

UNIVERSIDADE FEDERAL DE MINAS GERAIS
PROGRAMA DE PÓS GRADUAÇÃO EM ENGENHARIA ELÉTRICA

PEDRO DE MENDONÇA MAIA

MODELO DE OTIMIZAÇÃO E SIMULAÇÃO PARA O PROBLEMA DE ROTEAMENTO
DE VEÍCULOS COM JANELAS DE TEMPO E INCERTEZA NOS TEMPOS DE
VIAGEM

BELO HORIZONTE - MG

2021

PEDRO DE MENDONÇA MAIA

MODELO DE OTIMIZAÇÃO E SIMULAÇÃO PARA O PROBLEMA DE ROTEAMENTO
DE VEÍCULOS COM JANELAS DE TEMPO E INCERTEZA NOS TEMPOS DE
VIAGEM

Dissertação apresentada ao Programa de Pós
Graduação em Engenharia Elétrica da Universi-
dade Federal de Minas Gerais, como requisito
parcial para a obtenção do título de Mestre em
Engenharia Elétrica..

Orientador: Prof Michel Bessani

BELO HORIZONTE - MG

2021

M217m

Maia, Pedro de Mendonça.

Modelo de otimização e simulação para o problema de roteamento de veículos com janelas de tempo e incerteza nos tempos de viagem [recurso eletrônico] / Pedro de Mendonça Maia. - 2021.

1 recurso online (ix ,78 f. : il., color.) : pdf.

Orientador: Michel Bessani.

Dissertação (mestrado) - Universidade Federal de Minas Gerais, Escola de Engenharia.

Apêndices: f.65-68.

Bibliografia: f.57-64.

Exigências do sistema: Adobe Acrobat Reader.

1. Engenharia Elétrica - Teses. 2. Monte Carlo, Método de. - Teses. I. Bessani, Michel. II. Universidade Federal de Minas Gerais. Escola de Engenharia. III. Título.

CDU: 621.3(043)

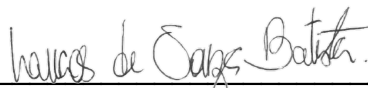
"Modelo de Otimização e Simulação Para O Problema de Roteamento de Veículos Com Janelas de Tempo e Incerteza Nos Tempos de Viagem"

Pedro de Mendonça Maia

Dissertação de Mestrado submetida à Banca Examinadora designada pelo Colegiado do Programa de Pós-Graduação em Engenharia Elétrica da Escola de Engenharia da Universidade Federal de Minas Gerais, como requisito para obtenção do grau de Mestre em Engenharia Elétrica.

Aprovada em 05 de julho de 2021.

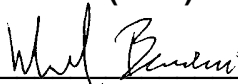
Por:



Prof. Dr. Lucas de Souza Batista
DEE (UFMG)



Prof. Dr. Alexandre Cláudio Botazzo Delbem
ICMC (USP)



Prof. Dr. Michel Bessani (Orientador)
DEE (UFMG)

AGRADECIMENTOS

Meus agradecimentos principais vão para meus pais, Remison e Claudia, por todo o incentivo e apoio para que eu pudesse me dedicar totalmente ao mestrado. Agradeço também minha irmã Maria Clara, pelo apoio. Também agradeço a minha namorada, Júlia, pelo companheirismo e por sempre me incentivar em todos os momentos.

Agradecimentos especiais ao pessoal do laboratório ORCS pelo convívio diários, boas conversas e trocas de conhecimento. Agradeço principalmente ao André Batista e André Maravilha, por me auxiliarem com os servidores e equipamentos do laboratório.

Por fim, agradeço a CAPES pelo suporte financeiro que me permitiu dedicar à essa pesquisa.

RESUMO

O Problema de Roteamento de Veículos com Janelas de Atendimento (PRV-JA) é um problema de otimização combinatória com várias aplicações em logística. Seu objetivo é estabelecer rotas para que um conjunto de veículos atenda às demandas de um conjunto de clientes durante seus períodos de disponibilidade com menor custo possível. Modelos tradicionais consideram tempos fixos de serviço e deslocamento, o que pode gerar soluções ruins ou até mesmo impraticáveis em problemas reais, onde há incerteza envolvida. Neste trabalho, primeiramente as metaheurísticas ILS+VND são usadas para resolver o PRV-JA da forma tradicional, com o único objetivo de minimizar o custo. Então, os tempos de viagem e atendimento da solução final são substituídos por funções de densidade de probabilidade e simulações de Monte Carlo são usadas para avaliar o risco de não respeitar a disponibilidade dos clientes. Experimentos mostram que o nível de influência da incerteza depende das características de cada instância. Identificado o efeito da incerteza na qualidade das soluções, um modelo de otimização multiobjetivo é proposto visando minimizar simultaneamente custo e risco. Uma abordagem que combina otimização e simulação é adotada, onde o custo é calculado de forma determinística e o risco é estimado a partir de simulações de Monte Carlo. Três variações do NSGA-II são propostas para resolver o problema e três métricas de performance são aplicadas às fronteiras Pareto estimadas para medir a qualidade das soluções encontradas por cada algoritmo. Os resultados mostraram que soluções obtidas consideram uma boa relação custo-benefício entre custo e risco.

Palavras-chaves: Problema de Roteamento de Veículos com Janelas de Atendimento Estocástico, ILS, NSGAI, Simulação de Monte Carlo.

ABSTRACT

The Vehicle Routing Problem with Time Windows (VRP-TW) is a combinatorial optimization problem with many applications in logistics areas. Its goal is to determine routes for the available vehicles to attend the demands of a set of customers during their available times with minimum cost. Traditional models use fixed service and travel times, which may lead to bad or even impractical solutions in real world-problems, where there is uncertainty. In this work, metaheuristics ILS+VND are initially used to solve the VRP-TW in the traditional way, with the only objective of minimizing cost. Then, the travel and attendance times are replaced by probability density functions and Monte Carlo simulations are used to evaluate the risk of not respecting customers' availability. Experiments show the level of uncertainty effect depends on instance characteristics. A multiobjective optimization model to minimize cost and risk simultaneously is proposed once the uncertainty effect is identified. An approach combining optimization and simulation is adopted, where the cost is calculated deterministically and Monte Carlo simulations are used to estimate the risk. Three NSGA-II variations are proposed to solve the problem and three performance metrics are applied to the estimated Pareto fronts to measure the solution quality obtained by each algorithm. The results show the solutions found have a nice trade-off between cost and risk.

