração do algoritmo que é representado das linhas 3 a 14 termina somente quando o critério de parada é satisfeito, que corresponde ao número máximo de iterações do algoritmo ou quando todas as estruturas de vizinhança estão indisponíveis. A linha 4 seleciona aleatoriamente uma estrutura de vizinhança. A linha 5 verifica se a estrutura de vizinhança sorteada na linha anterior está disponível, em caso positivo, na linha 6 executa a vizinhança sorteada anteriormente. A linha 7 verifica se obteve melhora na solução

corrente, caso tenha melhora, na linha 8 atualiza a solução corrente e na linha 9 deixa todas as estruturas de vizinhanças disponíveis novamente. Em caso de piora da solução corrente, na linha 11 deixa a estrutura de vizinhança sorteada anteriormente temporariamente indisponível. A linha 12 incrementa o contador do método.

Para mais informações sobre a Busca Local RVND, consultar Souza (2019).

---

**Algoritmo 2:** Randomized Variable Neighborhood Descent (RVND)

---

```

1  bloqueado = [];
2  k ← 1;
3  while k ≤ kmax do
4    estrutura ← SorteiaVizinhanca();
5    if estrutura not in bloqueado then
6      Sol' ← ExecutavizinhancaSorteada(Sol, estrutura);
7      if Sol' ≤ Sol then
8        Sol ← Sol';
9        bloqueado.clear();
10     else
11       bloqueado.append(estrutura);
12       k = k + 1;
13     end
14   end
15 end
16

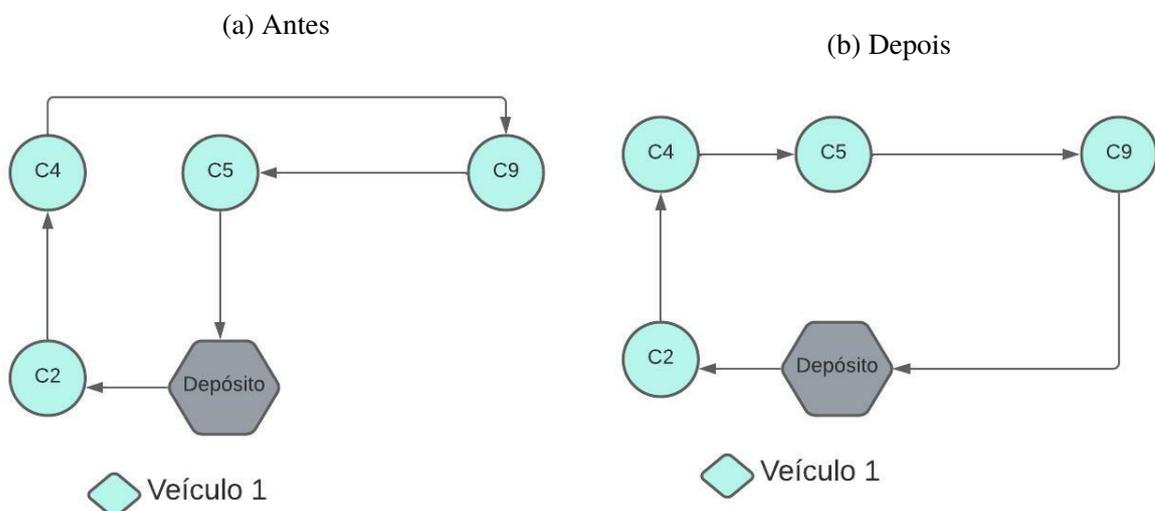
```

---

O método *RVND* utiliza as vizinhanças descritas a seguir:

**Trocas intrarota:** Realiza trocas entre todos os clientes possíveis dentro de uma mesma rota, como demonstrado na Figura 3.2.

