

UNIVERSIDADE FEDERAL DE OURO PRETO
ESCOLA DE MINAS - EM
DEPARTAMENTO DE ENGENHARIA DE PRODUÇÃO

LUCAS FERNANDES VASCONCELOS
Orientador: Prof. Dr. Helton Cristiano Gomes
Coorientador: Prof. Dr. Aloísio de Castro Gomes Júnior

**RESOLUÇÃO DO PROBLEMA DE ROTEAMENTO DE VEÍCULOS
COM JANELAS DE TEMPO E INCERTEZA NOS TEMPOS DE
VIAGEM E SERVIÇO POR MEIO DE UM ALGORITMO
SIMHEURÍSTICO**

Ouro Preto, MG
2022

UNIVERSIDADE FEDERAL DE OURO PRETO
ESCOLA DE MINAS - EM
DEPARTAMENTO DE ENGENHARIA DE PRODUÇÃO

LUCAS FERNANDES VASCONCELOS

**RESOLUÇÃO DO PROBLEMA DE ROTEAMENTO DE VEÍCULOS COM JANELAS
DE TEMPO E INCERTEZA NOS TEMPOS DE VIAGEM E SERVIÇO POR MEIO DE
UM ALGORITMO SIMHEURÍSTICO**

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

Orientador: Prof. Dr. Helton Cristiano Gomes

Coorientador: Prof. Dr. Aloísio de Castro Gomes Júnior

Ouro Preto, MG
2022

SISBIN - SISTEMA DE BIBLIOTECAS E INFORMAÇÃO

V331r Vasconcelos, Lucas Fernandes.

Resolução do problema de roteamento de veículos com janelas de tempo e incerteza nos tempos de viagem e serviço por meio de um algoritmo simheurístico. [manuscrito] / Lucas Fernandes Vasconcelos. - 2022.

39 f.: il.: color., tab..

Orientador: Prof. Dr. Helton Gomes.

Coorientador: Prof. Dr. Aloísio Júnior.

Monografia (Bacharelado). Universidade Federal de Ouro Preto. Escola de Minas. Graduação em Engenharia de Produção .

1. Pesquisa operacional. 2. Otimização combinatória. 3. Veículos - Rotas. 4. Programação heurística. 5. Parâmetros estocásticos. I. Gomes, Helton. II. Júnior, Aloísio. III. Universidade Federal de Ouro Preto. IV. Título.

CDU 658.5

Bibliotecário(a) Responsável: Maristela Sanches Lima Mesquita - CRB-1716



FOLHA DE APROVAÇÃO

Lucas Fernandes Vasconcelos

Resolução do Problema de Roteamento de Veículos com Janelas de Tempo e Incerteza nos Tempos de Viagem e Serviço por Meio de um Algoritmo Simheurístico

Monografia apresentada ao Curso de Engenharia de Produção da Universidade Federal de Ouro Preto como requisito parcial para obtenção do título de Engenheiro de Produção.

Aprovada em 05 de maio de 2022.

Membros da Banca

Doutor - Helton Cristiano Gomes - Orientador - Universidade Federal de Ouro Preto
Doutor - Aloisio de Castro Gomes Júnior - Universidade Federal de Ouro Preto
Mestrando - Ruan Carlos Silva Menezes Pinheiro - Universidade Federal de Ouro Preto

Helton Cristiano Gomes, orientador do trabalho, aprovou a versão final e autorizou seu depósito na Biblioteca Digital de Trabalhos de Conclusão de Curso da UFOP em 09/05/2022.



Documento assinado eletronicamente por **Helton Cristiano Gomes, PROFESSOR DE MAGISTERIO SUPERIOR**, em 05/05/2022, às 20:16, 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, informando o código verificador **0320594** e o código CRC **CE504D43**.

Agradecimentos

Agradeço, primeiramente, aos meus pais, Adalgisa e José Aloizio, pelo apoio incondicional e por estarem ao meu lado não só durante todos esses anos de universidade, mas em toda minha trajetória até aqui. Sem vocês nada disso seria possível.

Aos meus familiares, que me auxiliaram nas diversas fases da minha vida e que me motivaram e inspiraram na tarefa de me tornar a pessoa que sou hoje.

Agradeço também à Sabrina e aos amigos João, Davi Ferreira, Davi Moraes e Cacau, que, independente da distância, sempre estiveram presentes nos momentos bons e nos momentos ruins da minha vida, me aconselhando e me apoiando.

Aos amigos do grupo dos Canelas, que me sempre me proporcionaram boas conversas e momentos de descontração, alegrias e risadas, mesmo estando longe.

À grandiosa República NosTravamus e à todos que fazem parte dessa incrível família. Com vocês tive a oportunidade de viver momentos incríveis, que jamais esquecerei. Sem vocês talvez Ouro Preto nunca tivesse se tornado uma segunda casa para mim.

Por fim, agradeço à Universidade Federal de Ouro Preto e à todos seus professores, que tiveram a capacidade de transmitir o conhecimento necessário para me preparar para o mercado de trabalho e, principalmente, para a vida.

"Mas não precisamos saber para onde vamos, nós só precisamos ir."

— trecho da música Infinita Highway - Engenheiros do Hawaii

Resumo

Uma das formas de adotar processos logísticos mais ágeis e adaptados aos perfis dos clientes é a otimização da utilização de veículos e de suas rotas de entrega, objetivando a minimização dos custos envolvidos na operação. Na literatura, este problema corresponde ao Problema de Roteamento de Veículos (PRV). Para tratar complicações realistas e generalizações do modelo básico de roteamento, surge o Problema de Roteamento de Veículos com Janelas de Tempo (PRVJT), onde os clientes devem ser atendidos dentro de um certo período de tempo. Em cenários reais, o tempo de viagem e serviço das rotas pode ser afetado por diversos aspectos, como a má qualidade das vias e condições meteorológicas, o que gera uma incerteza intrínseca aos problemas deste tipo. O presente trabalho tem como objetivo o desenvolvimento de um algoritmo baseado nas chamadas *simheuristics*, de forma a combinar as meta-heurísticas ILS-VNS com as Simulações de Monte-Carlo para encontrar boas soluções para o PRVJT com incerteza nos tempos de viagem e serviço em tempo computacional aceitável. Em um primeiro momento o problema é resolvido em sua forma determinística e, posteriormente, os tempos determinísticos são substituídos por amostras aleatórias de uma função densidade de probabilidade pré-determinada. Por fim, são avaliados os impactos da presença da incerteza nas soluções encontradas. O método proposto foi validado através da comparação dos resultados obtidos com resultados encontrados na literatura para instâncias de pequeno porte, onde a diferença entre os resultados foi mínima ou nenhuma. O algoritmo se mostrou eficiente para resolução de instâncias com clientes distribuídos aleatoriamente e para instâncias com clientes agrupados. Entretanto, os resultados foram consideravelmente piores para instâncias que possuem essas duas características em conjunto.

Palavras-chave: Pesquisa operacional. Otimização combinatória. Roteamento de veículos. Meta-heurísticas. Parâmetros estocásticos.

Abstract

One of the ways to adopt more agile logistics processes and adapted to customer profiles is to optimize the use of vehicles and their delivery routes, aiming to minimize of the costs involved in the operation. In the literature, this problem corresponds to the Vehicle Routing Problem (VRP). To treat realistic complications and generalizations from the basic routing model, the Vehicle Routing Problem with Windows of Time (VRPWT), where customers must be served within a certain period of time. In real scenarios, the travel and service time of the routes can be affected by several aspects, such as the poor quality of the roads and meteorological conditions, which generates an intrinsic uncertainty to the problems of this kind. The present work aims to develop an algorithm based on the so-called simheuristics, in order to combine the ILS-VNS metaheuristics with the Monte-Carlo Simulations to find good solutions for the VRPWT with uncertainty in the travel times and service in acceptable computational time. At first, the problem is solved in its deterministic form and, later, the deterministic times are replaced by random samples of a predetermined probability density function. Finally, the impacts of the presence of uncertainty in the solutions found are evaluated. The method proposed was validated by comparing the results obtained with results found in the literature for small instances, where the difference between the results was minimal or none. The algorithm proved to be efficient for resolving instances with randomly distributed clients and for instances with clustered clients. However, the results were considerably worse for instances that have these two characteristics together.

Keywords: Operational Research. Combinatorial optimization. Vehicle routing. Metaheuristics. Stochastic parameters.

Lista de Ilustrações

Figura 3.1 – Exemplo de solução candidata com três veículos e doze clientes.	14
Figura 3.2 – Movimento 1: configuração inicial	16
Figura 3.3 – Movimento 1: configuração após movimento	16
Figura 3.4 – Movimento 2: configuração inicial	16
Figura 3.5 – Movimento 2: configuração após movimento	16
Figura 3.6 – Movimento 3: configuração inicial	16
Figura 3.7 – Movimento 3: configuração após movimento	17
Figura 3.8 – Movimento 4: configuração inicial	17
Figura 3.9 – Movimento 4: configuração após movimento	17
Figura 3.10–Perturbação: configuração inicial	18
Figura 3.11–Perturbação: configuração após movimento	19
Figura 4.1 – <i>Boxplot</i> do GAP entre os resultados obtidos e resultados da literatura para as instâncias de 25 clientes	31
Figura 4.2 – <i>Boxplot</i> do GAP entre os resultados obtidos e resultados da literatura para as instâncias de 50 clientes	32
Figura 4.3 – <i>Boxplot</i> da diferença entre os veículos utilizados nas soluções obtidas e nas soluções da literatura para instâncias de 25 clientes	32
Figura 4.4 – <i>Boxplot</i> da diferença entre os veículos utilizados nas soluções obtidas e nas soluções da literatura para instâncias de 50 clientes	33
Figura 4.5 – <i>Boxplot</i> do GAP entre os resultados obtidos durante a fase simulações e os resultados obtidos durante a fase de otimização para as instâncias de 25 clientes	34
Figura 4.6 – <i>Boxplot</i> do GAP entre os resultados obtidos durante a fase simulações e os resultados obtidos durante a fase de otimização para as instâncias de 50 clientes	35

Lista de Tabelas

Tabela 4.1 – Resultados determinísticos para os problemas do tipo R1 e R2 com 25 clientes	25
Tabela 4.2 – Resultados determinísticos para os problemas do tipo C1 e C2 com 25 clientes	25
Tabela 4.3 – Resultados determinísticos para os problemas do tipo RC1 e RC2 com 25 clientes	26
Tabela 4.4 – Resultados da simulação para os problemas do tipo R1 e R2 com 25 clientes	27
Tabela 4.5 – Resultados da simulação para os problemas do tipo C1 e C2 com 25 clientes	27
Tabela 4.6 – Resultados da simulação para os problemas do tipo RC1 e RC2 com 25 clientes	28
Tabela 4.7 – Resultados determinísticos para os problemas R1, R2, C1, C2, RC1 e RC2 com 50 clientes	29
Tabela 4.8 – Resultados da simulação para os problemas R1, R2, C1, C2, RC1 e RC2 com 50 clientes	30

Lista de Abreviaturas e Siglas

EM	Escola de Minas
UFOP	Universidade Federal de Ouro Preto
PRV	Problema de Roteamento de Veículos
PRVE	Problema de Roteamento de Veículos Estocástico
PRVJT	Problema de Roteamento de Veículos com Janelas de Tempo
ILS	<i>Iterated Local Search</i>
VNS	<i>Variable Neighborhood Search</i>
CCP	<i>Chance Constrained Program</i>
SPR	<i>Stochastic Program With Recourse</i>
FA	Função de Avaliação

Sumário

1	Introdução	1
2	Referencial Teórico	3
2.1	Logística e otimização combinatória	3
2.2	Problema de Roteamento de Veículos	4
2.3	Problema de Roteamento de Veículos com Janelas de Tempo	5
2.4	Problema de Roteamento de Veículos Estocástico	6
2.5	Meta-heurísticas	7
2.6	Iterated Local Search	8
2.7	Variable Neighborhood Search	10
2.8	Método de Monte Carlo	10
2.9	Simheuristics	11
3	Metodologia	13
3.1	Implementação do algoritmo ILS-VNS	13
3.1.1	Leitura dos dados	13
3.1.2	Representação de uma solução	14
3.1.3	Cálculo da função de avaliação	14
3.1.4	Geração da solução inicial	15
3.1.5	Busca local	15
3.1.6	Funcionamento do ILS	18
3.1.7	Critério de parada	18
3.1.8	Perturbações e critério de aceitação	18
3.2	Simulações de Monte Carlo	19
3.3	Integração entre ILS-VNS e simulações	21
4	Resultados	24
4.1	Análise dos resultados obtidos	31
5	Conclusão e trabalhos futuros	36
	Referências	38

1 Introdução

Para se manterem competitivas, é fundamental que as organizações consigam alocar seus recursos disponíveis da forma mais eficiente possível. Além disso, precisam oferecer serviços e produtos que correspondam às expectativas dos clientes. O avanço das tecnologias de informação e da maior acessibilidade aos meios virtuais possibilitou a consolidação do comércio eletrônico, onde a experiência de entrega das mercadorias proporcionada aos clientes deve ser rápida e confiável.

Uma das formas de adotar processos logísticos mais ágeis e adaptados aos perfis dos clientes é a otimização da utilização de veículos e de suas rotas de entrega, objetivando a minimização dos custos envolvidos na operação. Na literatura, este problema de otimização corresponde ao Problema de Roteamento de Veículos (PRV).

De acordo com [Arenales et al. \(2007\)](#) o PRV envolve o projeto de rotas de entrega e/ou coleta de custo mínimo, partindo de um ou mais depósitos para um número de clientes, sujeito a restrições adicionais. Este tipo de problema tem um papel fundamental na área de gerenciamento da distribuição e logística.

Conforme o contexto de sua aplicação, o PRV pode assumir certas características e estar sujeito a diferentes restrições. Algumas das principais variações encontradas nos estudos sobre o problema são: existência de múltiplos depósitos; roteamento com necessidade de entrega e coleta; veículos com diferentes capacidades; clientes com janelas de atendimento. Ressalta-se, entretanto, que tais variações estão relacionadas apenas com parâmetros determinísticos e estáticos, ou seja, parâmetros que são conhecidos antes das rotas dos veículos serem iniciadas e que não variam de acordo com o tempo. Em situações reais, essas características e restrições podem não considerar aspectos essenciais do problema, como as incertezas intrínsecas aos processos logísticos. Nesse contexto, surge o Problema de Roteamento de Veículos Estocástico (PRVE), onde pelo menos um dos parâmetros do problema é considerado estocástico, ou seja, incerto.

Com o desenvolvimento da teoria da complexidade no início dos anos 1970, tornou-se claro que problemas como o PRV eram classificados como NP-difíceis ([GENDREAU, 2019](#)). Problemas deste tipo possuem uma grande complexidade e o tempo para encontrar soluções exatas em instâncias médias ou altas pode ser muito elevado. Portanto, em contextos onde a tomada de decisão têm de ser feita de forma ágil, encontrar a solução exata para esses problemas torna-se inviável. Essa percepção enfatizou o papel de heurísticas para resolver os problemas combinatórios que foram encontrados em aplicações da vida real e que precisavam ser resolvidos, fossem eles NP-difíceis ou não ([GENDREAU, 2019](#)).

Tendo em vista tais considerações, este trabalho aborda o Problema de Roteamento de Veículos com Janelas de Tempo (PRVJT) e tempos de viagem e serviço estocásticos. São

propostos algoritmos baseados nas chamadas *simheuristics*, técnicas que permitem combinar simulações e meta-heurísticas, como a *Iterated Local Search* (ILS) e *Variable Neighborhood Search* (VNS), para encontrar soluções aproximadas para o problema. Visando incorporar e avaliar a presença de incerteza ao modelo, são utilizadas simulações de Monte Carlo.