Key-words: Stochastic Vehicle Routing Problem with Time Windows. ILS. NSGAI. Monte Carlo simulation.

LISTA DE ILUSTRAÇÕES

FIGURA 1 – Exemplo de duas rotas R_1 e R_2 . Figura adaptada de Tan, Cheong e Goh (2007)	1
FIGURA 2 – Exemplo de fronteira Pareto de um problema com dois objetivos. O espaço de soluções aparece em azul e a fronteira Pareto ótima em vermelho.	13
FIGURA 3 – Simulação de Monte Carlo. Figura adaptada de O'Connor e Kleyner (2011)	14
FIGURA 4 – Procedimentos do NSGA-II. Figura adaptada de K. Deb et al. (2002).	18
FIGURA 5 – Crowding Distance. Figura adaptada de K. Deb et al. (2002)	20
FIGURA 6 – Hipervolume para problema com dois objetivos.	22
FIGURA 7 – Exemplo de solução do PRV-JA com 3 veículos e 13 clientes. Os valores no interior de cada cliente representam seu tempo de atendimento e os valores entre colchetes o início e fim das janelas de atendimento.	23
FIGURA 8 – Fluxograma da avaliação do efeito da incerteza.	25
FIGURA 9 – Exemplo de solução candidata com três veículos e doze clientes.	26
FIGURA 10 – Exemplo dos operadores de busca local aplicados na configuração inicial (a)	29
FIGURA 11 – Exemplo dos operadores de perturbação aplicados na configuração inicial (a)	30
FIGURA 12 – Exemplo de cruzamento usando o BCRC. Figura adaptada de Ombuki, Ross e Hanshar (2006).	34
FIGURA 13 – Fluxograma das versões do NSGA-II. As setas contínuas ligam os blocos do algoritmo original. Os blocos tracejados são executados apenas quando o ILS é usado e os pontilhados apenas no algoritmo Memético.	36
FIGURA 14 – Diferença percentual média de distância total entre resultados encontrados e valores de referência.	39
FIGURA 15 – Diferença média de veículos usados entre resultados encontrados e valores de referência.	40
FIGURA 16 – Média de clientes violados por classe de problema.	41
FIGURA 17 – Probabilidade de falha média por classe de problema.	42
FIGURA 18 – Fronteiras Pareto estimadas dos problemas C1	43
FIGURA 19 – Fronteiras Pareto estimadas dos problemas C2	43

FIGURA 20 – Fronteiras Pareto estimadas dos problemas R1	43
FIGURA 21 – Fronteiras Pareto estimadas dos problemas R2	44
FIGURA 22 – Fronteiras Pareto estimadas dos problemas RC1	44
FIGURA 23 – Fronteiras Pareto estimadas dos problemas RC2	44
FIGURA 24 – Média de Gerações por algoritmo	45
FIGURA 25 – Média do número de soluções por algoritmo	45
FIGURA 26 – Média do crescimento de HV por algoritmo	46
FIGURA 27 – Normalidade dos resíduos para o modelo ajustado com a bloca- gem para o HV.	47
FIGURA 28 – Normalidade dos resíduos para o modelo ajustado com a bloca- gem para o Delta.	48
FIGURA 29 – Normalidade dos resíduos para o modelo ajustado com a bloca- gem para o RNI.	48
FIGURA 30 – Resíduos por cada algoritmo com blocagem para o HV.	49
FIGURA 31 – Resíduos por cada algoritmo com blocagem para o Delta.	49
FIGURA 32 – Resíduos por cada algoritmo com blocagem para o RNI.	50
FIGURA 33 – Intervalo de confiança de Tukey para o HV.	51
FIGURA 34 – Intervalo de confiança de Tukey para o Delta.	52
FIGURA 35 – Intervalo de confiança de Tukey para o RNI.	53

LISTA DE TABELAS

TABELA 1 – Resultados da blocagem para comparação de Algoritmos	47
TABELA 2 – Probabilidade de não existir diferença entre algoritmos no teste de Tukey com a blocagem	50
TABELA 3 – Resultados da otimização na análise do efeito da incerteza	66
TABELA 4 – Resultados da simulação na análise do efeito da incerteza	67
TABELA 5 – Resultados médios dos algoritmos multiobjetivo.	68

SUMÁRIO

1	INTRODUÇÃO	1
1.1	MOTIVAÇÃO E JUSTIFICATIVA	2
1.2	OBJETIVOS	3
1.3	CONTRIBUIÇÕES	4
1.3.1	Publicação Decorrente dessa Pesquisa	4
1.4	ESTRUTURA DO TRABALHO	4
2	REVISÃO BIBLIOGRÁFICA	5
3	FUNDAMENTAÇÃO TEÓRICA	12
3.1	OTIMIZAÇÃO	12
3.2	OTIMIZAÇÃO MULTIOBJETIVO	12
3.3	SIMULAÇÃO DE MONTE CARLO	13
3.4	<i>SIMHEURISTICS</i>	14
3.5	ALGORITMOS	15
3.5.1	Iterated Local Search	15
3.5.2	Variable Neighborhood Descent	16
3.5.3	Non-dominated Sorting Genetic Algorithm II	16
3.5.3.1	Classificação Não Dominada Rápida	17
3.5.3.2	Distância de Aglomeração	18
3.5.3.3	Criação da Nova População	19
3.5.4	Métricas de Qualidade	21
3.5.4.1	Ratio of Nondominated Individuals	21
3.5.4.2	Delta	21
3.5.4.3	Hipervolume	22
4	METODOLOGIA	23
4.1	MODELAGEM DO PROBLEMA DE ROTEAMENTO DE VEÍCULOS COM JANELAS DE ATENDIMENTO	23
4.2	AVALIAÇÃO DO EFEITO DA INCERTEZA NA SOLUÇÃO DO PROBLEMA DE ROTEAMENTO DE VEÍCULOS COM JANELA DE ATENDIMENTO	25
4.2.1	Representação da Solução	25
4.2.2	Operadores do ILS	27
4.2.2.1	Criação da População Inicial	27
4.2.2.2	Busca Local	28
4.2.2.3	Heurísticas de Perturbação	29