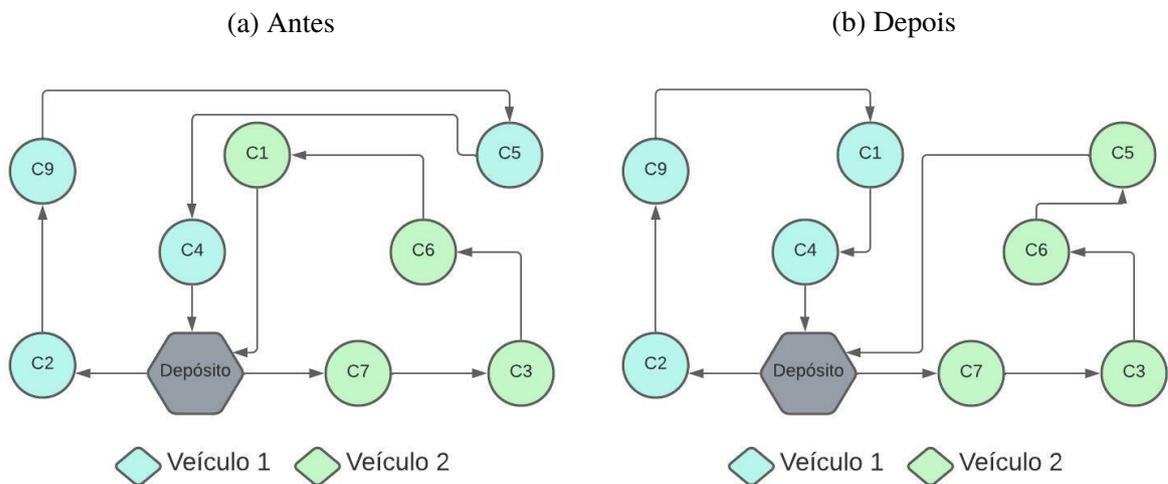
Figura 3.2 – Exemplo de aplicação de vizinhança intrarrotas



A Figura 3.2 apresenta um exemplo de aplicação da estrutura de vizinhança de trocas intrarota, onde a Figura 3.2a apresenta a solução original, onde o Veículo 1 atende os clientes C2, C4, C9 e C5 respectivamente nessa ordem. A Figura 3.2b apresenta a aplicação da estrutura de vizinhança de trocas intrarota, considerando a troca da ordem de visita do cliente C9 pelo cliente C5 realizado pelo Veículo 1.

**Trocas inter-rotas:** Realiza trocas entre todos os clientes possíveis entre duas rotas de um mesmo período com veículos diferentes, como demonstrado na Figura 3.3.

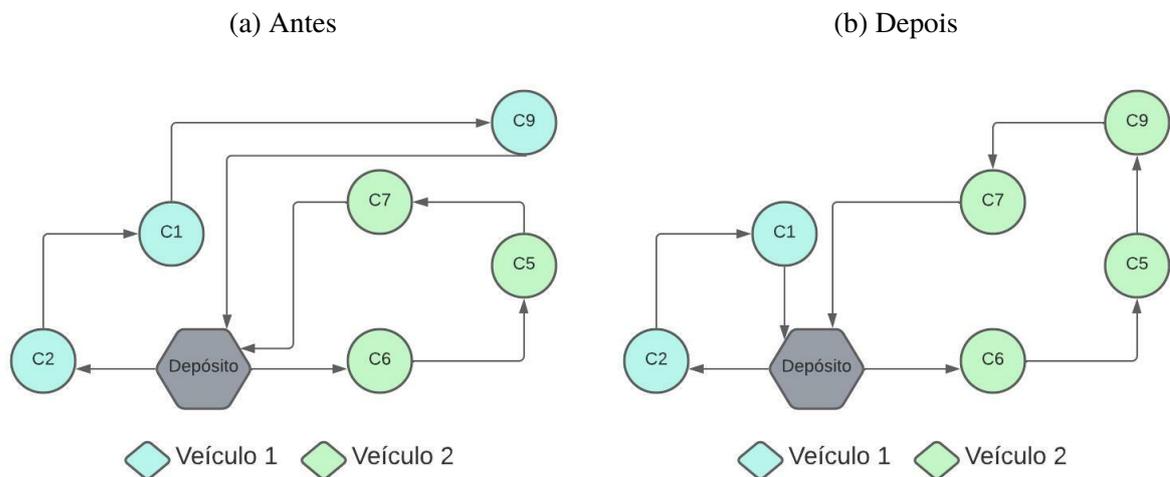
Figura 3.3 – Exemplo de aplicação de vizinhança inter-rote



A Figura 3.3 apresenta um exemplo de aplicação da estrutura de vizinhança de trocas inter-rote, onde a Figura 3.3a apresenta a solução original, onde o Veículo 1 atende os clientes C2, C9, C5 e C4 e o Veículo 2 atende os clientes C7, C3, C6 e C1. A Figura 3.3b apresenta a aplicação da estrutura de vizinhança de trocas inter-rote, considerando a troca do cliente C1 do veículo 1 pelo cliente C5 do veículo 2, alterando a ordem de visita de ambos. Portanto, o Veículo 1 passa a atender em ordem de visita os clientes C2, C9, C1 e C4 e o Veículo 2 os clientes C7, C3, C6 e C5.