A necessidade de propor algoritmos para resolver essa variação do problema se dá pois nem sempre os modelos que consideram todos os parâmetros como determinísticos conseguem refletir as situações reais de forma detalhada. Nos casos onde os clientes devem ser atendidos dentro de um espaço de tempo, como é o caso de estudo deste trabalho, a violação das janelas de atendimento pode tornar uma rota inviável e gerar grandes prejuízos.

Em cenários reais, o tempo de viagem das rotas pode ser afetado por diversos aspectos, como condições meteorológicas, má qualidade das vias, acidentes no trajeto, congestionamentos, entre outros. Sendo assim, considerar o tempo de viagem como um parâmetro estocástico pode tornar o modelo muito mais próximo da realidade, de forma que as soluções encontradas sejam, de fato, viáveis de serem aplicadas.

O presente trabalho tem como objetivo geral o desenvolvimento de algoritmos baseados nas *simheuristics*, de forma que a meta-heurística ILS seja combinada com as simulações, visando encontrar boas soluções para o PRVJT com incerteza nos tempos de viagem e serviço em tempo computacional aceitável. A meta-heurística VNS será utilizada durante a fase de busca local do ILS. Os objetivos específicos do trabalho são: (i) revisar a literatura sobre os métodos utilizados para resolução do PRVJT com incerteza nos tempos de viagem e serviço; (ii) relaxar as restrições de capacidades dos veículos e de janelas de tempo do modelo matemático do PRVJT; (iii) propor um algoritmo baseado na meta-heurística ILS para resolução do problema de forma determinística; (iv) incorporar a meta-heurística VNS durante a fase de busca local do ILS; (v) validar o algoritmo proposto através da comparação com soluções exatas em instâncias de pequeno porte; (vi) incorporar a presença da incerteza dos tempos de viagem e serviço no método proposto e analisar seus impactos; (vii) comparar as soluções obtidas com soluções encontradas em instâncias teste da literatura.

2 Referencial Teórico

Este capítulo, conforme descrito em [Venanzi \(2016\)](#), tem como objetivo reunir ideias e teorias de diferentes fontes, de modo a construir uma nova forma de apresentação de um assunto já estudado e conhecido.

2.1 Logística e otimização combinatória

Conforme definição encontrada em [Ballou \(2018\)](#), logística é o processo de planejamento, implantação e controle do fluxo eficiente e eficaz de mercadorias, serviços e das informações relativas desde o ponto de origem até o ponto de consumo com o propósito de atender às exigências dos clientes. Entre as atividades-chave da logística estão o transporte e a manutenção de estoques, atividades primárias na absorção de custos que podem representar entre metade e dois terços dos custos logísticos totais.

As atividades de transporte, de acordo com [Grant \(2013\)](#), podem ser definidas como o planejamento e a execução da movimentação de mercadorias por uma transportadora entre dois pontos com um custo efetivo e cumprimento de prazos e condições especificadas pelo expedidor. [Ballou \(2018\)](#) afirma que um sistema de transportes eficiente e barato contribui para intensificar a competitividade no mercado, aumentar as economias de escala na produção e reduzir os preços dos produtos em geral.

Fica evidenciada, portanto, a importância das atividades de transporte dentro da logística. Tornar estas atividades o mais eficientes possível pode ser um fator de grande relevância em termos de competitividade. Nesse contexto, a otimização combinatória surge como uma das formas de potencializar a redução de custos nas atividades de transporte.

Nos últimos anos, o desenvolvimento da capacidade de processamento e memória dos computadores, aliado a modelos mais eficientes e a uma crescente tecnologia de algoritmos, tem tornado possível a solução de aplicações práticas reais de programação inteira, algumas envolvendo centenas de milhares de variáveis e restrições. Uma parte significativa destes problemas, dentre eles, os problemas de logística, podem ser resolvidos através de técnicas de otimização combinatória e programação inteira. ([GOLDBARG; GOLDBARG; LUNA, 2015](#)).

Foco deste trabalho, os problemas de roteirização de veículos, por exemplo, possuem importantes aplicações na área da logística, principalmente na redução de custos associados ao transporte de mercadorias. Como consequência, eles se tornaram alguns dos problemas mais estudados na literatura de otimização combinatória e um grande número de artigos e livros lidam com os numerosos procedimentos que foram propostos para resolvê-los ([GENDREAU; LAPORTE; SEGUIN, 1996](#)).

2.2 Problema de Roteamento de Veículos

O Problema de Roteamento de Veículos (PRV) foi abordado pela primeira vez em [Dantzig e Ramser \(1959\)](#) e pode ser considerado como uma generalização do Problema do Caixeiro Viajante com a introdução de condições adicionais.

[Cordeau et al. \(2002\)](#) define a versão clássica do PRV da seguinte maneira. Seja $G = (V, A)$ um grafo, onde $V = \{v_0, v_1, \dots, v_n\}$ é o conjunto de vértices e $A = \{(v_i, v_j) : v_i, v_j \in V, i \neq j\}$ é o conjunto de arcos. O vértice v_0 representa o Centro de Distribuição (CD), enquanto os demais vértices representam os clientes a serem atendidos. Com A são associadas uma matriz de custos c_{ij} e uma matriz de tempos de viagens t_{ij} . Se estas matrizes são simétricas, o que é o caso mais comum, então o padrão é definir o PRV em um grafo não direcionado $G = (V, E)$ onde $E = \{(v_i, v_j) : v_i, v_j \in V, i < j\}$ é um conjunto de arestas. Cada cliente tem uma demanda não negativa q_n e um tempo de serviço t_s . Uma frota de K veículos idênticos de capacidade Q é atribuída ao CD. O número de veículos é conhecido com antecedência ou tratado como uma variável de decisão. Nesse contexto, o PRV consiste em designar um conjunto de no máximo m rotas de entrega e/ou coleta de forma que: (i) cada rota inicia e termina no CD; (ii) cada cliente é visitado exatamente uma vez por exatamente um veículo; (iii) a demanda total de cada rota não exceda Q ; (iv) a duração total de cada rota (incluindo os tempos de viagem e atendimento) não exceda um limite pré-definido D ; (v) o custo total da rota é minimizado.

As equações que representam a formulação matemática do problema, adaptadas de [Aquino \(2015\)](#), podem ser observadas logo em seguida. O modelo utiliza uma variável binária x_{ijk} que assume valor 1 quando o veículo k utiliza o arco (i, j) para integrar a solução final e 0 quando contrário.

$$\text{Minimizar } \sum_{k \in K} \sum_{i \in V} \sum_{j \in V} c_{ij} x_{ijk} \quad (2.1)$$

$$\text{Sujeito a } \sum_{k \in K} \sum_{i \in V} x_{ijk} = 1 \quad j \in V \quad (2.2)$$

$$\sum_{j \in V} x_{0jk} = 1 \quad k \in K \quad (2.3)$$

$$\sum_{i \in V} x_{ijk} - \sum_{i \in V} x_{jik} = 0 \quad j \in V, k \in K \quad (2.4)$$

$$\sum_{i \in V} x_{i0k} = 1 \quad k \in K \quad (2.5)$$

$$\sum_{i \in V} q_i \sum_{j \in V} x_{ijk} \leq Q \quad k \in K \quad (2.6)$$

$$x_{ijk} \in \{0, 1\} \quad i, j \in V, k \in K \quad (2.7)$$

A expressão (2.1) representa o custo total a ser minimizado. A equação (2.2) garante que todos os clientes sejam atendidos e que cada um seja atendido por apenas um veículo. As

expressões (2.3), (2.4) e (2.5) garantem que o fluxo percorrido pelo veículo k seja mantido. A expressão (2.6) garante que as restrições de capacidade dos veículos sejam atendidas. Por fim, a expressão (2.7) define o domínio das variáveis.

2.3 Problema de Roteamento de Veículos com Janelas de Tempo

De acordo com [Cordeau et al. \(2002\)](#), uma das variações mais comuns do PRV clássico é o Problema de Roteamento de Veículos com Janelas de Tempo (PRVJT), onde uma janela de tempo $[a_i, b_i]$ é imposta na visita de cada cliente. Sendo assim, um cliente não pode ser atendido antes de a_i e nem após o tempo de b_i .

[Solomon \(1987\)](#) foi o primeiro autor a abordar o PRVJT, ressaltando que na presença das janelas de tempo o roteamento total e os custos de programação incluem não apenas as distâncias totais de viagem e seus custos embutidos, mas também os custos do tempo de espera incorridos quando um veículo chega muito cedo ao cliente ou quando o veículo é carregado ou descarregado. Além disso, o autor considera que o PRVJT surgiu como uma área importante para o progresso no tratamento de complicações realistas e generalizações do modelo básico de roteamento, uma vez que as janelas de tempo surgem naturalmente em problemas enfrentados por organizações empresariais que funcionam em horários fixos. Alguns dos exemplos dados pelo autor são as entregas bancárias, entregas postais, coleta de lixo industrial e roteamento de ônibus escolar.

[Miranda \(2011\)](#) define o PRVJT como a generalização do problema de roteamento de veículos na qual se associa, a cada cliente, um período de tempo no qual algum veículo deve começar a atendê-lo. A esse intervalo dá-se o nome de janela de tempo. O objetivo primário do problema é minimizar o número de veículos e o objetivo secundário é minimizar a distância total percorrida, neste caso, o tempo total de viagem. O autor ainda destaca a existência de problemas em que as janelas de tempo podem ser violadas mediante o pagamento de penalidades, classe conhecida como *Soft Time Windows*. Em casos onde essa violação não é permitida, a classe é denominada *Hard Time Windows*.

A formulação matemática do PRVJT tem como base as expressões do PRV clássico, mostradas no tópico anterior. Entretanto, com a inserção das janelas de tempo, duas novas restrições são incorporadas ao modelo. As restrições adicionais, adaptadas de [Arenales et al. \(2007\)](#), são apresentadas nas equações (2.8) e (2.9):

$$s_{ik} + t_{ij} + ts_i \leq s_{jk} + (1 - x_{ijk})M_{ij}, \quad i, j \in A, k \in K \quad (2.8)$$

$$a_i \leq s_{ik} \leq b_i \quad i \in V, k \in K \quad (2.9)$$

A variável s_{ik} indica o instante em que o veículo k começa a atender o cliente i . Já t_{ij} representa o tempo de viagem entre o cliente i e o cliente j . O tempo de serviço do cliente i é

representado por t_{si} . Por fim, $M_{ij} = \max\{b_i + s_{ik} + t_{ij} + t_{si}, 0\}$. Dessa forma, as expressões (2.8) e (2.9) asseguram que os veículos visitem os clientes dentro das janelas de tempo disponíveis.

2.4 Problema de Roteamento de Veículos Estocástico

Segundo Gendreau, Laporte e Seguin (1996), o Problema de Roteamento de Veículos Estocástico (PRVE) surge sempre que alguns elementos do problema são aleatórios. O PRVE difere do PRV com parâmetros determinísticos em vários aspectos fundamentais, uma vez que o conceito de solução é diferente e várias propriedades fundamentais de PRVs determinísticos não se sustentam no caso estocástico.

De acordo com Maia (2021), geralmente quatro parâmetros do problema costumam ser considerados incertos:

- Disponibilidade do cliente: a presença do cliente no momento do atendimento é incerta;
- Tempos de viagem: o tempo exato de viagem entre clientes é desconhecido;
- Tempos de serviço: o tempo exato de atendimento de cada cliente não é conhecido;
- Demandas: não se sabe ao certo o volume da demanda de cada cliente.

Em seu trabalho de revisão de literatura sobre o Problema de Roteamento de Veículos Estocástico (PRVE), Oyola, Arntzen e Woodruff (2018) afirmam que a estocasticidade pode ser incorporada ao PRV por meio de diferentes aspectos e, normalmente, um ou dois elementos são considerados estocásticos. Esta limitação se dá provavelmente devido à dificuldade de resolver um problema onde muitos parâmetros diferentes são estocásticos. No presente trabalho, o tempo de viagem entre os clientes e o tempo de atendimento são considerados como parâmetros estocásticos.

Diversos trabalhos sobre o PRVE com tempos de viagem estocásticos já foram realizados. Na maioria deles, os autores costumam utilizar duas formas de modelar os problemas: como um *Chance Constrained Program* (CCP) ou como um *Stochastic Program with Recourse* (SPR). No caso do CCP, o problema é resolvido de forma que a probabilidade de falha da rota esteja abaixo de um certo nível e os custos das falhas geralmente são ignorados (GENDREAU; LAPORTE; SEGUIN, 1996). Uma rota é considerada falha quando alguma das restrições é violada. Por outro lado, no SPR permitem-se falhas de rota, mas o tomador de decisões deve definir uma política de recurso, descrevendo quais ações devem ser tomadas a fim de reparar a solução após um fracasso, de forma que o custo de transporte esperado é otimizado. Problemas modelados como SPR são mais difíceis de resolver, mas os objetivos são mais significativos (GENDREAU; LAPORTE; SEGUIN, 1996).

Uma versão do PRVE com *Soft Time Windows* e tempos de viagem estocásticos é encontrada em Ando e Taniguchi (2006) apud Oyola, Arntzen e Woodruff (2018). Neste modelo,

um veículo pode fazer várias rotas por dia e todas as mercadorias de cada cliente devem ser carregadas no veículo ao mesmo tempo. O peso total das mercadorias em uma rota não deve exceder a capacidade do veículo. Além disso, há uma *Hard Time Window* para o depósito. Uma distribuição triangular é estimada para o tempo de viagem usando dados reais. O objetivo do problema é minimizar o custo total, que é dado pelo custo fixo de uso dos veículos, os custos operacionais e as penalidades por chegar fora das janelas de tempo. A penalidade para chegadas tardias ou antecipadas pode ser vista como um recurso. O tempo de serviço é assumido para ser determinístico. Os testes são realizados para o caso de um único veículo.

Outra versão do PRVE com tempos de viagem estocásticos que inclui coletas e entregas é apresentada em Zhang, Chaovalitwongse e Zhang (2012) apud Oyola, Arntzen e Woodruff (2018), onde cada cliente pode ter demandas tanto de entrega como de coleta. O veículo tem um tempo máximo de viagem B . O problema é modelado como CCP, então em uma solução viável a probabilidade do tempo de viagem de um veículo ser menor ou igual a B deve ser maior ou igual a um determinado parâmetro. O teste é realizado em instâncias com tempo de viagem seguindo uma distribuição normal.

Um PRVE com janelas de tempo e tempos de viagem e serviço estocásticos é estudado em Li, Tian e Leung (2010) apud Oyola, Arntzen e Woodruff (2018). Neste trabalho, o problema é modelado usando um modelo CCP e um modelo SPR. Na abordagem CCP dois aspectos são considerados, a probabilidade de chegar a cada cliente dentro das janelas de tempo e a probabilidade de terminar uma rota dentro de um dado tempo. Na segunda abordagem, o valor esperado de certos custos extras é computado: a penalidade por chegar após o prazo das janelas de tempo e o custo das horas extras do motorista. Os tempos de viagem e de serviço são assumidos com uma distribuição de probabilidade normal.

Uma terceira forma de abordar o PRVE é através da combinação de algoritmos de otimização e simulação, as chamadas *simheuristics*. Guimarans et al. (2018) apud Maia (2021) usam um algoritmo baseado no ILS para resolver o PRVJT com tempos de viagem e serviço estocásticos, sendo que os custos das soluções são calculados a partir de simulações de Monte Carlo. Durante o processo de busca são realizadas simulações rápidas e todos os mínimos locais são armazenados como melhores soluções candidatas. Ao final da otimização é executada uma etapa de refinamento, onde as soluções armazenadas passam por uma simulação mais extensa com o objetivo de encontrar a melhor solução no contexto estocástico.