		11
4.2.3	Simulação de Monte Carlo	30
4.2.4	Base de dados	32
4.3	MODELO MULTIOBJETIVO PARA O PROBLEMA DE ROTEAMENTO DE VEÍCULOS COM JANELA DE ATENDIMENTO E TEMPOS ESTO- CÁSTICOS	33
4.3.1	Operadores NSGA II	33
4.3.1.1	Criação da população Inicial	33
4.3.1.2	Operador de Cruzamento	33
4.3.1.3	Operadores de Mutação	34
4.3.1.4	Hipervolume	34
4.3.1.5	Parâmetros do Algoritmo	35
4.3.2	Planejamento Experimental	35
4.4	COMPARAÇÃO ENTRE ALGORITMOS	36
5	RESULTADOS	38
5.1	AVALIAÇÃO DO EFEITO DE INCERTEZA	38
5.1.1	Resultados da otimização	38
5.1.2	Análise do efeito da incerteza nas soluções	40
5.2	OTIMIZAÇÃO MULTIOBJETIVO	42
5.2.1	Resultados da Otimização	43
5.2.2	Comparação entre os algoritmos	46
6	CONCLUSÃO	54
6.1	PROPOSTA DE TRABALHOS FUTUROS	55
	REFERÊNCIAS	57
	APÊNDICES	65
	APÊNDICE A TABELAS DE RESULTADOS	66

1 INTRODUÇÃO

O Problema de Roteamento de Veículos (PRV) pode ser definido como uma classe de problemas de otimização combinatória em que um conjunto de clientes distribuídos geograficamente devem ser atendidos por um conjunto de veículos. Cada veículo sai de um depósito e percorre uma rota distinta para atender um conjunto de clientes, até retornar ao depósito no final do percurso. O objetivo do problema é encontrar o conjunto de rotas de cada veículo que minimize alguma métrica definida previamente, como o custo de transporte ou o tempo total de percurso.

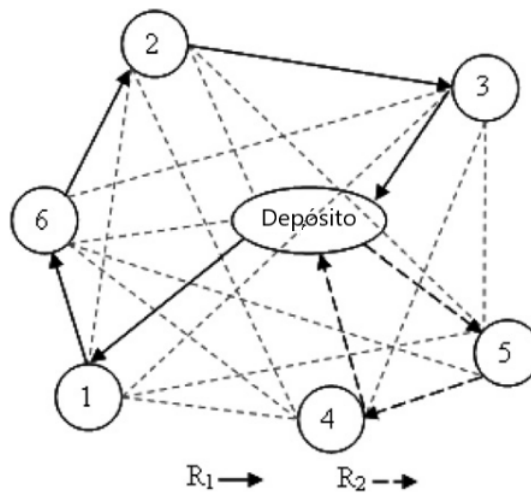


FIGURA 1 – Exemplo de duas rotas R_1 e R_2 . Figura adaptada de Tan, Cheong e Goh (2007)

A Figura 1 mostra um pequeno exemplo do problema com um depósito, seis clientes e dois veículos. O primeiro veículo executa a rota R_1 (*Depósito-1-6-2-3-Depósito*) e o segundo segue a rota R_2 (*Depósito-5-4-Depósito*). As linhas tracejadas finas mostram a quantidade de trajetórias possíveis considerando esses poucos clientes. O aumento no número de clientes faz com que o número de opções cresça rapidamente, elevando a complexidade do problema. Além disso, a quantidade de alternativas também cresce drasticamente com o aumento da disponibilidade de veículos, pois é necessário definir quais veículos atendem quais clientes (DANTZIG; RAMSER, 1959).

O PRV tem diversas aplicações práticas, entretanto dificilmente esse modelo tradicional é suficiente para suprir todas as necessidades do mundo real. Cada tipo de aplicação possui características e restrições específicas, então modelos que incorporem aspectos distintos são necessários para lidar com cada problema, como múltiplos depósitos, veículos heterogêneos, roteamento com coleta e entrega, clientes com janelas de atendimento, tempos de viagem dinâmicos, demandas estocásticas, e outros. Atualmente existe uma família de tipos de problemas de roteamento com diferentes

nomenclaturas dependendo das características que são adotadas. As restrições consideradas mais comuns são a carga horária dos motoristas, a capacidade dos veículos e janelas de atendimento dos clientes (BRAEKERS; RAMAEKERS; NIEUWENHUYSE, 2016).

Geralmente os modelos que incorporam múltiplas características são capazes de encontrar soluções melhores. Esses problemas são chamados de Problemas de Roteamento de Veículos Ricos (Rich Vehicle Routing Problem - RVRP) (LAHYANI; KHEMAKHEM; SEMET, 2015). Por outro lado, algoritmos mais complexos são necessários para resolvê-los e maior capacidade computacional é requerida.

Uma das famílias de PRV de difícil solução é a que considera incerteza em pelo menos um dos parâmetros, dando nome aos Problemas de Roteamento de Veículos Estocásticos (PRV-E) (GENDREAU; LAPORTE; SÉGUIN, 1996). Em aplicações urbanas há muita incerteza quanto aos tempos de viagem devido aos diversos fatores externos de tráfego. Aliado a isso, pode existir ainda incerteza na duração de cada atendimento. Quando há incerteza nos tempos de viagem ou atendimento o problema ganha o nome de Problemas de Roteamento de Veículos com Tempos Estocásticos (PRV-TE).

A simulação de soluções estocásticas é uma das formas de considerar incerteza no processo de otimização. A combinação de técnicas aumenta ainda mais a complexidade computacional, porém o aumento da capacidade de processamento dos computadores tem permitido esse tipo de solução.

Bons algoritmos para resolver um PRV-TE são ainda mais relevantes quando se trata também de um Problema de Roteamento de Veículos com Janelas de Atendimento (PRV-JA), isto é, quando cada cliente possui horários de atendimento específicos. Nessa situação atrasos não considerados podem inviabilizar o atendimento de alguns clientes.