**Realocação de clientes inter-rotas:** Realiza a exclusão de um cliente em uma rota e realocação do mesmo em outra rota de mesmo período com veículo diferente na melhor posição possível após realizar todas comparações necessárias, como demonstrado na Figura 3.4.

Figura 3.4 – Exemplo de aplicação de vizinhança de realocação de clientes inter-rotas



A Figura 3.4 apresenta um exemplo de aplicação da estrutura de vizinhança de realocação de clientes inter-rotas, onde a Figura 3.4a apresenta a solução original onde o Veículo 1 atende os clientes C2, C1 e C9 e o Veículo 2 atende os clientes C6, C5 e C7. A Figura 3.4b apresenta a aplicação da estrutura de vizinhança de realocação de clientes inter-rotas, considerando a remoção do cliente C9 da rota do veículo 1 e a realocação do mesmo na rota do veículo 2 de forma otimizada. Portanto, o Veículo 1 passa a atender em ordem de visita os clientes C2 e C1 e o Veículo 2 passa a atender os clientes C6, C5, C9 e C7.

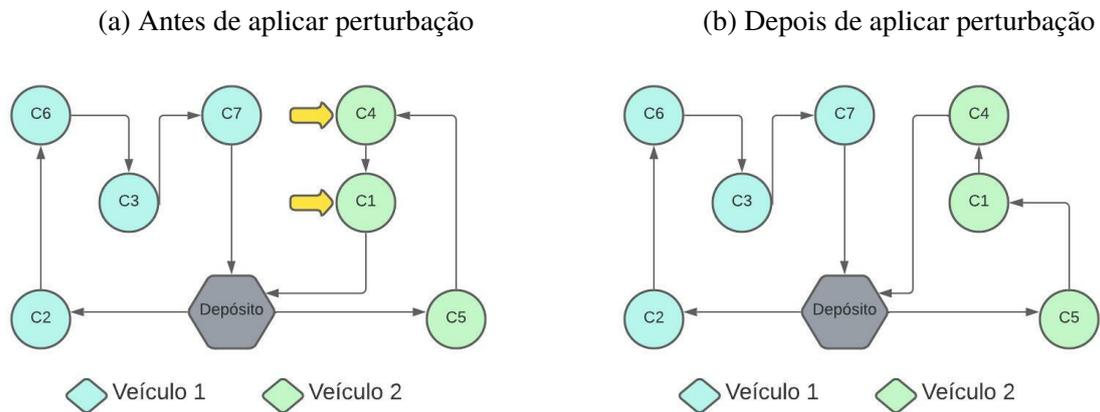
### 3.1.4 Perturbação

Aplicar uma perturbação consiste em modificar a solução corrente com o intuito de explorar outras regiões do espaço de busca.

Na fase de perturbação foi utilizado a seguinte estrutura de vizinhança:

- **Trocas intrarota:** Seleciona aleatoriamente um veículo e um período dentro do conjunto rotas. Caso haja mais de um cliente dentro da rota, seleciona aleatoriamente dois clientes diferentes e então altera a ordem de visita a ser realizada pelo veículo selecionado, como demonstrado no exemplo das Figura 3.5a e Figura 3.5b.

Figura 3.5 – Exemplo de execução do método de Perturbação



A Figura 3.5 apresenta um exemplo de aplicação do método de perturbação descrito anteriormente, sendo a Figura 3.5a a solução antes de aplicar o método de perturbação, onde o Veículo 1 atende os clientes C2, C6, C3 e C7 e o Veículo 2 atende os clientes C5, C4 e C1. Sendo que os clientes C1 e C4 do Veículo 2 foram selecionados aleatoriamente para aplicação do método. A Figura 3.5b apresenta a solução após a aplicação do método de perturbação, onde os clientes C1 e C4 do Veículo 2 tiveram sua ordem de visita alterada. Portanto, a ordem de visita dos clientes do Veículo 2 ficaram da seguinte forma C5, C1 e C4.