2.5 Meta-heurísticas

Os problemas de otimização consistem basicamente em encontrar a melhor combinação dentre um conjunto de variáveis, de forma a maximizar ou minimizar uma determinada função objetivo. Tendo em vista a complexidade em encontrar soluções exatas para os problemas da classe NP-Difícil, nos últimos anos intensificaram-se os estudos que tem como objetivo o

desenvolvimento e aprimoramento de métodos conhecidos como meta-heurísticas.

De acordo com [Pereira \(2003\)](#), as meta-heurísticas têm subjacente um processo iterativo que guia e modifica as operações de heurísticas subordinadas para, de uma forma eficiente, produzir soluções de boa qualidade. Tal procedimento manipula uma solução ou um conjunto de soluções em cada iteração. As heurísticas subordinadas podem ser de baixo ou alto nível, simplesmente uma pesquisa local, ou somente um método construtivo.

[Bezerra \(2020\)](#) enfatiza a existência de dois grupos de meta-heurísticas: de população e de solução única. Conforme o nome indica, as meta-heurísticas de população, baseiam-se no comportamento de populações da natureza, como é o exemplo das abelhas, formigas, etc. Já as de solução única recorrem a ferramentas de memória ou probabilísticas. A grande diferença entre os dois grupos prende-se com o número de soluções atuais que utilizam e o número de soluções vizinhas. Enquanto as meta-heurísticas de solução única apresentam sempre uma única solução atual, as de população apresentam sempre um número superior, o que lhe permite originar um maior número de soluções vizinhas, tornando assim a sua pesquisa mais intensa e abrangente. Essa característica permite às meta-heurísticas de população, na maior parte dos casos, encontrarem a solução do problema mais rapidamente.

Neste trabalho são utilizadas as meta-heurísticas de solução única ILS e VNS para resolução do problema.

2.6 Iterated Local Search

Introduzido por [Lourenço, Martin e Stützle \(2003\)](#), o *Iterated Local Search* (ILS) é um método baseado em buscas locais que focam a busca não no espaço completo de soluções, mas em um pequeno subespaço definido por soluções que são ótimas locais de determinado procedimento de otimização. A principal ideia do método é a de que um procedimento de busca local pode ser melhorado a partir de perturbações na solução ótima local. A perturbação precisa ser suficientemente forte para permitir que a busca local explore diferentes soluções, mas também fraca o suficiente para evitar um reinício aleatório. Conforme descrito em [Lourenço, Martin e Stützle \(2003\)](#), para sua aplicação devem ser especificadas quatro componentes básicos:

1. Um procedimento para gerar uma solução inicial para o problema: pode-se começar com uma solução aleatória ou retornada por alguma heurística de construção gulosa;
2. Um procedimento de busca local: para a maioria dos problemas, um algoritmo de busca local está apto para ser utilizado;
3. Procedimento de perturbação da solução corrente: movimentos aleatórios em uma vizinhança de ordem superior se mostram muito eficazes;

4. Um critério de aceitação: um dos critérios de aceitação mais utilizados é forçar o custo da solução a diminuir. De forma geral, algoritmos ILS com este critério de aceitação apresentam um desempenho muito melhor do que as abordagens de reinicialização aleatória.

Abaixo, segue o pseudocódigo que representa o método ILS:

Algoritmo 1: Iterated Local Search

```
 $s_0 \leftarrow \text{GeraSolucaoInicial}();$   
 $s^* \leftarrow \text{BuscaLocal}(s_0);$   
enquanto (o critério de parada não estiver satisfeito) faça  
     $s' \leftarrow \text{Pertubacao}(s^*);$   
     $s^{*'} \leftarrow \text{BuscaLocal}(s');$   
     $s^* \leftarrow \text{CriterioDeAceitacao}(s^*, s^{*'}, \text{historico});$   
fim  
retorne  $s^*$ 
```

Fonte: Lourenço, Martin e Stützle (2003)

A primeira linha do algoritmo tem como objetivo gerar uma solução inicial para o problema que, conforme descrito anteriormente, pode ser uma solução aleatória ou retornada por alguma heurística de construção gulosa.

A segunda linha realiza uma busca local na atual solução. No caso, é realizada uma busca na solução gerada na primeira linha. Para realizar a busca local é preciso gerar uma vizinhança para a solução vigente e buscar a melhor solução dentre os vizinhos. A vizinhança é definida como um conjunto de soluções derivadas da solução atual por meio de pequenos movimentos.

A terceira linha controla a repetição das linhas 4, 5 e 6. Enquanto o critério de parada do algoritmo não estiver satisfeito, a meta-heurística ILS será executada. Exemplos para o critério de parada são o número de iterações e o tempo de execução do algoritmo.

A quarta linha é a fase de perturbação, onde a solução atual recebe alguma outra solução gerada a partir de modificações na melhor solução encontrada até o momento.

Na quinta linha é realizada uma busca local na solução vigente, atualizando a melhor solução da atual iteração. É comum que esta busca local seja realizada por meio de outras meta-heurísticas, como, por exemplo, o VNS.

A sexta linha é responsável por comparar a melhor solução da atual iteração com a melhor solução encontrada até o momento e atualiza-la ou não. Para isso é estabelecido um critério de aceitação. É comum que o critério de aceitação obrigue que a melhor solução da atual iteração seja melhor que a solução vigente.

2.7 Variable Neighborhood Search

Apresentada por Mladenović e Hansen (1997) a meta-heurística *Variable Neighborhood Search* (VNS) tem como objetivo a resolução de problemas de otimização combinatória. É um método de busca local que consiste em explorar o espaço de soluções por meio de trocas sistemáticas de estruturas de vizinhança. Ao contrário da maioria dos outros métodos de busca local, o VNS não segue uma trajetória, mas explora vizinhanças cada vez mais distantes da solução atual e as explora se, e somente se, um movimento de melhora é realizado.

O VNS é inicializado através da geração de uma solução inicial S . Logo após, é feita a seleção de um conjunto de k_{max} estruturas de vizinhança N_k , $k = (1, 2, \dots, k_{max})$, que representam a quantidade de vizinhanças que serão exploradas. Partindo da solução inicial, é realizada a primeira vizinhança $N_1(S)$. Caso a solução encontrada seja melhor que a solução corrente, a solução passa a ser a nova solução. Caso em uma vizinhança não seja encontrada uma solução melhor que a atual, passa-se para a próxima vizinhança $N_{k+1}(S)$. O critério de parada é acionado quando a última vizinhança $N_{k_{max}}(S)$ for analisada. Abaixo, encontra-se o pseudocódigo do método para um problema de minimização (OLIVEIRA, 2020):

Algoritmo 2: Variable Neighborhood Search

```

Entradas: ( $S$ ,  $k_{max}$ );
 $S = SolucaoInicial$ ;
 $k = 1$ ;
enquanto  $k \leq k_{max}$  faça
     $S' = BuscaLocal(S)$ ;
    se  $f(S') \leq f(S)$  então
         $S = S'$ ;
         $k = 1$ ;
        senão
             $k = k + 1$ 
        fim
    fim
fim
Retorna  $S$ ;
Fim VNS

```

2.8 Método de Monte Carlo

O Método de Monte Carlo, ou Simulação de Monte Carlo, é uma ferramenta que permite estimar resultados de eventos incertos por meio de repetidas iterações que calculam valores aproximados de probabilidade. Quando se utiliza um modelo matemático para descrever um sistema, é possível que o modelo seja complexo demais, ou então, não permita uma solução analítica. Nesse caso, a simulação computacional pode ser considerada uma ferramenta de grande valia na obtenção de uma resposta para um problema particular. Quando o modelo envolve

amostragem aleatória de uma distribuição probabilística, o método é designado Simulação de Monte Carlo (DONATELLI; KONRATH, 2005). Segundo Rajabalinejad (2010), cada simulação é independente das simulações realizadas anteriormente.

Entre as vantagens da utilização deste tipo de simulação estão a ausência de erros sistemáticos, proporcionada pela independência entre as simulações, e a relativa facilidade em compreender seus resultados. Uma desvantagem, no entanto, é o grande número de execuções necessários, particularmente em modelos complexos, onde cada execução pode implicar um análise de elementos finitos ou outro procedimento demorado (RAJABALINEJAD, 2010).

A importância da utilização da simulação de Monte Carlo para o problema deste trabalho se dá pois, conforme Juan et al. (2015) comprovam em seu trabalho, em contextos onde a incerteza é baixa ou moderada, as soluções viáveis para a versão determinística do problema são consideradas soluções promissoras para sua versão estocástica. Sendo assim, a simulação é utilizada para obter estimativas dos valores associados à cada solução. Em outras palavras, a simulação é utilizada para determinar qual das soluções, dentre as melhores encontradas na versão determinística do problema, apresentam os menores tempos de rota e respeitam as restrições de janelas de atendimento quando considerados os tempos de viagem e serviço como estocásticos. No entanto, os autores afirmam que não necessariamente a melhor solução encontrada para a versão determinística irá se tornar a melhor solução encontrada para a versão estocástica.

2.9 Simheuristics

As *simheuristics* são técnicas que permitem lidar com as incerteza da vida real de maneira natural, integrando a simulação (em qualquer uma de suas variantes) em uma estrutura orientada por meta-heurísticas. Esses algoritmos orientados à otimização baseiam-se no fato de que já existem meta-heurísticas eficientes para a versão determinística do problema de otimização combinatória correspondente. As *simheuristics* também facilitam a introdução de critérios de análise de risco e/ou confiabilidade durante a avaliação de soluções alternativas de alta qualidade para problemas estocásticos (JUAN et al., 2015).

As simulações podem ser aplicadas tanto na função objetivo quanto na avaliação de restrições em problemas de otimização com componentes estocásticos. As simulações dentro de uma otimização podem ocorrer de várias formas. A maneira mais simples é realizar um número constante de simulações a cada iteração, mas também é possível intensificar as simulações com a convergência do algoritmo, com o objetivo de ter soluções cada vez mais refinadas a medida que a busca vai se aproximando da solução final. Outra alternativa é fazer simulações rápidas durante a otimização e incluir uma simulação mais intensa ao final do processo para refinar a solução final (MAIA, 2021).

Em seu trabalho sobre as aplicações desta técnica, Juan et al. (2019) destacam alguns exemplos da utilização das *simheuristics* para problemas de logística. Gonzalez-Martin et al.

(2018) propõem um algoritmo baseado em *simheuristics* para resolver o PRV com demandas estocásticas. Aqui, os autores usam a Simulação de Monte Carlo para estender a heurística RandSHARP, que foi originalmente projetado para resolver a versão determinística do problema.

Gruler et al. (2017) discutem a necessidade de otimizar a coleta de lixo urbano em cidades inteligentes e formulam o problema como uma extensão do PRV. Primeiramente, os autores desenvolvem uma meta-heurística baseada em VNS para resolver a variante determinística do problema. Em seguida, eles estendem sua abordagem em uma *simheuristic* para lidar com níveis inesperados de desperdício dentro dos containers. Seu algoritmo é testado usando um conjunto de *benchmark* em grande escala para o problema de coleta de resíduos com várias restrições da vida real. Seus resultados incluem uma análise de risco considerando a variação dos resíduos nível e capacidades de segurança do veículo.

3 Metodologia

Conforme os conceitos propostos por [Venanzi \(2016\)](#), o presente trabalho caracteriza-se como uma pesquisa de natureza aplicada, uma vez que tem como objetivo gerar uma aplicação prática, de modo a tentar solucionar problemas específicos do cotidiano. Além disso, trata-se de um trabalho de abordagem quantitativa, onde os resultados obtidos serão quantificados segundo uma escala de intervalo ou de razão e, posteriormente, serão analisados utilizando-se recursos e técnicas estatísticas.

Do ponto de vista dos objetivos, a pesquisa realizada é considerada descritiva, visto que serão realizadas análises que buscam descrever o comportamento da aplicação do método desenvolvido em diferentes instâncias da literatura. Além disso, os resultados do método desenvolvido são comparados com outros estudos sobre o assunto.

Os procedimentos técnicos foram baseados principalmente em pesquisas bibliográficas e pesquisas experimentais. As pesquisas bibliográficas justificam-se pela utilização de métodos já existentes na literatura para resolução de um problema já conhecido, atualizando seus possíveis campos de aplicação. Por outro lado, as pesquisas experimentais justificam-se pois o trabalho visa alterar certos aspectos e variáveis de métodos da literatura, de forma a observar o comportamento dos resultados obtidos.

3.1 Implementação do algoritmo ILS-VNS

Nesta seção serão detalhados os parâmetros de ajuste relativos à implementação das meta-heurísticas ILS e VNS utilizadas para resolução do problema, assim como seu funcionamento. A implementação do algoritmo foi feita na linguagem de programação Python, em sua versão 3.10.2.

3.1.1 Leitura dos dados

A base de dados utilizada para validação dos resultados do algoritmo foi a de [Solomon \(1987\)](#). Desta forma, o algoritmo recebe um arquivo em formato *.txt* que contém o número de veículos disponíveis, a capacidade dos veículos, o número de clientes, suas coordenadas e demandas, o tempo de início e de fim das janelas de atendimento e os tempos de serviço.

Pelo fato de não ser disponibilizada uma matriz com as distâncias entre os clientes, é necessário que esta seja calculada a partir das coordenadas informadas. Sendo assim, o algoritmo gera uma matriz quadrada com tamanho igual ao número de clientes acrescida de 1 (que representa o depósito). Por exemplo, caso o número de locais a serem visitados seja de 100, será gerada uma matriz quadrada de tamanho 101. Posteriormente, a matriz é preenchida a partir do cálculo

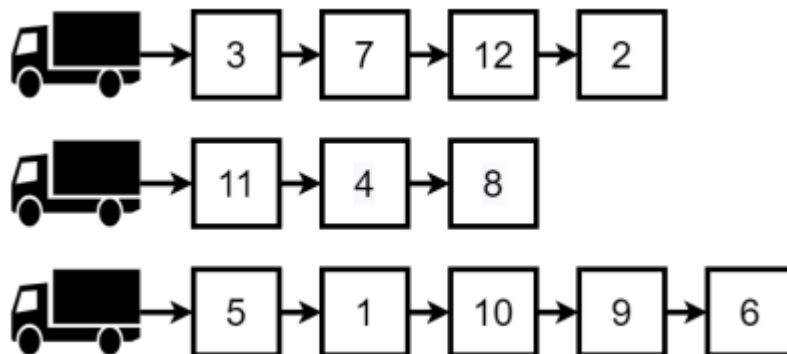
da distância euclidiana entre os clientes. Por se tratar de um trabalho onde os tempos de viagem e serviços são os parâmetros incertos, foi considerado que cada unidade de distância percorrida pelos veículos equivale à uma unidade de tempo.

Após o término desta etapa, os dados são processados e disponibilizados em forma de vetores ou de matrizes para utilização nas etapas posteriores do algoritmo.

3.1.2 Representação de uma solução

Em certas etapas da implementação, o algoritmo irá retornar saídas que contém as soluções geradas. Estas soluções representam uma lista de rotas que contém a sequência de clientes visitados pelos veículos. Todos os clientes devem ser visitados apenas uma vez, por apenas um veículo. Por sua vez, um veículo pode atender a diferentes clientes, desde que as restrições impostas ao problema sejam atendidas. A Figura 3.1 ilustra uma possível solução onde três veículos atendem doze clientes:

Figura 3.1 – Exemplo de solução candidata com três veículos e doze clientes.