Quando o PRV-JA é resolvido com objetivo de minimizar custo, algoritmos de otimização mono-objetivo tendem a encontrar soluções que explorem toda a janela de atendimento dos clientes. Entretanto, ao incluir incerteza nos tempos, atrasos tendem a ocorrer nos atendimentos no fim da janela, o que leva a soluções de alto risco. Um algoritmo multiobjetivo que considere a minimização do risco na segunda função objetivo tem a capacidade de encontrar tanto soluções de menor distância e maior risco quanto soluções de maior distância e menor risco.

1.1 MOTIVAÇÃO E JUSTIFICATIVA

O PRV é um problema com inúmeras aplicações práticas, mas por ser NP-difícil possui uma complexidade elevada e o tempo de solução em instâncias médias ou

grandes é alto (SALEHIPOUR et al., 2011). Muitas das aplicações desse problema demandam uma solução rápida, portanto é extremamente necessário que sejam desenvolvidos métodos aproximados que consigam encontrar resultados satisfatórios em pouco tempo. Além disso, modelos capazes de lidar com as incertezas do mundo real trazem ganhos na qualidade da solução (GUTIERREZ et al., 2018).

O problema de roteamento e suas variações possuem alta complexidade e diversas aplicações em logística. Nem sempre os modelos utilizados conseguem refletir bem todos os detalhes dos problemas reais. Sendo assim, uma análise do nível de impacto de incertezas de tempo se torna necessária para ajudar a definir quando a modelagem dessa característica se faz necessária. Isso é importante, pois existem aplicações em que atrasos são inviáveis ou que geram alto prejuízo, como no transporte de materiais altamente perecíveis. Mesmo em aplicações menos críticas, falhas frequentes nos atendimentos prejudicam a imagem da empresa com seus clientes, o que pode ser prejudicial em longo prazo.

Existem abordagens para otimizar o PRV-JA estocástico que consideram custo e risco, mas esses modelos são específicos para o caso do tempo com distribuição normal. A motivação para este trabalho é propor um método mais genérico, que seja independente da distribuição.

1.2 OBJETIVOS

O PRV é um problema de otimização combinatória com várias aplicações, porém é cercado por incertezas e de difícil solução. Dessa forma, o objetivo desse trabalho é construir um modelo estocástico de roteamento de veículos que seja capaz de encontrar uma solução em tempo razoável e que seja capaz de representar bem as incertezas do mundo real. Para isso, serão desenvolvidos modelos com e sem incertezas e algoritmos capazes de obter soluções satisfatórias. Com tudo isso, será possível comparar os resultados dos diferentes modelos e avaliar o efeito das incertezas na qualidade da solução.

Geralmente a utilização de um único objetivo não é capaz de abordar todos os aspectos do problema, então uma abordagem multiobjetivo será desenvolvida, o que dá maior autonomia para o tomador de decisão na escolha da solução empregada. Considerar incertezas na abordagem permite estimar a probabilidade de falha das soluções obtidas e assim incluir uma análise de risco no processo de tomada de decisão.

Um dos objetivos desse trabalho é apresentar um estudo do efeito de incertezas de tempo na qualidade da solução do PRV-JA com diferentes configurações de clientes, considerando diferentes distribuições geográficas e duração das janelas de atendimento.

Outro objetivo é estabelecer uma metodologia para solução do problema que considere as incertezas e seja capaz de obter soluções menos suscetíveis à falhas, sem que isso dependa da forma como as incertezas de tempo são consideradas. Dessa forma, os objetivos específicos deste trabalho são:

- Investigar se a organização espacial e a duração das janelas de atendimento dos clientes afetam no risco da solução do PRV-JA;
- Propor um modelo para o PRV-JA com tempos estocásticos que considere custo e risco combinando otimização e simulação;
- Propor e comparar métodos multiobjetivos para solucionar o PRV-JA com incertezas de tempo independente da distribuição de probabilidade adotada para os tempos de viagem e atendimento;
- Avaliar o efeito de uma heurística construtiva na qualidade da solução de um modelo multiobjetivo.

1.3 CONTRIBUIÇÕES

Este trabalho tem duas principais contribuições. A primeira é uma análise do efeito da incerteza nos tempos de viagem e atendimento em soluções do PRV-JA para instâncias com clientes com diferentes organizações espaciais e períodos de disponibilidade. A segunda é um método adequado para a solução do PRV-JA estocástico independente das distribuições de probabilidade adotadas.

1.3.1 Publicação Decorrente dessa Pesquisa

Pedro de Mendonça Maia; Michel Bessani. ANÁLISE DO EFEITO DE INCERTEZAS DE TEMPO APLICADAS AO PROBLEMA DE ROTEAMENTO DE VEÍCULOS COM JANELAS DE ATENDIMENTO. In: ANAIS DO LII SIMPÓSIO BRASILEIRO DE PESQUISA OPERACIONAL, 2020, João Pessoa. (MAIA; BESSANI, 2020).

1.4 ESTRUTURA DO TRABALHO

No capítulo 2 é apresentado um estudo sobre o PRV, suas diferentes versões, nomenclaturas e aplicações, com foco nas versões estocásticas e com janela de tempo. No capítulo 3 são mostrados os conceitos relacionados ao problema e os algoritmos usados para resolvê-lo são explicados. No capítulo 4 os experimentos realizados são descritos, as configurações usadas são mostradas e a base de dados utilizada para os testes é apresentada. No capítulo 5 os resultados obtidos nos testes são mostrados e discutidos. No capítulo 6 são mostradas as conclusões do trabalho.

2 REVISÃO BIBLIOGRÁFICA

O Problema de Roteamento de Veículos (PRV) (*Vehicle Routing Problem - VRP*) foi introduzido por Dantzig e Ramser (1959) com o nome de *Truck Dispatching Problem* e pode ser entendido como uma generalização do clássico Problema do Caixeiro Viajante (PCV). Desde que foi proposto, diversos autores identificaram a aplicabilidade do PRV em várias situações e assim o volume de artigos sobre problemas de roteamento cresceu continuamente desde a década de 1970, considerando diversas variantes, métodos de solução e aplicações (EKSIUGLU; VURAL; REISMAN, 2009).