## 4 Resultados

Este capítulo apresenta o cenário de teste utilizado e os resultados alcançados. Encontrase organizado da seguinte forma: a Seção 4.1 apresenta o cenário de teste e as instâncias utilizadas e a Seção 4.2 apresenta os resultados obtidos utilizando o método proposto na Seção 3.

### 4.1 Cenário de Testes

O método proposto no Capítulo 3 foi implementado utilizando Python 3.6. Os testes foram realizados utilizando um *desktop* com processador Intel Core i5-10400F CPU 2.90 GHz, 8GB de memória e sistema operacional *Windows 10* 64 bits.

O método tem sua eficiência testada a partir de instâncias para múltiplos veículos propostas por Coelho (2013), que foram modificadas a partir das instâncias de um único veículo propostas por Archetti et al. (2012). A coleção é composta por 60 instâncias, sendo dividida em três conjuntos com 20 instâncias cada e com diferentes quantidades de clientes, sendo 50, 100 e 200 clientes, respectivamente. As mesmas, são apresentadas como *large-n-low* ou *large-n-high*, onde *low* e *high* representam, respectivamente, baixo e alto custo de estoque. O método foi avaliado com 6 períodos e o número de veículos como sendo 2. O algoritmo proposto foi executado 10 vezes para cada uma das instâncias, realizando a média aritmética dos resultados obtidos. Foram utilizados os seguintes parâmetros para teste de eficiência do algoritmo proposto:

- *vezesNivel*: Inicia com valor de um único movimento de troca intrarota e após algumas iterações sem melhora, incrementa em uma unidade a quantidade de trocas à ser aplicada, assim sucessivamente até que se obtenha melhora e então retorna ao valor inicial ou chegue ao fim da execução do método.
- *MaxIterILS*: representa o número máximo de perturbações consecutivas sem melhora, sendo definido por:
- $\text{MaxIterILS} = (\text{Número de Clientes} / k) * \text{Número de Períodos} * \text{Número de Veículos}$   
Onde *k*, é definido baseado no número de clientes:  
Até 100 clientes:  
 $k = \text{Numero de clientes} * 0,1;$   
Acima de 100 clientes:  
 $k = \text{Número de clientes} * 0,2;$
- *kmax*: Número máximo de iterações do método de busca local RVND, definido como:  
 $kmax = (\text{Número de Clientes} / k) * \text{Número de Períodos} * \text{Número de Veículos}$

Como não foi utilizado nenhuma ferramenta de calibração de parâmetros, os valores de  $k$ , apresentados anteriormente, foram definidos realizando testes empíricos até encontrar um resultado de boa qualidade.

## 4.2 Resultados Obtidos

A [Tabela 4.1](#) representa a média dos resultados obtidos aplicando o método proposto na [Seção 3](#), considerando o cenário de teste apresentado na [Seção 4.1](#). Os resultados obtidos foram comparados com o trabalho de [Coelho \(2013\)](#).

As colunas da [Tabela 4.1](#), representam respectivamente, **Tipo**: como sendo a classe da instância de soluções podendo ser de alto ou baixo custo, **N**: número de clientes, **Coelho**: resultados obtidos no trabalho de [Coelho \(2013\)](#), **ILS**: método proposto nesse trabalho para resolução do problema, **Tempo(s)**: tempo de execução do método proposto e **GAP**: cálculo do GAP entre o método proposto e os resultados obtidos por [Coelho \(2013\)](#), considerando a seguinte definição  $GAP = (\text{Método proposto} - \text{Coelho})/\text{Coelho}$ .