Fonte: Maia (2021)

O exemplo acima indica que o primeiro veículo parte do depósito, visita os clientes indicados na sequência dentro dos quadrados, e retorna ao depósito. O mesmo vale para o segundo e para o terceiro veículo.

3.1.3 Cálculo da função de avaliação

A função que avalia o custo de uma solução considera os tempos utilizados pelos veículos para atender todos os clientes. Além disto, no trabalho em questão as restrições referentes às janelas de tempo e capacidades dos veículos foram relaxadas, sendo adicionadas em forma de penalidades à função de avaliação. Sendo assim, quando um veículo viola sua capacidade de carga, uma penalidade P_1 é aplicada. Enquanto, a penalidade para veículos que transgredirem as janelas de atendimentos dos clientes são representadas por P_2 . Abaixo, segue a de avaliação (FA) utilizada no algoritmo:

$$FA(S) = \sum_{k \in K} \sum_{i \in V} \sum_{j \in V} c_{ij} x_{ijk} + \sum_{R \in S} P_1 (\max(0, \sum_{i \in R} d_i - Q)) + P_2 \sum_{i \in R} \max(0, (ts_{ik} - b_i))$$

Onde:

- S é uma solução que contém um conjunto de rotas R ;
- c_{ij} é o custo, ou tempo, de viagem entre os clientes i e j ;
- x_{ijk} é uma variável binária que assume valor 1 quando o veículo k utiliza o arco (i, j) para integrar a solução final e 0 quando contrário;
- ts_{ik} é o momento em que o cliente i começa a ser atendido pelo veículo k ;
- t_0 é o tempo em que o veículo retorna ao depósito.

Os valores das penalidades P_1 e P_2 foram definidos arbitrariamente com o valor de 100.

3.1.4 Geração da solução inicial

Após a leitura dos dados, o algoritmo dá início ao funcionamento do ILS-VNS em si. A geração da solução inicial é feita de forma randômica, sem utilizar critérios como, por exemplo, os das heurísticas gulosas ou semi-gulosas. Os vetores que contém os veículos e suas respectivas capacidades são percorridos e os clientes são adicionados de forma aleatória à uma rota. Sendo assim, a capacidade restante dos veículos é subtraída da demanda de cada cliente visitado.

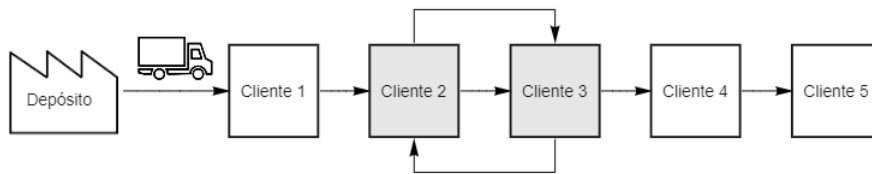
3.1.5 Busca local

O método escolhido para a fase de busca local do ILS foi o VNS. Nesse contexto, é importante que sejam detalhados certos aspectos da meta-heurística, como a quantidade de vizinhanças que serão exploradas e os movimentos de trocas nas estruturas de vizinhança.

Foram escolhidos quatro tipos de movimentos nas estruturas de vizinhança de uma solução, divididos em estruturas intra-rotas, onde são realizados movimentos que envolvam os clientes de uma mesma rota, e estruturas inter-rotas, que consistem na execução de movimentos que envolvam clientes de rotas diferentes. A seguir, são apresentados os movimentos escolhidos:

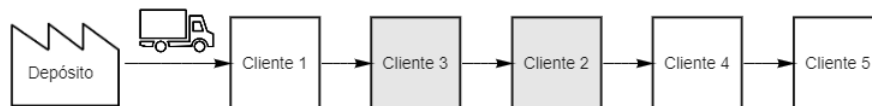
1. Troca da ordem de visitação de dois clientes - Mesmo veículo: movimento intra-rota que consiste em trocar a ordem de visitação de clientes quaisquer pertencentes à rota de um mesmo veículo. As figuras 3.2 e 3.3 ilustram esta estrutura:

Figura 3.2 – Movimento 1: configuração inicial



Fonte: Autor

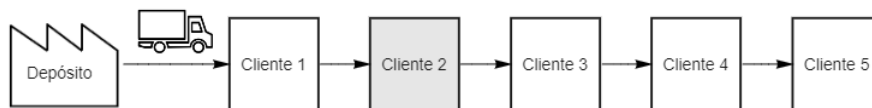
Figura 3.3 – Movimento 1: configuração após movimento



Fonte: Autor

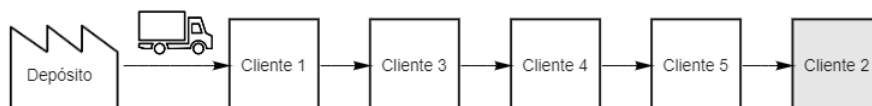
2. Troca da ordem de visitaç o de um cliente - Mesmo ve culo: movimento intra-rota que consiste em remover um cliente da rota e inseri-lo em outra posi o. As figuras 3.4 e 3.5 ilustram esta estrutura:

Figura 3.4 – Movimento 2: configura o inicial



Fonte: Autor

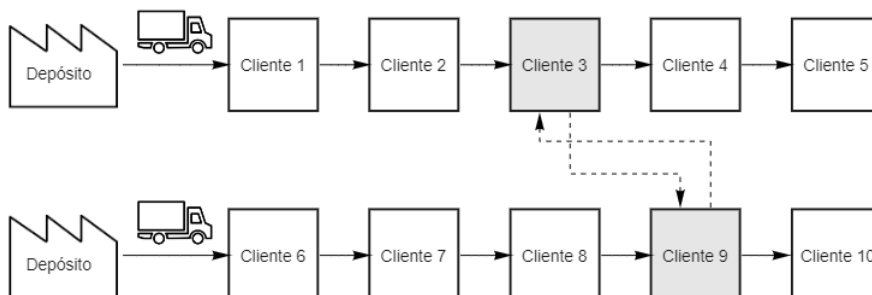
Figura 3.5 – Movimento 2: configura o ap s movimento



Fonte: Autor

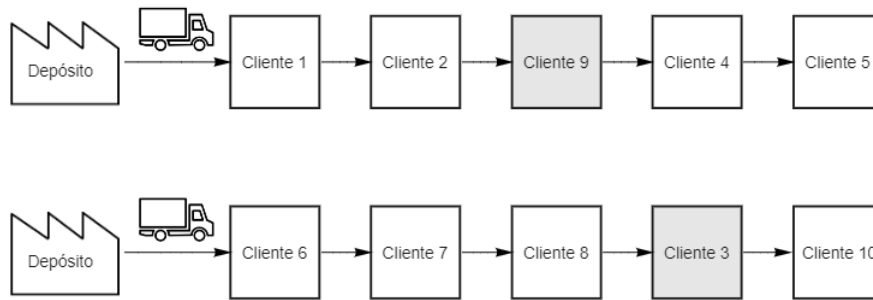
3. Troca da ordem de visita o de dois clientes - Ve culos diferentes: movimento inter-rota que consiste em trocar dois clientes de rotas. As figuras 3.6 e 3.7 ilustram esta estrutura:

Figura 3.6 – Movimento 3: configura o inicial



Fonte: Autor

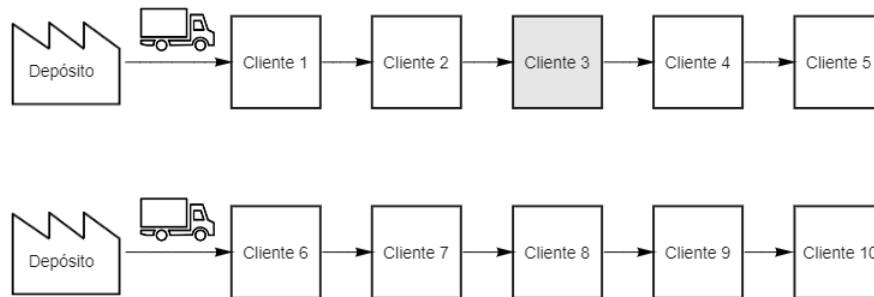
Figura 3.7 – Movimento 3: configuração após movimento



Fonte: Autor

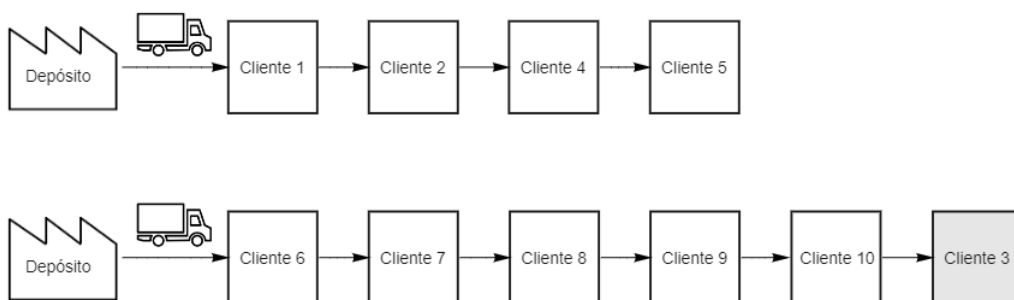
4. Troca da ordem de visitação de um cliente - Veículos diferentes: movimento inter-roteira que consiste em remover um cliente de uma rota e adicioná-lo em outra. As figuras 3.8 e 3.9 ilustram esta estrutura:

Figura 3.8 – Movimento 4: configuração inicial



Fonte: Autor

Figura 3.9 – Movimento 4: configuração após movimento



Fonte: Autor

A meta-heurística inicia com a definição dos parâmetros, sendo $k = 1$, que representa a estrutura de vizinhança a ser executada, e $k_{max} = 4$, que representa o critério de parada, concretizado no caso do último movimento, $k = 4$, não apresentar nenhuma melhora da solução.

A ordem de execução dos movimentos segue a mesma sequência mostrada acima. Sendo assim, o primeiro movimento a ser realizado pelo método é o de troca da ordem de visitação de dois clientes. Após a execução do movimento, o algoritmo compara o valor da FA da nova solução

com o valor da FA da configuração inicial. Caso o valor da nova configuração seja melhor que o anterior, k permanece com o valor 1. Caso contrário, k é acrescido de 1 e o próximo movimento é executado, assim como mostrado no Algoritmo 2.

3.1.6 Funcionamento do ILS

Após o refinamento da solução inicial ocorrida durante a fase de busca local inicial, o algoritmo do ILS entra em funcionamento. Desta forma, torna-se necessário a definição dos critérios de parada e das perturbações realizadas nas novas soluções geradas durante a busca local da meta-heurística. Essas definições são apresentadas nas subseções a seguir.

3.1.7 Critério de parada

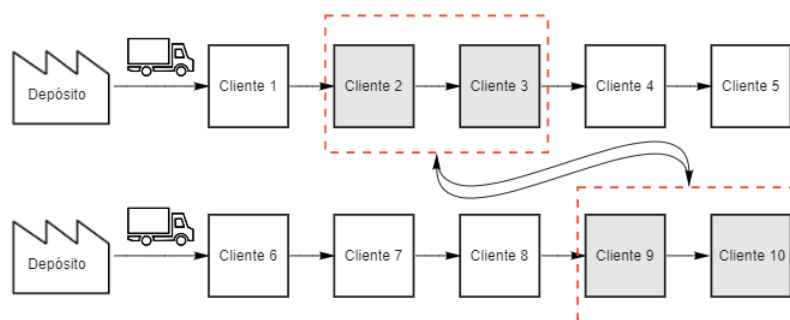
O critério de parada da execução do ILS foi definido como o número de iterações da meta-heurística realizadas. Neste contexto, foi criado um parâmetro denominado *iterMax*, com valor igual a 1000, para instâncias com 25 clientes, e com valor igual a 200, para instâncias com 50 clientes. Além disso, foi criada uma variável denominada *iterCorrente*, com valor inicial igual a 0. A cada iteração do ILS soma-se 1 ao valor de *iterCorrente*. A execução acontece enquanto *iterCorrente* for menor ou igual a *iterMax*.

3.1.8 Perturbações e critério de aceitação

Esta fase do ILS é responsável por modificar a solução corrente, gerando uma nova solução de partida para a busca local. As perturbações devem ser fortes o suficiente para permitirem que as buscas locais, realizadas por meio do VNS, escapem de mínimos locais, mas não tão fortes de forma a resultar em reinícios aleatórios da solução.

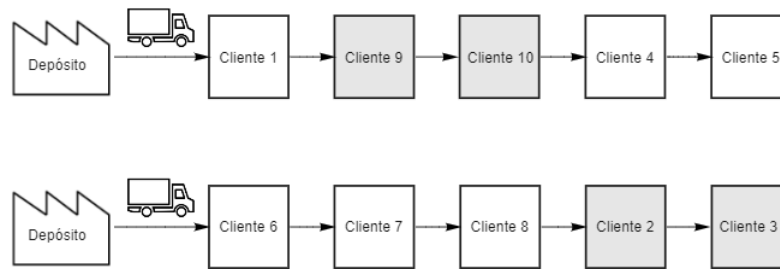
O movimento escolhido para modificar as soluções baseia-se na troca de 2 ou mais clientes de uma rota com os clientes de outra, em posições aleatórias, assim como ilustrado nas figuras 3.10 e 3.11:

Figura 3.10 – Perturbação: configuração inicial



Fonte: Autor

Figura 3.11 – Perturbação: configuração após movimento



Fonte: Autor

Após a perturbação, o algoritmo executa a busca local na solução perturbada e a solução retornada é avaliada e comparada com a melhor solução corrente. A solução gerada torna-se a melhor solução corrente se, e somente se, sua FA for menor ou igual à da melhor solução corrente. Neste caso, o contador de iterações sem melhora da solução é zerado. Caso contrário, ele é acrescido de 1.

O contador de iterações sem melhora funciona como regulador do nível das perturbações realizadas nas soluções. Ao serem atingidas 5 iterações sem alterações na melhor solução corrente, o número de trocas da perturbação passa de sua configuração inicial, que é de 2 trocas, para 3 trocas. Este processo se repete até que o número de trocas atinja o valor 6, quando reinicia-se o número de trocas, retornando seu valor para 2. O limite de 6 trocas foi definido arbitrariamente e baseia-se na concepção de que perturbações muito potentes podem gerar reinícios aleatórios da solução.

3.2 Simulações de Monte Carlo

A Simulação de Monte Carlo é utilizada com o objetivo de incorporar a presença de incerteza aos tempos de viagem e de serviço. Dessa maneira, os tempos determinísticos são substituídos por valores obtidos via simulação. Assim como nos trabalhos de [Guimarans et al. \(2018\)](#), [Gutierrez et al. \(2018\)](#) e [Maia \(2021\)](#), a distribuição de probabilidade adotada foi a Lognormal para os tempos de viagem e tempos de serviço. Essa distribuição é adequada para representar os parâmetros do problema por não apresentar valores negativos e possuir curvas com características similares à realidade. A distribuição Lognormal é útil para modelar dados com distribuições aproximadamente simétricas ou assimétricas à direita, o que se assemelha com as características do trânsito ([MAIA, 2021](#)).

De forma similar a [Gutierrez et al. \(2018\)](#) e [Maia \(2021\)](#), os parâmetros das distribuições foram ajustados considerando a média igual ao seu valor determinístico. Já a variância é ajustada de forma a se obter um Coeficiente de Variação (CV) igual a 0,2.

Presume-se que após um grande número de simulações os valores esperados dos tempos estocásticos sejam próximos aos seus respectivos tempos determinísticos. Levando isso em

consideração, o Método de Monte Carlo é utilizado para calcular não só as variações dos tempos de viagem e de serviço, mas, principalmente, as violações às janelas de tempo causadas por tais incertezas.