Um exemplo de aplicação é apresentado por Chen, Hsueh e Chang (2009), que o aplicam na distribuição de alimentos perecíveis. Os autores consideram que os alimentos deterioram com a passagem de tempo e a função objetivo visa minimizar a perda de alimentos.

A aplicação de variações do PRV na manutenção de redes elétricas é muito comum. Um planejamento de reparos emergenciais de uma rede de distribuição após falhas de larga escala é proposto por Pang et al. (2016). Nesse contexto prejuízos sócio-econômicos são acumulados até que os reparos sejam executados, então o planejamento deve priorizar a manutenção de cargas mais relevantes. Garcia et al. (2018) realizam o planejamento da manutenção de redes de distribuição de energia *smart grid* usando uma variação do PRV. Na formulação consideram que clientes prioritários devem ser atendidos primeiro e resolvem o problema em tempo real considerando o surgimento de eventuais reparos emergenciais. Irawan et al. (2021) aplicam o PRV na manutenção de usinas eólicas *off-shore* considerando o tipo de manutenção e de profissionais necessários.

Por se tratar de um problema NP-difícil, atualmente heurísticas dominam a pesquisa em relação ao PRV. De acordo com Braekers, Ramaekers e Nieuwenhuyse (2016), menos de 20% dos artigos entre 2009 e 2016 usaram métodos exatos, sendo os algoritmos evolutivos e outras metaheurísticas os métodos mais populares nesse período.

Métodos exatos garantem a solução ótima do problema, mas demandam muito tempo de execução e por esse motivo são limitados a pequenas instâncias. Heurísticas têm se mostrado eficientes para encontrar soluções próximas do ótimo global em tempo consideravelmente inferior. Salehipour et al. (2011) utilizam o GRASP para resolver instâncias de médio e grande porte do PRV com um único veículo. Os resultados obtidos foram comparados com o modelo exato resolvido pelo software comercial de otimização CPLEX. Soluções similares ao do modelo exato foram encontradas pela

metaheurística com poucos segundos de execução, enquanto o CPLEX necessitou de até um dia para chegar na solução final.

Uma abordagem que combina o modelo exato com heurística é apresentada por Patrizia Beraldi et al. (2015). Os autores propõem um modelo exato usando branch-and-cut para resolver pequenas instâncias do PRV estocástico e uma heurística de busca em vizinhança onde o branch-and-cut é usado em uma parcela da rota para instâncias grandes. Abordagens desse tipo são bem promissoras, pois combinam a rapidez das heurísticas com a qualidade dos métodos determinísticos, e por isso podem ser mais pesquisadas no futuro.

As diversas aplicações do PRV levaram a inúmeras adaptações na modelagem necessárias para suprir diferentes aspectos da realidade que se mostraram relevantes em diferentes casos. Existem modelos que consideram veículos heterogêneos, múltiplos depósitos, janelas de atendimento, demandas fracionadas, entre outros (TOTH; VIGO, 2014; LAHYANI; KHEMAKHEM; SEMET, 2015; BRAEKERS; RAMAEKERS; NIEUWENHUYSE, 2016).

Um dos exemplos de variação de modelagem são os Problemas de Roteamento de Veículos Dinâmicos (PRV-D), onde algum parâmetro do problema varia ao longo do tempo. Malandraki e Daskin (1992) introduziram os tempos dinâmicos, em que o tempo de viagem depende do horário do dia. Esse modelo é útil em aplicações urbanas em que há horários de pico de congestionamento, e o modelo tenta criar rotas que evitem trechos nos seus piores horários.

No Problema de Roteamento de Veículos com Coleta e Entrega (PRV-CE) cada veículo pode tanto receber ou entregar volume para os clientes (ASSIS et al., 2012). Nesse problema a violação da capacidade dos veículos deve ser checada a cada parada, e não apenas no início ou final da rota como nos problemas com apenas entrega ou coleta.

Em outros casos, um nível de incerteza está associado a parâmetros como os tempos de viagem, de serviço e o volume de demanda dos clientes. Sendo assim, modelos que consideram todas as grandezas fixas muitas vezes acabam encontrando soluções que ao serem aplicadas na prática não retornam os resultados esperados (GUTIERREZ et al., 2018). Por esse motivo, foi desenvolvida a classe de Problemas de Roteamento de Veículos Estocásticos (PRV-E), em inglês *Stochastic Vehicle Routing Problem* (SVRP). Essas versões do PRV substituem parâmetros fixos por alguma distribuição de probabilidade, tornando o problema mais desafiador.

Em problemas com demanda incerta pode ser inviável o atendimento de todas as demandas devido à capacidade dos veículos. Já quando a incerteza reside nos tempos de viagem e/ou serviço ou na disponibilidade dos clientes, problemas relacionados

à carga horária dos motoristas ou à janela de tempo dos clientes podem ocorrer.

Gendreau, Laporte e Séguin (1996) e Gendreau, Jabali e Rei (2016) apresentam a evolução das abordagens em torno do PRV-E e suas variantes. 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.

Oyola, Woodruff e Arntzen (2016) apresentam uma revisão da literatura sobre o PRV-E desde 1997. De acordo com os autores, o mais comum é considerar demandas incertas. Dentre os trabalhos avaliados pelos autores, praticamente 50% trabalham dessa forma.

Nos problemas estocásticos falhas podem ocorrer quando uma rota excede alguma restrição. Quando isso acontece, duas abordagens são usadas para contornar o problema. A primeira é o *chance constrained program* (CCP), onde garante-se que a probabilidade de uma rota falhar fica abaixo de certo nível pré-determinado (BERALDI, Patrizia et al., 2015; GÓMEZ et al., 2016; MIRANDA, Douglas Moura; CONCEIÇÃO, Samuel Vieira, 2016; ZHANG; LAM, W. H.; CHEN, 2016). A outra abordagem é um pouco mais complexa, chamada de *stochastic program with recourse* (SPR), onde alguma medida deve ser tomada quando uma falha ocorre (TAN; CHEONG; GOH, 2007; GAUVIN; DESAULNIERS; GENDREAU, 2014; GOODSON, 2015; ZHANG; LAM, W. H.; CHEN, 2016; GUTIERREZ et al., 2018).