Tabela 4.1 – Comparativo de resultados utilizando 2 veículos

<b>Tipo</b>	<b>N</b>	<b>Coelho</b>	<b>ILS</b>	<b>Tempo (s)</b>	<b>Gap(%)</b>
<b>Low</b>	50	13049,9	15480,11	29,52	18,62
	100	25546,1	28039,86	308,87	9,76
	200	46524,7	54725,84	1737,74	17,63
	Média	28373,6	32748,6	692,04	15,47
<b>High</b>	50	32585,8	35433,63	29,74	8,73
	100	60773,1	65372,83	230,1	7,57
	200	121982,4	136525,4	2404,12	11,92
	Média	71780,4	79110,63	887,98	10,21

O método proposto na [Seção 3](#) obteve resultados inferiores se comparado ao método apresentado por [Coelho \(2013\)](#). Considerando que foi obtido um GAP mínimo de 9,76% e máximo de 18,62%, chegando em um GAP médio de 15,47% para as instâncias de baixo custo (*Low*) com tempo de processamento aceitável. Porém, o método proposto apresentou maior eficiência para instâncias de alto custo (*High*), chegando a um GAP médio de 10,21%, considerando que o GAP mínimo e máximo foi respectivamente de 7,57% e 11,92%, com um tempo de processamento um pouco pior se comparado as instâncias de baixo custo.

Em relação ao tempo computacional, este variou entre 29,52 e 1737,74 segundos para instâncias de baixo custo, chegando a um tempo médio de execução de 692,04 segundos. Para as instâncias de alto custo, o tempo de processamento médio aumentou para 887,98 segundos, com o tempo de processamento mínimo e máximo variando entre 29,74 e 2404,12 segundos.

As [Tabelas 4.2](#) e [4.3](#) apresentam os resultados individuais do conjunto de instâncias de baixo e alto custo respectivamente.

As colunas das Tabelas 4.2 e 4.3, representam respectivamente, **Tipo**: como sendo a classe da instância de soluções podendo ser de alto ou baixo custo, **N**: número de clientes, **Instância**: nome individual da instância, **ILS**: método proposto nesse trabalho para resolução do problema, **Tempo(s)**: tempo de execução do método proposto.

Tabela 4.2 – Resultados completo instâncias de baixo custo com 2 veículos

<b>Tipo</b>	<b>N</b>	<b>Instância</b>	<b>ILS</b>	<b>Tempo (s)</b>
Low	50	absH6low1n50	15944,8	28,9
		absH6low2n50	14270,5	30,85
		absH6low3n50	15607,14	29,5
		absH6low4n50	16599,64	26,62
		absH6low5n50	15722,7	32,08
		absH6low6n50	13705,05	31,78
		absH6low7n50	14435,17	27,56
		absH6low8n50	15933,15	32,09
		absH6low9n50	16116,93	29,86
		absH6low10n50	16465,98	25,97
				<b>Média</b>
Low	100	absH6low1n100	29432,76	298,32
		absH6low2n100	26397,8	309,94
		absH6low3n100	27778,92	261,8
		absH6low4n100	27443,91	360,98
		absH6low5n100	26792,52	320,1
		absH6low6n100	28462,47	310,29
		absH6low7n100	29547,36	346,57
		absH6low8n100	27632,21	280,28
		absH6low9n100	29491,05	290,32
		absH6low10n100	27419,63	310,17
				<b>Média</b>
Low	200	absH6low1n200	57775,03	1745,92
		absH6low2n200	51139,53	1667,76
		absH6low3n200	52228,68	1565,22
		absH6low4n200	58578,22	1479,54
		absH6low5n200	52139,91	1752,14
		absH6low6n200	53107,36	1587,69
		absH6low7n200	56571,44	1615,34
		absH6low8n200	55362,84	1796,77
		absH6low9n200	54187,23	1932,21
		absH6low10n200	56168,17	2234,85
				<b>Média</b>