Logo abaixo, no Algoritmo 3, é apresentado o pseudocódigo do cálculo das violações via Simulação de Monte Carlo utilizada neste trabalho:

Algoritmo 3: Cálculo das violações às janelas de tempo

Entradas: (sol, T, TS, TI, TF, N);

$TMV = 0$;

$CMV = 0$;

para $i = 1$ até N **faça**

para cada $rota \in sol$ **faça**

$T_{sim} = Simulacao(T)$; ▷ Simulação tempos de viagem da rota

$TS_{sim} = Simulacao(TS)$; ▷ Simulação tempos de serviço da rota

$DI = IA(T_{sim}, TS_{sim}, TF)$; ▷ Início de atendimento dos clientes da rota

para cada $cliente \in rota$ **faça**

se $DI \leq TI$ **então**

$TMV = TMV + (TI - DI)$;

$CMV = CMV + 1$;

fim

se $DI \geq TF$ **então**

$TMV = TMV + (DI - TF)$;

$CMV = CMV + 1$;

fim

fim

fim

fim

$TMV = TMV/N$;

$CMV = CMV/N$;

Retorna TMV, CMV ;

Fim

O algoritmo tem como entrada os seguintes parâmetros: solução a ser simulada (sol), matriz de tempos de viagem (T), tempos de serviço (TS), início da janela de atendimento (TI), fim da janela de atendimento (TF) e o número de replicações (N). Já as saídas são representadas pelo tempo médio violado (TMV) e número médio de clientes com janelas de tempo violadas (CMV). O parâmetro T é uma matriz que contém o tempo de viagem entre todos os clientes da base de dados. Já os parâmetros TS , TI e TF são vetores que contém os tempos de serviço, início da janela de atendimento e fim da janela de atendimento associados a cada cliente, respectivamente.

O funcionamento do algoritmo se dá da seguinte forma: inicialmente, os valores de

TMV e CMV são definidos como 0, pois nenhuma rodada de simulação das violações ainda foi executada. Posteriormente, o algoritmo executa N rodadas de simulação e, para cada uma delas, calcula novos valores simulados para os tempos de viagem e serviço de cada rota pertencente à solução. O tempo de início de atendimento de cada cliente da rota é calculado de acordo com os novos tempos encontrados na simulação.

O próximo passo do algoritmo é a execução da comparação entre o tempo de início de atendimento de cada cliente pertencente à rota com o início e fim das janelas de tempo. Caso um veículo chegue antecipadamente ao seu destino, o tempo é contabilizado e o número de clientes violados é acrescido. O mesmo vale para casos em que o veículo chega atrasado ao seu destino.

Por fim, calculam-se as médias do tempo total violado e do número de clientes com janelas de tempo violadas, que representam o retorno da função de simulação. Estes valores são utilizados no cálculo da função de avaliação das soluções geradas durante outras etapas do algoritmo, onde existe a penalização por violação das janelas de atendimento aos clientes.

3.3 Integração entre ILS-VNS e simulações

A integração entre a meta-heurística ILS-VNS e as simulações apoia-se no fato de que soluções viáveis para a versão determinística do problema são consideradas soluções promissoras para a versão estocástica, como citado em tópicos anteriores. De forma similar a [Guimarans et al. \(2018\)](#), foi proposto um algoritmo integrado que realiza simulações rápidas durante a fase de otimização e inclui uma simulação mais profunda ao final do processo, com o objetivo de refinar a solução final e avaliar os riscos e impactos gerados pela presença da incerteza no problema.

O algoritmo tem como entrada os *inputs*, representados por parâmetros retornados pela leitura de dados, como a quantidade de clientes a serem visitados, número de veículos disponíveis e suas respectivas capacidades, tempos de viagem entre clientes, tempos de serviço, tempo do início das janelas de tempo, tempo do final das janelas de tempo e demandas. Estes parâmetros são utilizados durante as fases de geração da solução inicial, buscas locais e perturbações.

Outros parâmetros são definidos ao longo do código, como n , que indica a realização de simulações rápidas das soluções encontradas, N , que indica a realização de simulações mais profundas das soluções encontradas, r , parâmetro que guia as estruturas de vizinhança das buscas locais, além dos parâmetros referentes ao ILS, como os critérios de parada e número de trocas das perturbações.

O primeiro passo é a definição dos parâmetros n , N e r . Posteriormente, é gerada uma solução inicial que é refinada durante a primeira busca local do algoritmo, executada pelo VNS. A solução retornada pela busca local torna-se a melhor solução corrente, assim como sua função de avaliação. Uma simulação rápida desta solução é executada, retornando o melhor valor para a FA simulada do algoritmo até então.

Durante o *loop* do ILS, o critério de parada *iterMax* é definido com o valor de 1000, enquanto o número de trocas, que pode ser incrementado posteriormente, é definido como 2. Até que o critério de parada seja satisfeito, o algoritmo executa perturbações na solução corrente, seguidas de buscas locais nas soluções perturbadas. O valor da melhor FA corrente e da FA gerada são comparados e, caso haja melhora neste valor, executa-se uma simulação rápida da solução que gerou melhora, retornando o valor da FA da solução simulada. Em seguida, compara-se o valor da FA da nova solução simulada com o melhor valor da FA simulada até então. Caso haja redução da melhor FA simulada encontrada, a solução determinística e a FA determinística tornam-se as melhores correntes e são inseridas em um *pool* de soluções que geraram melhoras das FA's.

Na última etapa do algoritmo, o ILS retorna todas as soluções determinísticas que geraram melhoras nas FA's simuladas. Estas soluções passam por simulações profundas, onde são realizadas N rodadas de simulação para cada uma e, posteriormente, são comparadas. Durante esta fase, o algoritmo registra os melhores e piores cenários das simulações. Por fim, retorna-se a solução que gerou o menor valor para a FA durante as simulações.

O pseudocódigo do algoritmo integrado é apresentado no Algoritmo 4:

Algoritmo 4: Algoritmo Sim-ILS**Entradas:** (*inputs*); $n = 100;$ $N = 1000;$ $r = 4;$ $solIni, faIni \leftarrow calculaSolIni(inputs);$ **Fase Busca Local;** $solBL, faBL \leftarrow VNS(solIni, faIni, inputs, r);$ $solMelhor, faMelhor = solBL, faBl;$ $faMelhorSim \leftarrow \mathbf{simulacaoRapida}(solBL, n);$ **Fase ILS;** $iterMax = 1000;$ $iterCorrente = 0;$ $iterSemMelhora = 0;$ $nTrocas = 2;$ **enquanto** $iterCorrente \leq iterMax$ **faça** $solPert, faPert \leftarrow perturbaSol(solBL, faBL, inputs, nTrocas);$ $solBl, faBl \leftarrow VNS(solPert, faPert, inputs, r);$ **se** $faBl \leq faMelhor$ **então** $faBLSim \leftarrow \mathbf{simulacaoRapida}(solBL, n);$ **se** $faBLSim \leq faMelhorSim$ **então** $solMelhor, faMelhor \leftarrow solBL, foBL;$ $faMelhorSim \leftarrow faBLSim;$ $insere(solBL, poolSimulacao);$ **fim** **senão** $iterSemMelhora = iterSemMelhora + 1$ **fim** $iterCorrente = iterCorrente + 1;$ **se** $iterSemMelhora > 5$ **então** $nTrocas = nTrocas + 1;$ **se** $nTrocas > 5$ **então** $nTrocas = 2;$ **fim** **fim****fim****para cada** $sol \in poolSimulacao$ **faça** $faPool \leftarrow \mathbf{simulacaoProfunda}(sol, N);$ **se** $faPool \leq faMelhorSim$ **então** $solMelhor \leftarrow sol;$ $faMelhor \leftarrow faPool$ **fim****fim****Retorna** $solMelhor, faMelhor;$ **Fim**

- ▷ Simulações rápidas
- ▷ Simulações profundas
- ▷ Movimentos Busca Local

4 Resultados

O algoritmo proposto foi desenvolvido na linguagem Python, em sua versão 3.10.2, e os testes foram realizados em um computador Intel Core i5-7400 CPU 3.00GHz, com 16 GB de memória RAM instalada, utilizando o sistema operacional Windows 10 Pro.

Com o objetivo de testar o método proposto, foram utilizadas como base de dados as instâncias teste criadas por Solomon (1987). A base contém 56 problemas para o PRV com Janelas de Tempo e frota homogênea. Estes problemas foram criados de forma que vários fatores podem afetar o comportamento dos métodos utilizados para resolvê-los, tais como dados geográficos, número de clientes atendidos por um veículo e as próprias janelas de tempo.

Os problemas são divididos em 6 conjuntos: R1, R2, C1, C2, RC1 e RC2. Nas categorias R1 e R2 os dados geográficos dos clientes são gerados por uma distribuição uniforme aleatória. Nas categorias C1 e C2 os clientes são distribuídos de forma agrupada. Já as categorias RC1 e RC2 representam um *mix* das categorias anteriores, de forma que metade dos clientes são distribuídos aleatoriamente no espaço geográfico e a outra metade é agrupada. Além disso, os problemas do tipo R1, C1 e RC1 possuem janelas de atendimento curtas que, em conjunto com as baixas capacidade dos veículos, não permitem que muitos clientes sejam atendidos pelo mesmo veículo sem que as restrições sejam violadas. Em contraste, os problemas do tipo R2, C2 e RC2 possuem janelas de atendimento mais longas e veículos com capacidades maiores, de forma que vários clientes podem ser atendidos por um mesmo veículo.

Apesar de ter sido elaborada décadas atrás, a escolha dessa base de dados é justificada por sua grande utilização em trabalhos científicos sobre o PRVJT atuais, o que permite comparar as soluções encontradas com outras da literatura. Além disso, é uma base que pode ser facilmente adaptada para outras variações do problema, permitindo a realização de diferentes abordagens.

Nas tabelas 4.1, 4.2 e 4.3 são apresentados os melhores resultados já encontrados na literatura para os problemas R1, R2, C1, C2, RC1 e RC2 com 25 clientes, assim como o número de veículos utilizados para atendê-los. Além disso, são apresentados os resultados obtidos pelo método sem considerar a incerteza nos tempos de viagem e serviço. As duas últimas colunas da tabela referem-se à comparação dos resultados obtidos pelo método ILS proposto com os resultados da literatura, onde o GAP representa a diferença percentual entre o valor das soluções encontradas e ΔNV representa a diferença de veículos utilizados para tais soluções.

Tabela 4.1 – Resultados determinísticos para os problemas do tipo R1 e R2 com 25 clientes

Problema	Literatura		Resultados		GAP (%)	ΔNV
	Melhor valor	NV	Melhor valor	NV		
R101.25	617,10	8	619,17	8	0,34%	0
R102.25	547,10	7	548,10	7	0,18%	0
R103.25	454,60	5	464,82	6	2,25%	1
R104.25	416,90	4	437,08	5	4,84%	1
R105.25	530,50	6	531,53	6	0,19%	0
R106.25	465,40	3	466,48	5	0,23%	2
R107.25	424,30	4	425,26	4	0,23%	0
R108.25	397,30	4	398,29	4	0,25%	0
R109.25	441,30	5	442,62	5	0,30%	0
R110.25	444,10	4	447,47	5	0,76%	1
R111.25	428,80	5	431,11	5	0,54%	0
R112.25	393,00	4	394,10	4	0,28%	0
R201.25	463,30	4	464,37	4	0,23%	0
R202.25	410,50	4	437,15	4	6,49%	0
R203.25	391,40	3	409,51	3	4,63%	0
R204.25	355,00	2	378,24	2	6,55%	0
R205.25	393,00	3	400,36	3	1,87%	0
R206.25	374,40	3	388,67	3	3,81%	0
R207.25	361,60	3	397,64	2	9,97%	-1
R208.25	328,20	1	329,33	1	0,34%	0
R209.25	370,70	2	377,39	3	1,80%	1
R210.25	404,60	3	405,47	3	0,22%	0
R211.25	350,90	2	363,92	1	3,71%	-1

Tabela 4.2 – Resultados determinísticos para os problemas do tipo C1 e C2 com 25 clientes

Problema	Literatura		Resultados		GAP (%)	ΔNV
	Melhor valor	NV	Melhor valor	NV		
C101.25	191,3	3	191,81	3	0,27%	0
C102.25	190,3	3	190,73	3	0,23%	0
C103.25	190,3	3	190,73	3	0,23%	0
C104.25	186,9	3	190,74	3	2,05%	0
C105.25	191,3	3	191,81	3	0,27%	0
C106.25	191,3	3	191,81	3	0,27%	0
C107.25	191,3	3	191,81	3	0,27%	0
C108.25	191,3	3	191,81	3	0,27%	0
C109.25	191,3	3	191,81	3	0,27%	0
C201.25	214,7	2	215,54	2	0,39%	0
C202.25	214,7	2	223,31	1	4,01%	-1
C203.25	214,7	2	223,31	1	4,01%	-1
C204.25	213,1	2	223,35	1	4,81%	-1
C205.25	214,7	2	237,14	2	10,45%	0
C206.25	214,7	2	236,51	2	10,16%	0
C207.25	214,5	2	215,33	2	0,39%	0
C208.25	214,5	2	215,37	2	0,41%	0

Tabela 4.3 – Resultados determinísticos para os problemas do tipo RC1 e RC2 com 25 clientes

Problema	Literatura		Resultados		GAP (%)	ΔNV
	Melhor valor	NV	Melhor valor	NV		
RC101.25	461,10	4	529,82	6	14,90%	2
RC102.25	351,80	3	401,99	4	14,27%	1
RC103.25	332,80	3	388,37	4	16,70%	1
RC104.25	306,60	3	307,13	3	0,17%	0
RC105.25	411,30	4	501,81	5	22,01%	1
RC106.25	345,50	3	408,84	4	18,33%	1
RC107.25	298,30	3	363,94	4	22,00%	1
RC108.25	294,50	3	356,67	4	21,11%	1
RC201.25	360,20	3	420,49	4	16,74%	1
RC202.25	338,00	3	338,82	3	0,24%	0
RC203.25	326,90	3	397,91	2	21,72%	-1
RC204.25	299,70	3	342,01	2	14,12%	-1
RC205.25	338,00	3	396,41	4	17,28%	1
RC206.25	324,00	3	388,26	3	19,83%	0
RC207.25	298,30	3	349,13	2	17,04%	-1
RC208.25	269,10	2	320,42	2	19,07%	0

A tabela 4.1 indica que os resultados das soluções obtidas para os problemas R1 e R2 se aproximam muito dos melhores resultados já encontrados na literatura, onde para os problemas do tipo R1 o GAP médio foi de 0,87%, enquanto para R2 foi de 3,60%. Na maioria dos casos, o GAP se manteve entre 0 e 1%, o que confirma a eficiência do método para as instâncias R1 e R2. De forma similar, na tabela 4.2 é possível perceber que os resultados também foram satisfatórios para os problemas do tipo C1 e C2, onde a diferença percentual média de C1 foi de 0,46% e em C2 foi de 4,33%. Por outro lado, a tabela 4.3 indica que o método não se mostrou eficiente para resolução dos problemas do tipo RC1 e RC2, onde o GAP foi consideravelmente maior quando comparado com os resultados dos problemas R1, R2, C1 e C2. O GAP médio para os problemas do tipo RC1 foi de 16,19%, enquanto para os problemas do tipo foi de 15,76%.