Quando o SPR é adotado, três métodos são usados para reparar falhas em problemas com demandas estocásticas, segundo Tan, Cheong e Goh (2007):

1. No ponto de falha, retornar ao depósito para reabastecimento e posteriormente retomar a rota no mesmo ponto;
2. Reotimizar as rotas para os clientes não atendidos após cada falha ou cada atendimento;
3. Permitir retornos ao depósito para reabastecimento em qualquer momento anterior à falha.

O primeiro método é o mais usado, sendo adotado em aproximadamente 50% dos artigos (OYOLA; WOODRUFF; ARNTZEN, 2016). Sua implementação é a mais simples, porém a utilização em problemas reais é limitada.

Métodos exatos e heurísticas foram utilizados na solução do PRV-E. Oyola, Arntzen e Woodruff (2016) revisam os métodos usados na literatura desde 1997. Dentre os artigos analisados, a proporção entre métodos exatos e aproximados foi semelhante aos métodos de solução do PRV em geral, com cerca de 25% usando alguma abordagem exata, sendo o algoritmo L-shaped e variações do branch-and-cut e branch-and-price os mais comuns (GENDREAU; LAPORTE; SÉGUIN, 1996). Esses métodos foram capazes de resolver instâncias de até aproximadamente 100 nós com um veículo ou 60 ou 80 nós considerando mais veículos (GAUVIN; DESAULNIERS; GENDREAU, 2014; TAŞ et al., 2014). Já considerando as metaheurísticas, a maioria dos trabalhos utilizaram algoritmos de busca local, sendo a Busca Tabu o mais comum. *Iterated Local Search* (ILS), *Greedy Randomized Adaptive Search Procedure* (GRASP), *Simulated Annealing* (SA), Ant Colony Optimization (ACO), algoritmos genéticos (Genetic Algorithms - GA) e programação dinâmica (Dynamic Programming - DP) também foram empregados (TAN; CHEONG; GOH, 2007; GOODSON, 2015; MENDOZA; ROUSSEAU; VILLEGAS, 2015; MIRANDA, Douglas Moura; CONCEIÇÃO, Samuel Vieira, 2016; ZHANG; LAM, W. H. K.; CHEN, 2013; GUTIERREZ et al., 2018).

O Problema de Roteamento de Veículos com Tempos Estocásticos (PRV-TE) foi introduzido por Laporte, Louveaux e Mercure (1992). Os tempos de viagem e/ou atendimento determinísticos são substituídos por valores incertos seguindo Funções de Densidade de Probabilidade (FDPs) pré definidas, o que adiciona a probabilidade de uma solução não respeitar as restrições temporais. Segundo o levantamento feito por Braekers, Ramaekers e Nieuwenhuyse (2016) com artigos sobre PRV entre 2009 e 2015, as janelas de atendimento estão presentes em 78% dos PRV-TE.

Douglas M. Miranda e Samuel V. Conceição (2010) propõem considerar o atendimento de um nível de serviço mínimo. Os autores aplicam uma penalização sempre que um cliente não é atendido e usam um modelo de Programação Linear Inteira Mista (Mixed Integer Linear Programming - MILP) que é resolvido usando o *branch-and-cut*. Eles atribuem uma distribuição *t-Student* para cada tempo de viagem e em cada trecho da rota calculam a probabilidade de o tempo de viagem ser superior ao tempo disponível. Caso essa probabilidade seja superior ao nível de serviço pré-estabelecido em qualquer trecho, a rota é considerada infactível. O desempenho do modelo considerando tempos determinísticos é comparado com o estocástico considerando os níveis de serviço e os resultados mostraram que o número de multas geradas foi menor ao considerar a incerteza.

Lei, Laporte e Guo (2011) trabalham com o PRV com tempos de serviço

estocásticos. Nesse caso, os tempos de viagem são pequenos se comparados com os de serviço, o que torna a variabilidade do tempo de viagem irrelevante em relação à variabilidade do tempo de serviço. Os autores utilizam as metaheurísticas VNS, VND e GVNS para resolver o problema. O algoritmo que encontrou melhores resultados foi o GVNS. Os resultados obtidos pelos algoritmos estocásticos foram superiores ao do modelo determinístico seguido da avaliação do valor esperado.

Gómez et al. (2016) trabalham com o PRV com tempo de viagem e de serviço estocásticos. Para lidar com a parte estocástica eles apresentam um modelo com distribuição Phase-Type (PH), realizando uma convolução das incertezas. Os autores utilizam heurísticas para resolver o problema e fazem uma simulação do modelo PH para cada avaliação de função objetivo. Este trabalho mostra como a qualidade da solução final depende da distribuição de probabilidade das incertezas.

Douglas Moura Miranda e Samuel Vieira Conceição (2016) desenvolvem um modelo estatístico para lidar com as incertezas e mostram como a chegada prematura em um cliente tem uma influência significativa na distribuição de probabilidade da rota. Um algoritmo ILS é aplicado para resolver o modelo e bons resultados são alcançados.

P. Beraldi et al. (2018) trabalham com uma versão do PRV com tempos de viagem estocásticos onde cada atendimento gera uma receita e despesas são acumuladas com o tempo para cada cliente ainda não visitado. Os autores propõem uma heurística *beam search* para solucionar o problema.

Outra abordagem para resolver o PRV-E consiste em combinar métodos de otimização e simulação, criando as *sim-heuristics* (JUAN, A. A. et al., 2015). Essa técnica permite avaliar a incerteza associada a cada solução. Guimarans et al. (2018) usam um algoritmo baseado no ILS para resolver o PRV-TE, 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. Os resultados encontrados foram ligeiramente piores que na versão determinística do PRV, porém ao considerar o problema estocástico os resultados encontrados foram melhores que usando outras abordagens.