Tabela 4.3 – Resultados completo instâncias de alto custo com 2 veículos

<b>Tipo</b>	<b>N</b>	<b>Instância</b>	<b>ILS</b>	<b>Tempo (s)</b>
High	50	absH6high1n50	36240,99	29,2
		absH6high2n50	35530,2	31,06
		absH6high3n50	35300,88	29,78
		absH6high4n50	34853,93	26,73
		absH6high5n50	35167,57	32,57
		absH6high6n50	35519,99	31,46
		absH6high7n50	35184,29	27,86
		absH6high8n50	34021,87	33,16
		absH6high9n50	36083,75	29,69
		absH6high10n50	36432,83	25,9
		<b>Média</b>		35433,63
High	100	absH6high1n100	64435,68	221,2
		absH6high2n100	64051,96	228,3
		absH6high3n100	65898,35	211,52
		absH6high4n100	67595,35	215,16
		absH6high5n100	65008,71	236,59
		absH6high6n100	66410,44	216,32
		absH6high7n100	64129,42	237,3
		absH6high8n100	65657,83	222,51
		absH6high9n100	65817,34	266,7
		absH6high10n100	64723,25	245,4
		<b>Média</b>		65372,83
High	200	absH6high1n200	148340,63	2530,22
		absH6high2n200	138960,46	2942,81
		absH6high3n200	133523,2	2165,72
		absH6high4n200	129329,37	2334,35
		absH6high5n200	134220,22	1959,27
		absH6high6n200	133873,16	2262,38
		absH6high7n200	139217,46	2067,34
		absH6high8n200	132298,11	2611,77
		absH6high9n200	137354,9	2414,53
		absH6high10n200	138136,7	2752,87
		<b>Média</b>		136525,421

## 5 Conclusões

O trabalho desenvolvido apresentou uma solução para o Problema de Estoque e Roteamento de Múltiplos Veículos (PERMV) considerando tempo finito, demanda determinística, estrutura Um-Para-Muitos (sendo um único depósito e vários clientes), política de estoque *Order-to-up* (OU), frota homogênea e decisão de estoque Não-negativo.

Foi utilizado a meta-heurística *ILS*, aplicando o método de Busca local *RVND* com três diferentes estruturas de vizinhança, sendo elas trocas inter-rotas, trocas intrarrotas e realocação inter-rotas, considerando a estratégia *Best Improvement*.

Para avaliação do algoritmo desenvolvido, foram utilizadas instâncias da literatura propostas por Coelho (2013), como apresentado anteriormente na Seção 4.2, porém, não foram obtidos resultados superiores aos já existentes na literatura. O método proposto, apresentou melhores resultados para instâncias de alto custo (*High*), com um GAP médio de 10,21%, porém com um maior tempo de processamento. Para instâncias de baixo custo (*Low*) o GAP médio ficou em 15,47% com um tempo de processamento melhor, se comparado a instâncias de alto custo.

Com o objetivo de evoluir nos resultados obtidos, em trabalhos futuros recomenda-se testar o método utilizando a política de estoque *Maximum Level* (ML), permitindo alteração da quantidade entregue a cada cliente e novas estruturas de vizinhança permitindo a troca de clientes em períodos diferentes. Além de novas estruturas de vizinhança na aplicação do método de perturbação.