Nas tabelas 4.4, 4.5 e 4.6 são apresentados os resultados referentes à fase de simulação do algoritmo. Durante a realização das simulações profundas do método proposto foram registrados os melhores e os piores cenários encontrados para as soluções obtidas, assim como seus respectivos valores para o número médio de clientes violados e tempo médio violado. Além disso, as tabelas a seguir apresentam o GAP entre os valores do melhor e do pior cenário em comparação com os valores encontrados durante a fase anterior, que considera todos os parâmetros como determinísticos. Por fim, apresentam-se as médias dos valores encontrados durante as simulações profundas.

Tabela 4.4 – Resultados da simulação para os problemas do tipo R1 e R2 com 25 clientes

Problema	Melhor cenário				Pior cenário				Média
	Valor	GAP (%)	CMV	TMV	Valor	GAP (%)	CMV	TMV	
R101.25	690,81	11,57%	0,98	1,14	2484,13	301,20%	2,61	18,27	1201,90
R102.25	486,87	-11,17%	0,00	0,00	1325,68	141,87%	1,46	7,66	704,02
R103.25	421,56	-9,31%	0,00	0,00	938,27	101,86%	0,65	4,55	494,46
R104.25	397,07	-9,15%	0,00	0,00	585,77	34,02%	0,29	1,42	441,20
R105.25	509,77	-4,09%	0,16	0,25	1683,34	216,70%	2,15	11,14	757,44
R106.25	451,65	-3,18%	0,15	0,29	3009,29	545,11%	2,57	25,17	1216,57
R107.25	403,22	-5,18%	0,01	0,02	1168,69	174,82%	0,93	7,38	503,20
R108.25	392,08	-1,56%	0,05	0,10	3990,53	901,92%	2,13	35,64	1045,20
R109.25	411,76	-6,97%	0,06	0,05	1389,54	213,94%	1,92	9,23	608,92
R110.25	434,20	-2,97%	0,07	0,11	1534,98	243,04%	1,67	10,69	757,71
R111.25	390,07	-9,52%	0,00	0,00	1143,02	165,13%	0,82	6,67	531,87
R112.25	355,74	-9,73%	0,01	0,01	1419,32	260,14%	0,94	10,15	546,02
R201.25	406,84	-12,39%	0,00	0,00	727,06	56,57%	0,63	2,51	492,22
R202.25	401,33	-8,19%	0,00	0,00	1238,80	183,38%	0,78	7,55	557,88
R203.25	365,27	-10,80%	0,00	0,00	1089,11	165,95%	0,74	6,79	484,91
R204.25	330,58	-12,60%	0,00	0,00	1222,52	223,21%	0,72	8,15	473,39
R205.25	360,05	-10,07%	0,01	0,02	2731,24	582,20%	1,75	23,37	740,35
R206.25	362,44	-6,75%	0,04	0,08	1377,73	254,47%	1,02	9,91	536,97
R207.25	361,51	-9,09%	0,00	0,00	1609,66	304,80%	0,76	11,82	576,97
R208.25	298,49	-9,36%	0,00	0,00	3389,24	929,13%	1,00	30,34	702,49
R209.25	334,77	-11,29%	0,00	0,00	572,64	51,74%	0,52	1,89	389,81
R210.25	354,25	-12,63%	0,00	0,00	440,33	8,60%	0,00	0,00	405,69
R211.25	306,23	-15,85%	0,00	0,00	1788,18	391,37%	0,95	14,05	445,20

Tabela 4.5 – Resultados da simulação para os problemas do tipo C1 e C2 com 25 clientes

Problema	Melhor cenário				Pior cenário				Média
	Valor	GAP (%)	CMV	TMV	Valor	GAP (%)	CMV	TMV	
C101.25	171,12	-10,79%	0,01	0,00	49417,46	>1000%	10,01	492,26	11333,87
C102.25	172,79	-9,41%	0,00	0,00	25579,44	>1000%	7,00	253,99	4338,30
C103.25	172,48	-9,57%	0,00	0,00	13054,54	>1000%	4,00	128,46	3230,54
C104.25	164,26	-13,88%	0,01	0,00	10612,38	>1000%	3,00	104,32	1591,38
C105.25	153,08	-20,19%	0,00	0,00	32213,05	>1000%	7,00	320,36	2663,11
C106.25	167,45	-12,70%	0,00	0,00	60498,69	>1000%	14,00	603,02	9028,63
C107.25	158,07	-17,59%	0,00	0,00	20960,47	>1000%	6,38	207,60	896,51
C108.25	151,25	-21,15%	0,00	0,00	18047,55	>1000%	4,01	178,20	681,68
C109.25	157,25	-18,02%	0,00	0,00	223,53	16,54%	0,00	0,00	191,99
C201.25	191,82	-11,00%	0,00	0,00	8852,34	>1000%	5,49	86,25	300,81
C202.25	193,10	-13,53%	0,00	0,00	181520,89	>1000%	12,71	1812,86	14758,60
C203.25	197,14	-11,72%	0,00	0,00	116898,62	>1000%	0,00	1166,30	11460,30
C204.25	198,33	-11,20%	0,00	0,00	4211,41	>1000%	1,00	39,75	325,41
C205.25	202,54	-14,59%	0,00	0,00	274,09	15,58%	0,00	0,00	238,14
C206.25	196,64	-16,86%	0,00	0,00	266,99	12,89%	0,00	0,00	237,47
C207.25	188,05	-12,67%	0,00	0,00	247,32	14,86%	0,00	0,00	216,34
C208.25	194,10	-9,88%	0,00	0,00	14017,91	>1000%	3,93	137,99	576,76

Tabela 4.6 – Resultados da simulação para os problemas do tipo RC1 e RC2 com 25 clientes

Problema	Melhor cenário				Pior cenário				Média
	Valor	GAP (%)	CMV	TMV	Valor	GAP (%)	CMV	TMV	
RC101.25	495,02	-6,57%	0,02	0,01	1759,65	232,12%	2,26	12,24	758,84
RC102.25	381,65	-5,06%	0,10	0,13	6479,88	>1000%	4,00	60,80	1044,66
RC103.25	361,42	-6,94%	0,04	0,10	2721,05	600,63%	2,04	23,11	863,21
RC104.25	270,93	-11,79%	0,03	0,03	2530,42	723,89%	1,31	22,24	701,69
RC105.25	466,01	-7,13%	0,01	0,01	4286,92	754,29%	3,14	37,70	1121,50
RC106.25	362,69	-11,29%	0,10	0,08	2149,10	425,66%	1,59	17,58	553,40
RC107.25	314,14	-13,68%	0,00	0,00	2312,52	535,41%	1,73	19,28	702,89
RC108.25	320,03	-10,27%	0,00	0,00	962,54	169,87%	1,66	6,46	425,18
RC201.25	357,85	-14,90%	0,00	0,00	477,80	13,63%	0,00	0,00	420,30
RC202.25	297,01	-12,34%	0,00	0,00	390,24	15,18%	0,00	0,00	340,24
RC203.25	342,25	-13,99%	0,00	0,00	438,40	10,18%	0,00	0,00	400,63
RC204.25	293,46	-14,20%	0,00	0,00	391,36	14,43%	0,00	0,00	342,90
RC205.25	338,62	-14,58%	0,00	0,00	457,38	15,38%	0,00	0,00	393,73
RC206.25	355,26	-8,50%	0,00	0,00	1786,97	360,25%	1,65	14,07	589,05
RC207.25	289,04	-17,21%	0,00	0,00	395,87	13,39%	0,00	0,00	347,41
RC208.25	261,31	-18,45%	0,00	0,00	5542,95	>1000%	1,90	52,17	1060,59

A tabela 4.4 mostra que, na maior parte dos casos, os melhores resultados encontrados durante as simulações para os problemas do tipo R1 e R2 superam os resultados quando considerados os parâmetros determinísticos. Apenas em um dos casos essa situação não se confirmou, onde a simulação indica que, em média, a solução violaria aproximadamente 1 cliente. Os piores cenários das simulações para estes problemas não se mostraram tão pessimistas, o que reflete médias dos cenários não muito distante dos resultados determinísticos. Na tabela 4.5 é possível visualizar que os melhores cenários dos problemas C1 e C2 sempre apresentam menores valores quando comparados aos seus respectivos valores determinísticos. Entretanto, quando analisados os piores cenários destes problemas, nota-se uma grande amplitude dos valores. Por fim, a tabela 4.6 indica que, de forma semelhante aos problemas C1 e C2, os problemas RC1 e RC2 possuem melhores cenários que sempre superam seus resultados determinísticos, porém com uma menor amplitude e variabilidade dos piores cenários encontrados.

As tabelas a seguir são um compilado dos resultados obtidos para as instâncias R1, R2, C1, C2, RC1 e RC2 com 50 clientes. Em decorrência do alto esforço computacional necessário para realizar testes em instâncias maiores, foram realizados testes apenas para as 5 primeiras bases de dados de cada problema. De forma similar às tabelas anteriores, os resultados da fase onde os parâmetros são determinísticos para instâncias com 50 clientes são apresentados na tabela 4.7. Já os resultados para a fase de simulação são apresentados na tabela 4.8

Tabela 4.7 – Resultados determinísticos para os problemas R1, R2, C1, C2, RC1 e RC2 com 50 clientes

Problema	Literatura		Resultados		GAP	DeltaNV
	Valor	NV	Valor	NV		
R101.50	1044,00	12	1061,64	13	1,69%	1
R102.50	909,00	11	924,16	12	1,67%	1
R103.50	772,90	9	802,09	10	3,78%	1
R104.50	625,40	6	659,1	7	5,39%	1
R105.50	899,30	9	946,26	11	5,22%	2
R201.50	791,90	6	884,36	8	11,68%	2
R202.50	698,50	5	744,19	6	6,54%	1
R203.50	605,30	5	652,99	5	7,88%	0
R204.50	506,40	2	565,95	3	11,76%	1
R205.50	690,10	4	768,3	6	11,33%	2
C101.50	362,4	5	418,41	6	15,46%	1
C102.50	361,4	5	411,73	5	13,93%	0
C103.50	361,4	5	362,17	5	0,21%	0
C104.50	358	5	366,00	5	2,23%	0
C105.50	362,4	5	363,24	5	0,23%	0
C201.50	360,2	3	361,79	2	0,44%	-1
C202.50	360,2	3	373,23	3	3,62%	0
C203.50	359,8	3	379,49	3	5,47%	0
C204.50	350,1	2	399,64	2	14,15%	0
C205.50	359,8	3	361,41	3	0,45%	0
RC101.50	944,00	8	985,07	10	4,35%	2
RC102.50	822,50	7	960,38	10	16,76%	3
RC103.50	710,90	6	808,67	8	13,75%	2
RC104.50	545,80	5	640,15	6	17,29%	1
RC105.50	855,30	8	984,00	10	15,05%	2
RC201.50	684,80	5	846,34	8	23,59%	3
RC202.50	613,60	5	759,38	6	23,76%	1
RC203.50	555,30	4	699,44	4	25,96%	0
RC204.50	444,20	3	548,96	2	23,58%	-1
RC205.50	630,20	5	818,69	6	29,91%	1

Tabela 4.8 – Resultados da simulação para os problemas R1, R2, C1, C2, RC1 e RC2 com 50 clientes

Problema	Melhor cenário				Pior cenário				Média
	Valor	GAP (%)	CMV	TMV	Valor	GAP (%)	CMV	TMV	
R101.50	1123,76	5,85%	0,63	1,15	6043,30	469,24%	5,43	49,65	2655,61
R102.50	1216,86	31,67%	1,67	3,28	7434,23	704,43%	4,58	65,25	2737,40
R103.50	809,07	0,87%	0,42	0,59	7023,16	775,61%	4,21	61,57	1992,73
R104.50	679,58	3,11%	0,13	0,27	7042,86	968,56%	4,11	63,89	2133,39
R105.50	1081,89	14,33%	0,68	1,73	5881,82	521,59%	4,49	49,19	1889,86
R201.50	918,91	3,91%	0,21	0,50	2744,87	210,38%	1,80	17,90	1531,33
R202.50	698,10	-6,19%	0,00	0,00	1068,87	43,63%	0,42	3,32	762,28
R203.50	596,83	-8,60%	0,00	0,00	696,14	6,61%	0,00	0,00	653,84
R204.50	515,28	-8,95%	0,00	0,00	610,31	7,84%	0,00	0,00	566,69
R205.50	717,81	-6,57%	0,00	0,00	1252,54	63,03%	0,87	4,69	840,73
C101.50	388,81	-7,07%	0,01	0,00	71883,09	>1000%	19,23	714,72	16133,81
C102.50	382,15	-7,18%	0,00	0,00	56659,07	>1000%	12,95	562,60	11907,70
C103.50	320,52	-11,50%	0,00	0,00	45902,67	>1000%	8,77	455,49	11467,94
C104.50	339,03	-7,37%	0,01	0,01	26186,79	>1000%	4,00	257,95	4163,87
C105.50	324,04	-10,79%	0,00	0,00	46797,79	>1000%	9,12	464,3	8717,8
C201.50	332,40	-8,12%	0,00	0,00	253108,39	>1000%	20,53	2527,34	16822,47
C202.50	343,05	-8,09%	0,00	0,00	184013,46	>1000%	18,64	1836,24	18981,81
C203.50	336,26	-11,39%	0,00	0,00	44758,18	>1000%	5,99	444,01	5098,25
C204.50	365,83	-8,46%	0,00	0,00	23581,72	>1000%	2,00	234,41	2846,42
C205.50	328,67	-9,06%	0,00	0,00	175882,31	>1000%	20,00	1755,08	2575,92
RC101.50	920,48	-6,56%	0,47	0,24	5806,12	489,41%	4,19	47,43	2206,49
RC102.50	969,47	0,95%	0,56	0,59	5254,41	447,12%	2,70	43,03	2356,89
RC103.50	799,20	-1,17%	0,13	0,37	6853,03	747,44%	4,09	60,38	2084,73
RC104.50	557,55	-12,90%	0,00	0,00	5527,46	763,46%	2,64	48,55	1411,36
RC105.50	987,26	0,33%	0,62	0,81	5580,76	467,15%	4,51	45,59	2260,69
RC201.50	759,60	-10,25%	0,01	0,01	5522,57	552,52%	2,87	46,17	1721,14
RC202.50	682,39	-10,14%	0,00	0,00	3803,99	400,93%	1,01	30,18	1113,18
RC203.50	608,29	-13,03%	0,02	0,02	1121,79	60,38%	0,95	3,91	712,41
RC204.50	489,05	-10,91%	0,01	0,04	2325,94	323,70%	0,94	17,62	730,06
RC205.50	773,80	-5,48%	0,02	0,01	3965,93	384,42%	1,34	31,74	1510,74

A tabela 4.7 indica que, de forma geral, os resultados para instâncias com 50 clientes foram ligeiramente piores quando comparados aos resultados para instâncias com 25 clientes. O GAP médio para os problemas do tipo R1 foi de 3,55%, enquanto para os problemas do tipo R2 foi de 9,84%. Já para os problemas do tipo C1 o GAP médio foi de 6,41%, enquanto para C2 foi de 4,83%. Assim como nas instâncias que possuem 25 clientes, os resultados para instâncias do tipo RC1 e RC2 não foram tão satisfatórios, com GAP médio de 13,44% para RC1 e 25,36% para RC2.