Anderluh et al. (2019) combinam otimização e simulação em um PRV com dois tipos de veículos. Os clientes localizados em determinadas regiões são atendidos por vans e os demais por bicicletas. As vans saem do depósito carregadas e abastecem as bicicletas em determinados postos para que estas possam atender seus clientes. Uma boa solução para esse problema requer que bicicletas e vans cheguem aos respectivos postos em horários próximos para evitar que o atraso de um veículo

implique na ociosidade de outros. Nesse cenário, incertezas nos tempos de viagem podem ter alto impacto na qualidade da solução e por isso os autores propõem um modelo que considera otimização e simulação. Um algoritmo GRASP é usado para resolver o problema determinístico, seguido de uma simulação de Monte Carlo e por fim reotimizações são executadas considerando informações da simulação. Os resultados mostraram que a qualidade das soluções melhorou ao considerar as incertezas.

A. Juan et al. (2011) trabalham com o PRV com demandas estocásticas considerando veículos com baixa capacidade, de forma que a probabilidade de uma falha ocorrer seja elevada. Metaheurísticas são usadas para resolver o problema e Simulações de Monte Carlo são feitas para calcular a confiabilidade de cada solução encontrada.

Em alguns casos existe o interesse em otimizar mais de um aspecto do PRV, como distância total e número de veículos. Uma das alternativas é ponderar os fatores e criar uma única função objetivo, como Baradaran, Shafaei e Hosseinian (2019), porém quando os objetivos são conflitantes um modelo multiobjetivo é mais adequado. Ao considerar que o problema tem mais de um objetivo, um conjunto de soluções é obtido, cada uma dando mais ou menos importância para cada aspecto do problema. Esse tipo de abordagem aumenta o poder de escolha do tomador de decisão, o que tende a levar a melhores resultados.

Assis et al. (2012) usam o MOILS e NSGA-II para resolver o PRV com entregas fixas e coletas opcionais considerando dois objetivos. O primeiro consiste na minimização do custo total e o segundo no volume não coletado. Nesse caso conflito entre os objetivos acontece pois são necessários mais veículos ou rotas mais longas para aumentar o volume coletado, mas isso faz com que o custo da solução cresça. Os algoritmos multiobjetivo são capazes de encontrar tanto soluções que tenham menor custo e menor coleta, quanto soluções com maior custo e maior volume coletado.

Miranda, Branke e Conceição (2018) modelam o PRV-TE considerando um limitante inferior para os tempos de viagem e serviço. Propõem uma abordagem multiobjetivo para solucionar o problema, visando minimizar o tempo total de viagem e maximizar o nível de serviço. Novamente os objetivos são conflitantes, porque a quantidade de veículos necessários tende a crescer para aumentar o nível de serviço, o que implica no aumento do tempo total de viagem. Versões multiobjetivo de um *Memetic Algorithm* (MA) e ILS são usados, além de um *Multi-Objective Evolutionary Algorithm* (MOEA).

Outros exemplos de modelo multiobjetivo para o PRV foram propostos por Tan, Cheong e Goh (2007) e Gee et al. (2015). Ambos consideram o PRV com demandas estocásticas. O primeiro visa minimizar o tempo de viagem, o custo dos motoristas e a quantidade de veículos. Um algoritmo evolucionário multiobjetivo foi utilizado para

resolver o problema, sendo que mecanismos de busca local foram usados para melhorar a qualidade das soluções obtidas. O último propõe um algoritmo evolucionário MOEA/D para resolver o problema. São testadas versões entre dois e quatro objetivos, sendo eles a minimização da distância total percorrida, do custo dos motoristas, do número de veículos e da diferença entre o custo dos motoristas.

Como pôde ser visto, o PRV é um problema de otimização tradicional, proposto há aproximadamente 60 anos. Desde então, muitos estudos já foram realizados resultando em diversas formulações e aplicações. Ainda assim, atualmente esse problema ainda continua desafiador e mais aplicações continuam sendo encontradas, cada vez mais complexas e que requerem representações próximas da realidade.

3 FUNDAMENTAÇÃO TEÓRICA

3.1 OTIMIZAÇÃO

Um processo de otimização consiste em encontrar o valor ideal de um conjunto de variáveis de decisão que minimize ou maximize uma determinada função objetivo. Em outras palavras, a otimização é o processo que usa métodos computacionais para encontrar a melhor combinação de valores para as variáveis do problema considerando seus objetivos e restrições (SUN; YUAN, 2006).

Um problema de otimização pode ser generalizado pela Equação (3.1), onde a função objetivo de minimização é calculada por $f(x)$ e as restrições por $g(x)$.

$$\begin{array}{ll} \min & f(x) \\ \text{sujeito a} & g(x) \leq 0 \end{array} \quad (3.1)$$

A solução do problema é dada pela combinação de valores das n variáveis de decisão $(x_1^*, x_2^*, \dots, x_n^*)$ que compõem x^* tal que $f(x^*)$ seja o menor possível respeitando todas as restrições $g(x^*)$.

Existem diversos algoritmos de otimização que garantem que a solução encontrada é a melhor possível. Entretanto, alguns problemas são muito complexos e obter a solução ótima requer tempo computacional elevado. Nesses casos é comum o uso de heurísticas ou metaheurísticas, que não garantem que a solução obtida é ótima, mas são capazes de encontrar soluções boas em tempo computacional baixo.

3.2 OTIMIZAÇÃO MULTIOBJETIVO

Um problema de otimização multiobjetivo pode ser generalizado pela Equação 3.2, onde as n funções objetivo são calculadas por $f_i(x), i \in \{1, n\}$ e as restrições por $g(x)$.

$$\begin{array}{ll} \min & f(x) \\ \text{sujeito a} & g(x) \leq 0 \end{array} \quad (3.2)$$

Essa abordagem é usada para problemas com duas ou mais funções objetivo conflitantes. Quando a melhora em um dos objetivos implica na piora de pelo menos um dos outros, uma relação de incomparabilidade na qualidade de soluções candidatas é criada. Dessa forma, a solução de um problema de otimização multiobjetivo é uma fronteira de soluções, chamada de fronteira Pareto-ótima, conforme Figura 2. Algoritmos

de otimização multiobjetivo geralmente retornam um conjunto de soluções, chamada fronteira Pareto estimada.