# Referências

- ALMEIDA, M. R. de. Heurística híbrida para um problema de estoque com roteamento de múltiplos veículos. *Simpósio Brasileiro de Pesquisa Operacional*, p. 1588–1599, 2014.
- ANDERSSON, H.; HOFF, A.; CHRISTIANSEN, M.; HASLE, G.; LØKKETANGEN, A. Industrial aspects and literature survey: Combined inventory management and routing. *Computers & Operations Research*, Elsevier, v. 37, n. 9, p. 1515–1536, 2010.
- ARCHETTI, C.; BERTAZZI, L.; HERTZ, A.; SPERANZA, M. G. A hybrid heuristic for an inventory routing problem. *INFORMS Journal on Computing*, INFORMS, v. 24, n. 1, p. 101–116, 2012.
- ARCHETTI, C.; BERTAZZI, L.; LAPORTE, G.; SPERANZA, M. G. A branch-and-cut algorithm for a vendor-managed inventory-routing problem. *Transportation science*, INFORMS, v. 41, n. 3, p. 382–391, 2007.
- AZUMA, R. M. et al. *Otimização multiobjetivo em problema de estoque e roteamento gerenciados pelo fornecedor*. Dissertação (Mestrado) — Universidade Estadual de Campinas, Faculdade de Engenharia Elétrica e de Computação, 2011.
- BELFIORE, P. P.; COSTA, O. L. d. V.; FÁVERO, L. P. L. Inventory routing problem: bibliographic review. *Production*, SciELO Brasil, v. 16, p. 442–454, 2006.
- BELFIORE, P. P.; COSTA, O. L. d. V.; FÁVERO, L. P. L. Problema de estoque e roteirização: revisão bibliográfica. *Production*, SciELO Brasil, v. 16, n. 3, p. 442–454, 2006.
- BELL, W. J.; DALBERTO, L. M.; FISHER, M. L.; GREENFIELD, A. J.; JAIKUMAR, R.; KEDIA, P.; MACK, R. G.; PRUTZMAN, P. J. Improving the distribution of industrial gases with an on-line computerized routing and scheduling optimizer. *Interfaces*, INFORMS, v. 13, n. 6, p. 4–23, 1983.
- BERTAZZI, L.; PALETTA, G.; SPERANZA, M. G. Deterministic order-up-to level policies in an inventory routing problem. *Transportation Science*, INFORMS, v. 36, n. 1, p. 119–132, 2002.
- COELHO, L. C. *Flexibility and consistency in inventory-routing*. 1–297 p. Tese (Doutorado) — HEC Montréal, 2013.
- COELHO, L. C.; CORDEAU, J.-F.; LAPORTE, G. Thirty years of inventory routing. *Transportation Science*, INFORMS, v. 48, n. 1, p. 1–19, 2013.
- COELHO, L. C.; LAPORTE, G. The exact solution of several classes of inventory-routing problems. *Computers & Operations Research*, Elsevier, v. 40, n. 2, p. 558–565, 2013.
- CURDULINO, R.; ALVES, P. Y. A. L.; DELGADO, K. V. Sistema para resolver o problema de roteamento de estoque baseado em técnicas de monte carlo. In: SBC. *Anais do XII Simpósio Brasileiro de Sistemas de Informação*. [S.l.], 2016. p. 369–376.
- FILHO, H. Z.; FIGUEIREDO, L.; JUNIOR, I. M. Os custos logísticos de uma distribuidora de alimentos não perecíveis. *XXIII Encontro Nacional de Engenharia de Produção-Ouro Preto, MG, Brasil*, v. 21, 2003.

JUAN, A. A.; GRASMAN, S. E.; CACERES-CRUZ, J.; BEKTAŞ, T. A simheuristic algorithm for the single-period stochastic inventory-routing problem with stock-outs. *Simulation Modelling Practice and Theory*, Elsevier, v. 46, p. 40–52, 2014.

KARP, R. M. On the computational complexity of combinatorial problems. *Networks*, Wiley Online Library, v. 5, n. 1, p. 45–68, 1975.

LARSEN, A.; MADSEN, O. B. The dynamic vehicle routing problem. Technical University of Denmark (DTU), 2000.

LOURENÇO, H. R.; MARTIN, O.; STÜTZLE, T. A beginner's introduction to iterated local search. In: *Proceedings of MIC*. [S.l.: s.n.], 2001. v. 4, p. 1–6.

MOIN, N. H.; SALHI, S.; AZIZ, N. An efficient hybrid genetic algorithm for the multi-product multi-period inventory routing problem. *International Journal of Production Economics*, Elsevier, v. 133, n. 1, p. 334–343, 2011.

SHIGUEMOTO, A. L. et al. *Heurísticas para o problema de estoque e roteamento de veículos*. Dissertação (Mestrado) — Universidade Estadual de Campinas, Faculdade de Engenharia Elétrica e de Computação, 2004.

SHIGUEMOTO, A. L. et al. *Métodos heurísticos para resolução de problemas integrados de produção, estoque e distribuição*. i–xii, 1–84 p. Tese (Doutorado) — Universidade Estadual de Campinas, Faculdade de Engenharia Elétrica e de Computação, 2008.

SILVEIRA, P. D. B. d. et al. *Algoritmos para a resolução do problema de estoque e roteamento de veículos*. Dissertação (Mestrado) — Universidade Estadual de Campinas, Faculdade de Engenharia Elétrica e de Computação, 1992.

SOLYALİ, O.; SÜRAL, H. A branch-and-cut algorithm using a strong formulation and an a priori tour-based heuristic for an inventory-routing problem. *Transportation Science*, INFORMS, v. 45, n. 3, p. 335–345, 2011.

SOUZA, A. L. d. S. *Algoritmos meta-heurísticos para o problema dial-a-ride*. Dissertação (Mestrado em Ciência da Computação) — Universidade Federal de Ouro Preto, 2019.