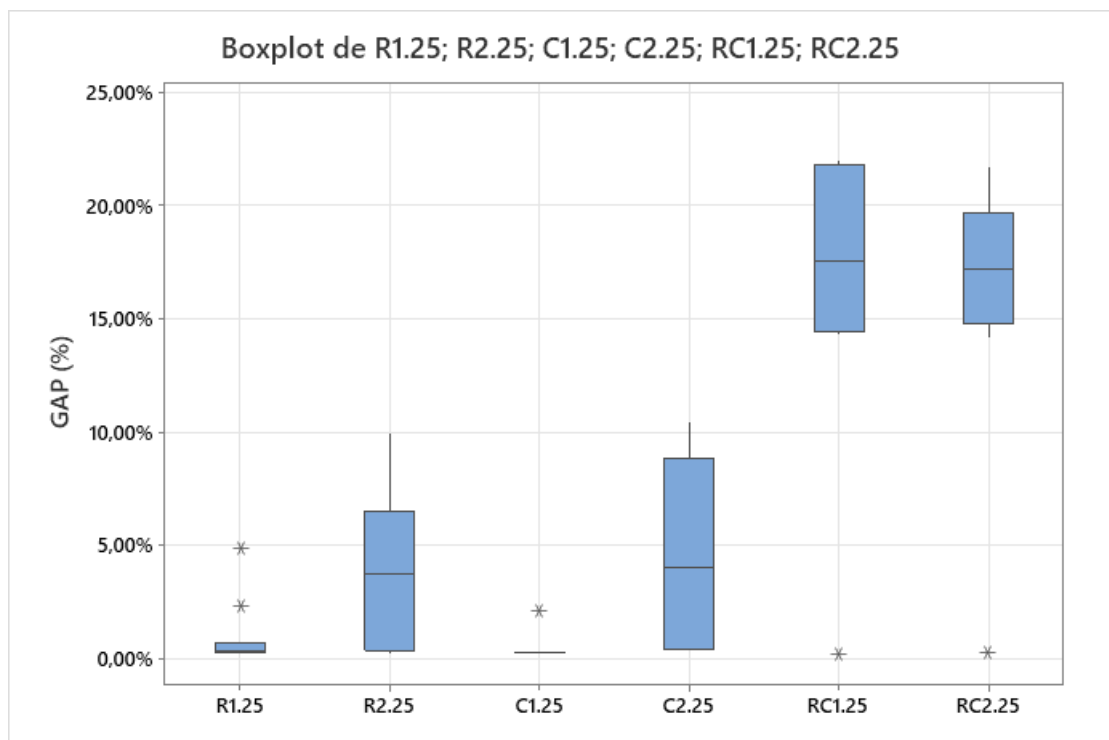
Os resultados das simulações para instâncias com 50 clientes, apresentados na tabela 4.8, indicam que, mesmo em seus melhores cenários, os resultados para os problemas do tipo R1 são piores que seus respectivos valores determinísticos. Já para os problemas do tipo R2, os melhores cenários quase sempre superam os resultados quando não consideradas as incertezas, com apenas um caso sendo pior. De forma semelhante às instâncias com 25 clientes, os resultados das simulações para os problemas do tipo C1 e C2 sempre apresentam melhores cenários que superam seus resultados determinísticos, além de piores cenários que possuem um GAP muito elevado,

como mostra a tabela. Os problemas do tipo RC1 e RC2 apresentaram melhores cenários que superaram seus resultados determinísticos em 8 dos 10 problemas testados, onde, diferentemente dos problemas do tipo C1 e C2, os piores cenários possuem GAPS não tão grandes.

4.1 Análise dos resultados obtidos

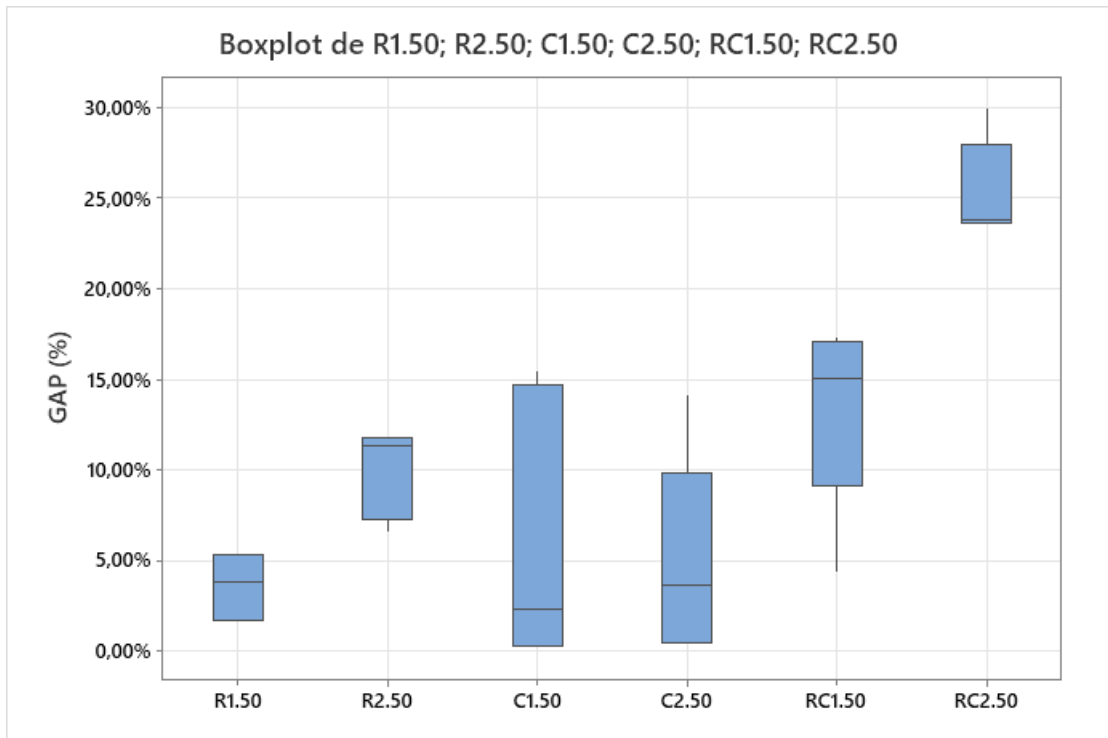
Com o objetivo de proporcionar uma melhor compreensão dos resultados, foram elaborados diversos *boxplots* para os dados apresentados nas tabelas anteriores. Os dados analisados foram o GAP entre os resultados obtidos pelo método proposto e os resultados da literatura e a diferença de veículos utilizados nas soluções encontradas em comparação com o número de veículos utilizados nas melhores soluções da literatura. Os *boxplots* são apresentados nas figuras 4.1, 4.2, 4.3 e 4.4 a seguir:

Figura 4.1 – *Boxplot* do GAP entre os resultados obtidos e resultados da literatura para as instâncias de 25 clientes



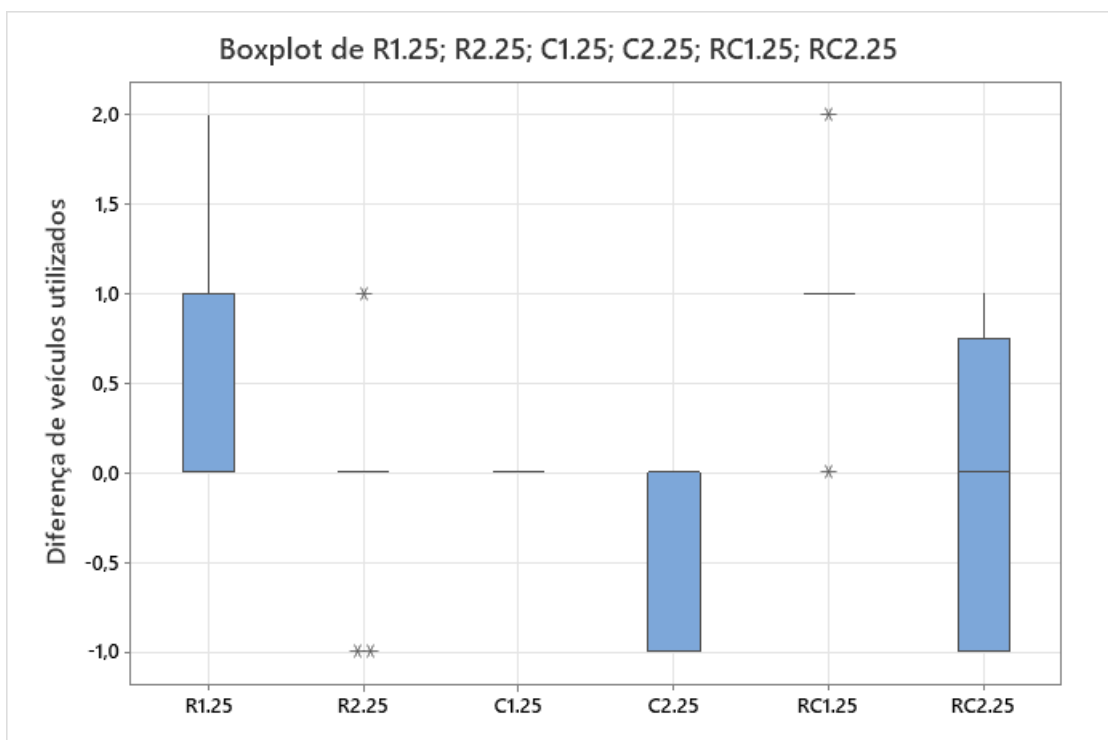
Fonte: Autor

Figura 4.2 – *Boxplot* do GAP entre os resultados obtidos e resultados da literatura para as instâncias de 50 clientes



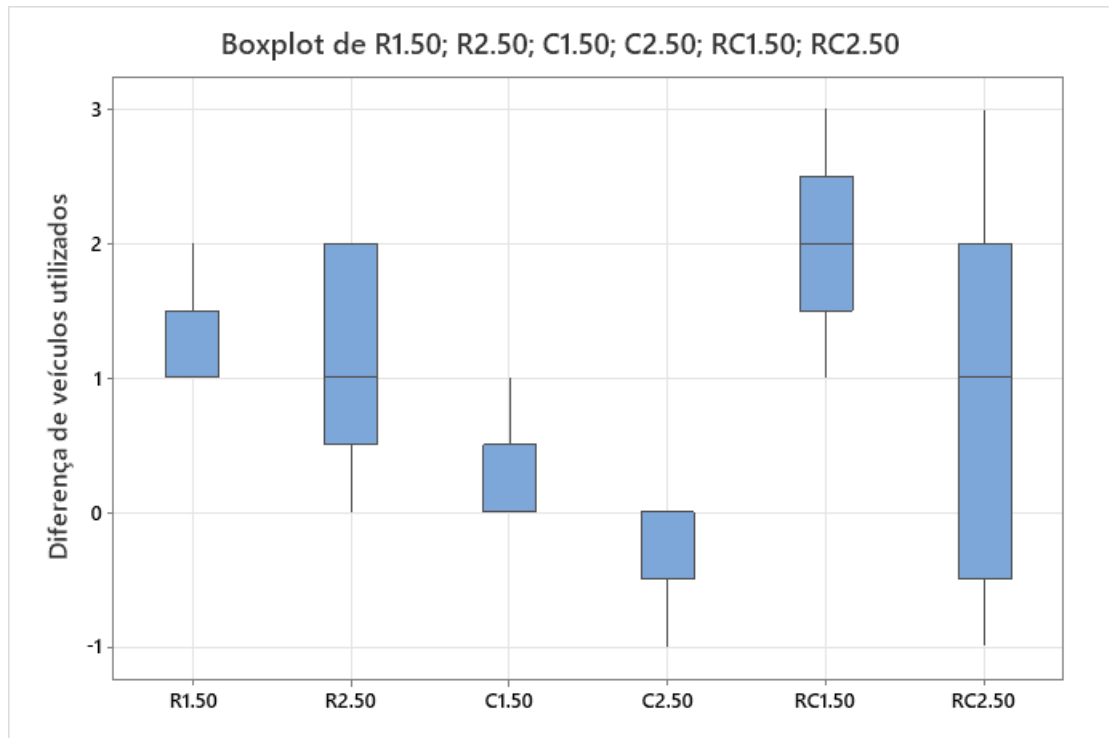
Fonte: Autor

Figura 4.3 – *Boxplot* da diferença entre os veículos utilizados nas soluções obtidas e nas soluções da literatura para instâncias de 25 clientes



Fonte: Autor

Figura 4.4 – *Boxplot* da diferença entre os veículos utilizados nas soluções obtidas e nas soluções da literatura para instâncias de 50 clientes



Fonte: Autor

A figura 4.1 mostra que os resultados obtidos para os problemas R1 e C1 em instâncias com 25 clientes se aproximaram muito dos melhores resultados da literatura, sem grandes variações, sendo que em R1 o GAP máximo foi de 4,84%, enquanto em C1 foi de apenas 2,05%, ambos valores discrepantes quando comparados com o restante dos resultados. Os resultados obtidos para os problemas R2 e C2 também foram satisfatórios, sendo que em R2 a média dos GAPS foi de 3,60% e em C2 a média foi de 4,33%. Em contrapartida, o método proposto não se mostrou tão eficiente para resolução dos problemas RC1 e RC2, apresentando diferenças que podem variar de aproximadamente 14% até 22%.

Já na figura 4.2 pode-se observar, de forma geral, que os resultados obtidos para instâncias com 50 clientes foram piores que os das instâncias com 25 clientes. Os melhores resultados obtidos foram os do problema R1, onde a maioria dos GAPS se manteve abaixo dos 5%. As instâncias R2, C1, C2 e RC1 não apresentaram GAPS maiores que 20%, enquanto RC2 apresentou um GAP médio de aproximadamente 25%.

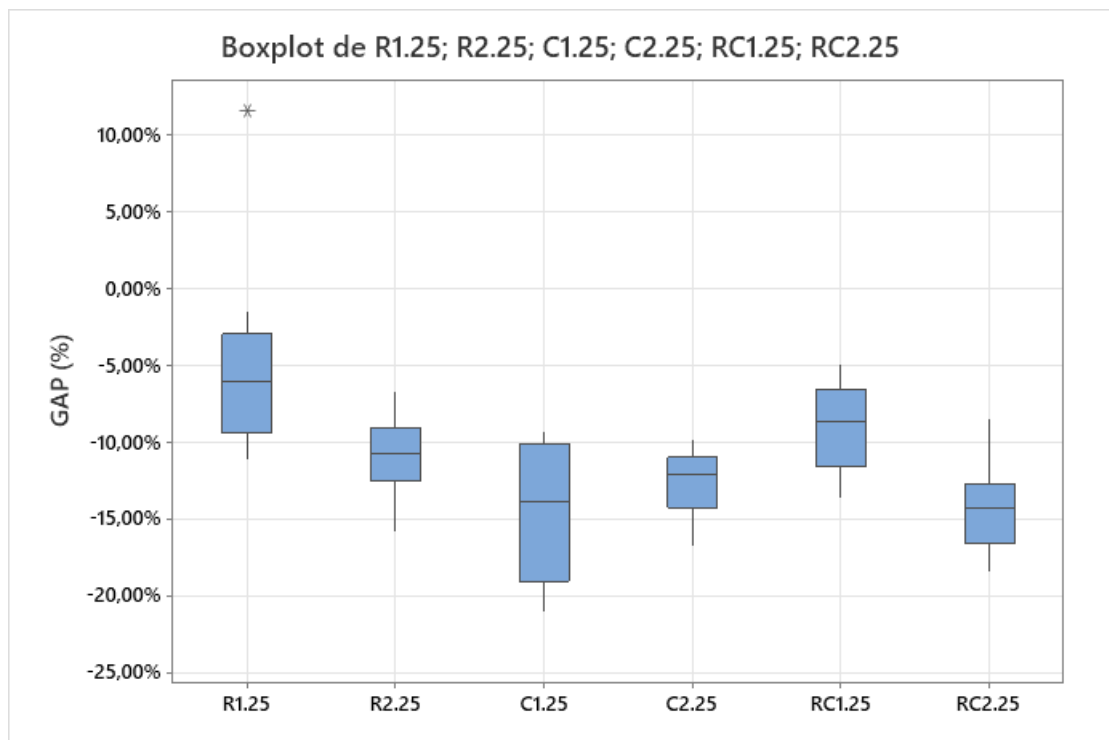
Percebe-se que, apesar dos problemas do tipo R1, C1 e RC1 apresentarem janelas de tempo mais curtas e veículos com menores capacidades, os resultados encontrados foram, de forma geral, melhores dos que os obtidos para os problemas R2, C2 e RC2, que apresentam janelas de tempos largas e veículos com maiores capacidades.

A figura 4.3, mostra que a quantidade de veículos utilizados pelas soluções dos problemas R2 e C1 em instâncias com 25 clientes foram basicamente as mesmas da literatura. Já as instâncias

R1, C2 e RC2 apresentaram grandes amplitudes, onde houveram casos em que o número de veículos utilizados foi maior e casos em que foi menor, com destaque para RC2. RC1 utilizou, na maioria dos casos, 1 veículo a mais do que nas soluções da literatura. Já a figura 4.4 revela que, de forma geral, as instâncias com 50 clientes apresentaram uma grande diferença na quantidade de veículos utilizados quando comparadas as soluções, onde na maior parte dos casos as soluções obtidas neste trabalho utilizaram mais veículos.

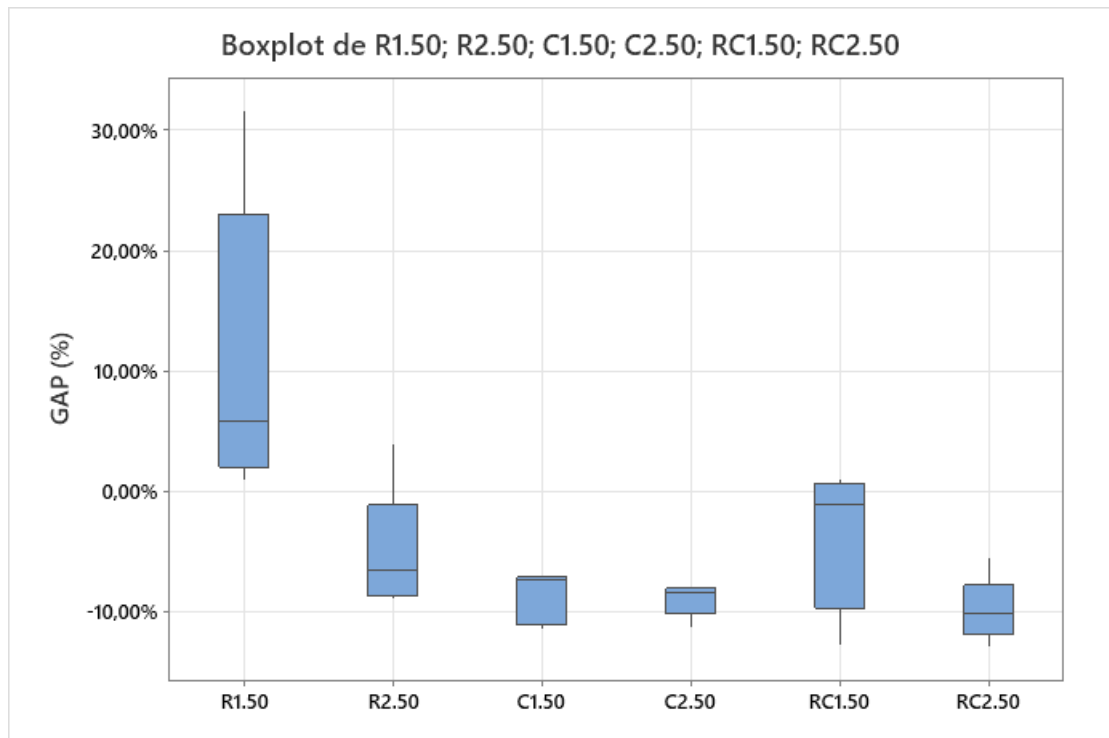
Logo a seguir são apresentadas os *boxplots* que ilustram o GAP entre os melhores resultados obtidos durante a fase de simulações e os resultados obtidos durante a fase onde os parâmetros são determinísticos para instâncias de 25 clientes e 50 clientes, respectivamente.

Figura 4.5 – *Boxplot* do GAP entre os resultados obtidos durante a fase simulações e os resultados obtidos durante a fase de otimização para as instâncias de 25 clientes



Fonte: Autor

Figura 4.6 – *Boxplot* do GAP entre os resultados obtidos durante a fase simulações e os resultados obtidos durante a fase de otimização para as instâncias de 50 clientes



Fonte: Autor

A figura 4.5 indica que os resultados dos melhores cenários das simulações para instâncias com 25 e 50 clientes superaram seus respectivos resultados determinísticos na maioria dos casos. Essa situação já era esperada, uma vez que são realizadas um grande número de rodadas de simulação para uma mesma solução determinística, possibilitando que em algum caso sejam simulados tempos que, além de serem menores que os determinísticos, não são penalizados pela violação de nenhuma restrição, seja de capacidade do veículo ou janelas de atendimento. Nota-se, entretanto, que nos problemas do tipo R1 houveram casos onde os resultados obtidos nas simulações se mostraram piores que os obtidos considerando os tempos de viagem e serviço determinísticos, sobretudo para instâncias de 50 clientes. Outro evento observado foi que, diferentemente dos resultados da fase determinística, os resultados das simulações para os problemas R2, C2 e RC2 foram melhores que dos problemas R1, C1 e RC1, que possuem janelas de tempo mais curtas e veículos com menores capacidades.

Em decorrência da grande variabilidade apresentada pelos resultados dos piores cenários durante a fase de simulações, não foram elaborados gráficos para representar o GAP entre os piores cenários e os resultados do problema em sua forma determinística. Percebe-se que os piores cenários para os problemas do tipo R1 e R2, que possuem clientes distribuídos aleatoriamente, não se distanciam tanto de seus melhores cenários simulados, em comparação com os problemas do tipo C1 e C2, que apresentaram uma grande diferença entre os piores e melhores cenários.

5 Conclusão e trabalhos futuros

Este trabalho abordou o PRVJT com incerteza nos tempos de viagem e serviço. Trata-se de um tema atual, que possui grande importância para o meio acadêmico e para o meio empresarial, sobretudo na área da logística. Visando se familiarizar com o assunto, foi necessária uma profunda revisão da literatura. Dessa forma, foi possível ter mais assertividade na escolha do método para resolver este problema, assim como nos parâmetros de ajuste do algoritmo. Foi proposto um algoritmo baseado na meta-heurística ILS que utiliza o VNS em sua fase de buscas locais. A incerteza nos tempos de viagem e serviço foi incorporada ao problema através das Simulações de Monte Carlo, de forma que os tempos determinísticos foram substituídos por amostras aleatórias de funções densidade de probabilidade pré-determinadas.

A integração entre a meta-heurística ILS-VNS e as simulações baseia-se na concepção de que, em ambientes onde a incerteza é baixa ou moderada, as soluções viáveis para a versão determinística do problema são consideradas soluções promissoras para sua versão estocástica. Neste contexto, em um primeiro momento o algoritmo resolve o problema em sua forma determinística, realizando simulações rápidas das soluções encontradas, para depois executar simulações profundas das soluções que apresentaram melhora da função de avaliação durante o *loop* do ILS. Estas soluções são refinadas e comparadas e a que apresenta um menor custo torna-se a função a ser retornada e analisada pelo algoritmo. Além de encontrar a solução que apresenta os menores custos em contextos de incerteza, o método proposto teve como objetivo secundário avaliar os impactos da incerteza da solução final. Sendo assim, esta solução passa por uma nova rodada de simulações que visa obter seu melhor e seu pior cenário, avaliando o número médio de clientes violados e o tempo médio violado.

O método proposto foi validado através da comparação dos resultados obtidos com resultados encontrados na literatura para instâncias de pequeno porte, onde a diferença entre os resultados foi mínima ou nenhuma. As bases de dados utilizadas para comparação dos resultados finais deste trabalho são conhecidas e amplamente utilizada na literatura. O algoritmo se mostrou eficiente para resolução de instâncias com clientes distribuídos aleatoriamente e para instâncias com clientes agrupados. Entretanto, os resultados foram consideravelmente piores para instâncias que possuem essas duas características em conjunto. Ao avaliar a incorporação da incerteza ao problema percebeu-se que quase sempre os melhores cenários das soluções simuladas irão apresentar menores tempos e custos quando comparados às soluções do problema em sua forma determinística, o que faz sentido. Contudo, ao analisar os piores cenários das soluções simuladas, é possível visualizar que a presença de parâmetros estocásticos pode inviabilizar uma solução. Ressalta-se, assim, a importância de se incorporar a incerteza à problemas deste tipo, uma vez que ela é intrínseca aos processos da vida real e não considerá-la pode gerar graves transtornos.

Como proposta para trabalhos futuros, indica-se o desenvolvimento de outras metodologias para resolução deste problema para efeitos de comparação. Além disso, indica-se a alteração dos parâmetros estatísticos apresentados neste trabalho, a fim de avaliar como o algoritmo se comporta em ambientes onde a incerteza é menor ou maior. Por fim, aponta-se a resolução do PRVJT com incerteza nos tempos de viagem e serviço e frota heterogênea.

Referências

- ANDO, N.; TANIGUCHI, E. Travel time reliability in vehicle routing and scheduling with time windows. *Networks and Spatial Economics*, v. 6, 2006. ISSN 1566113X.
- AQUINO, R. de F. Algoritmos de otimização multi-objetivo para o problema de roteamento de veículos com janelas de tempo. *Universidade Federal de Viçosa*, 2015.
- ARENALES, M.; ARMENTANO, V.; MORABITO, R.; YANASSE, H. *Pesquisa Operacional para Cursos de Engenharia*. [S.l.: s.n.], 2007. ISSN 16130073.
- BALLOU, R. H. Gerenciamento da cadeia de suprimentos: Logística empresarial. *Grupo A*, 2018.
- BEZERRA, T. F. B. Comparação de meta-heurísticas aplicadas a um problema de escalonamento em máquinas paralelas. *Instituto Superior de Engenharia do Porto*, 2020.
- CORDEAU, J. F.; GENDREAU, M.; LAPORTE, G.; POTVIN, J. Y.; SEMET, F. A guide to vehicle routing heuristics. *Journal of the Operational Research Society*, v. 53, 2002. ISSN 14769360.
- DANTZIG, G. B.; RAMSER, J. H. The truck dispatching problem. *Management Science*, v. 6, 1959. ISSN 0025-1909.
- DONATELLI, G. D.; KONRATH, A. C. Simulação de monte carlo na avaliação de incertezas de medição. *Revista de Ciência Tecnologia*, v. 13, 2005.
- GENDREAU, M.; LAPORTE, G.; SEGUIN, R. Stochastic vehicle routing. *European Journal of Operational Research*, v. 88, n. 1, p. 3–12, 1996. ISSN 0377-2217. Disponível em: <<https://www.sciencedirect.com/science/article/pii/037722179500050X>>.
- GENDREAU, M. J.-Y. P. Handbook of metaheuristic. *International Series in Operations Research and Management Science*, v. 272, 2019. ISSN 08848289.
- GOLDBARG, M. C.; GOLDBARG, E. G.; LUNA, H. P. L. Otimização combinatória e meta-heurísticas - algoritmos e aplicações. *Grupo GEN*, 2015.
- GONZALEZ-MARTIN, S.; JUAN, A. A.; RIERA, D.; ELIZONDO, M. G.; RAMOS, J. J. A simheuristic algorithm for solving the arc routing problem with stochastic demands. *Journal of Simulation*, v. 12, 2018. ISSN 17477786.
- GRANT, D. B. *Gestão de Logística e Cadeia de Suprimentos*. [S.l.: s.n.], 2013.
- GRULER, A.; QUINTERO-ARAÚJO, C. L.; CALVET, L.; JUAN, A. A. Waste collection under uncertainty: A simheuristic based on variable neighbourhood search. *European Journal of Industrial Engineering*, v. 11, 2017. ISSN 17515262.
- GUIMARANS, D.; DOMINGUEZ, O.; PANADERO, J.; JUAN, A. A. A simheuristic approach for the two-dimensional vehicle routing problem with stochastic travel times. *Simulation Modelling Practice and Theory*, v. 89, 2018. ISSN 1569190X.

- GUTIERREZ, A.; DIEULLE, L.; LABADIE, N.; VELASCO, N. A hybrid metaheuristic algorithm for the vehicle routing problem with stochastic demands. *Computers and Operations Research*, v. 99, 2018. ISSN 03050548.
- JUAN, A. A.; FAULIN, J.; GRASMAN, S. E.; RABE, M.; FIGUEIRA, G. A review of simheuristics: Extending metaheuristics to deal with stochastic combinatorial optimization problems. *Operations Research Perspectives*, v. 2, 2015. ISSN 22147160.
- JUAN, A. A.; KELTON, W. D.; CURRIE, C. S.; FAULIN, J. Simheuristics applications: Dealing with uncertainty in logistics, transportation, and other supply chain areas. *Proceedings - Winter Simulation Conference*, v. 2018-December, 2019. ISSN 08917736.
- LI, X.; TIAN, P.; LEUNG, S. C. Vehicle routing problems with time windows and stochastic travel and service times: Models and algorithm. *International Journal of Production Economics*, v. 125, 2010. ISSN 09255273.
- LOURENÇO, H. R.; MARTIN, O. C.; STÜTZLE, T. Chapter 11 iterated local search. *Handbook of metaheuristics*, 2003.
- MAIA, P. de M. *MODELO DE OTIMIZAÇÃO E SIMULAÇÃO PARA O PROBLEMA DE ROTEAMENTO DE VEÍCULOS COM JANELAS DE TEMPO E INCERTEZA NOS TEMPOS DE VIAGEM*. [S.l.], 2021.
- MIRANDA, D. M. Metaheurísticas para as variantes do problema de roteamento de veículos: Capacitado, com janela de tempo e com tempo de viagem estocástico. *Universidade Federal de Minas Gerais*, 2011.
- MLADENOVIĆ, N.; HANSEN, P. Variable neighborhood search. *Computers and Operations Research*, v. 24, 1997. ISSN 03050548.
- OLIVEIRA, J. W. de. Metaheurísticas variable neighborhood search (vns) e variable neighborhood descent (vnd) aplicada na distribuição de combustíveis em rede multimodal. *Universidade Tecnológica Federal do Paraná*, 2020.
- OYOLA, J.; ARNTZEN, H.; WOODRUFF, D. L. The stochastic vehicle routing problem, a literature review, part i: models. *EURO Journal on Transportation and Logistics*, v. 7, 2018. ISSN 21924384.
- PEREIRA, A. M. D. M. Aplicação de meta-heurísticas ao problema de escalonamento em ambiente dinâmico de produção discreta. *Universidade do Minho*, 2003.
- RAJABALINEJAD, M. Bayesian monte carlo method. *Reliability Engineering and System Safety*, v. 95, 2010. ISSN 09518320.
- SOLOMON, M. M. Algorithms for the vehicle routing and scheduling problems with time window constraints. *Operations Research*, v. 35, 1987. ISSN 0030364X.
- VENANZI, O. S. D. Introdução à engenharia de produção: conceitos e casos práticos. Rio de Janeiro, 2016.
- ZHANG, T.; CHAOVALITWONGSE, W. A.; ZHANG, Y. Scatter search for the stochastic travel-time vehicle routing problem with simultaneous pick-ups and deliveries. *Computers and Operations Research*, v. 39, 2012. ISSN 03050548.