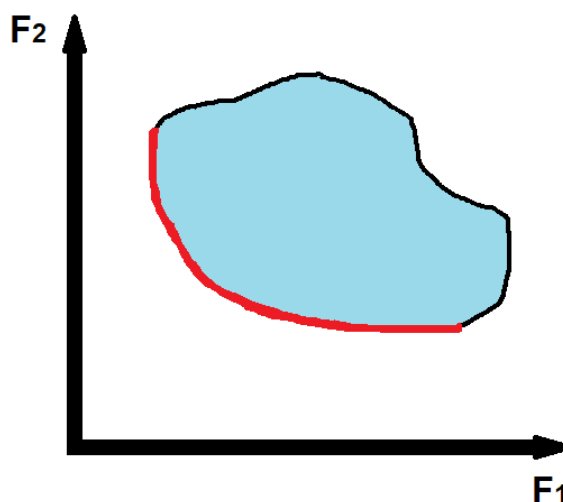


FIGURA 2 – Exemplo de fronteira Pareto de um problema com dois objetivos. O espaço de soluções aparece em azul e a fronteira Pareto ótima em vermelho.

A definição da fronteira Pareto ótima é feita a partir do conceito de Dominância (COLLETTE; SIARRY, 2004). Uma solução a domina outra solução b ($a \prec b$) caso duas condições sejam satisfeitas:

1. Nenhum dos objetivos de a é pior que b , ou seja, $f_i(a) \leq f_i(b) \quad \forall i$;
2. Pelo menos um dos objetivos de a é melhor que b , ou seja, existe algum objetivo i tal que $f_i(a) < f_i(b)$.

Se a solução a domina b pode-se dizer que a é melhor que b . Caso nenhuma das soluções sejam dominadas, diz-se que elas são incomparáveis ou não-dominadas entre si.

3.3 SIMULAÇÃO DE MONTE CARLO

A simulação de Monte Carlo é uma ferramenta utilizada para modelar o comportamento de sistemas sujeitos a incertezas nos parâmetros de entrada. O conceito básico do método é usado desde o século XVIII e atualmente é aplicado a diversas áreas, como cálculos matemáticos complexos, simulação de processos estocásticos e estimativa de confiabilidade de sistemas (BILLINTON; LI, 1994).

O processo é simples conceitualmente e computacionalmente e permite uma fácil análise do efeito das premissas consideradas. Ele é composto de duas etapas,

inicialmente é realizada uma amostragem intensiva de variáveis aleatórias de entrada x e depois observa-se o comportamento dos parâmetros de interesse do sistema y em função das entradas (HARRISON; GRANJA; LEROY, 2010), conforme Figura 3.

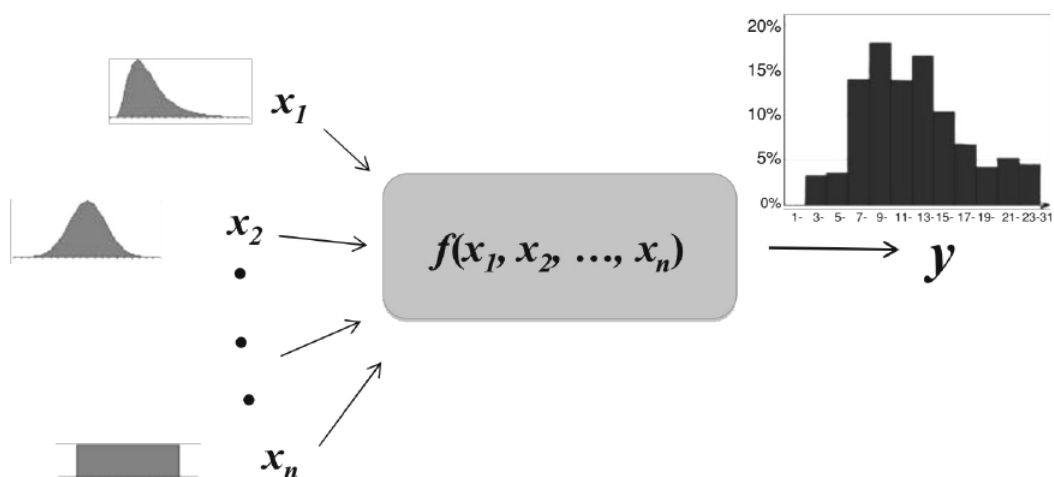


FIGURA 3 – Simulação de Monte Carlo. Figura adaptada de O'Connor e Kleyner (2011)

O método geral é apresentado no Algoritmo 1 e requer a definição do modelo e das distribuições de probabilidade das entradas. Então, o processo iterativo composto pela amostragem da entrada e observação da saída é feito.

Algoritmo 1: Abordagem Geral para Simulação de Monte Carlo

- 1 Definição do modelo paramétrico $y = f(x_1, x_2, \dots, x_n)$;
 - 2 Definição das distribuições de entrada;
 - 3 **para** $i = 1$ até N **faça**
 - 4 Gerar um conjunto aleatório de entrada $x_{i,1}, x_{i,2}, \dots, x_{i,n}$;
 - 5 $y_i = f(x_{i,1}, x_{i,2}, \dots, x_{i,n})$;
 - 6 Análise dos resultados obtidos;
-

3.4 SIMHEURISTICS

Simheuristics são técnicas que combinam heurísticas de otimização com simulação. Elas compõem parte das técnicas que combinam simulação e otimização (*Simulation Optimization* - SO). O uso desses métodos têm crescido nos últimos anos com avanços na modelagem e capacidade computacional (AMARAN et al., 2015).

A combinação de heurísticas de otimização com simulação permite a utilização de mais informações específicas do problema para aprimorar o processo de busca e alcançar resultados melhores (JUAN, A. A. 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.

3.5 ALGORITMOS

3.5.1 Iterated Local Search

O *Iterated Local Search* (ILS) é um algoritmo simples baseado em buscas locais guiadas e perturbações capaz de obter bons resultados em problemas de otimização combinatória (GENDREAU; POTVIN, 2019). Seu funcionamento é apresentado no Algoritmo 2. Uma heurística construtiva é usada para gerar uma boa solução inicial (linha 1). Então, um processo iterativo é realizado em que são aplicadas consecutivas buscas locais e perturbações até atingir um critério de parada (linhas 4 e 5). Por fim, um critério de aceitação é usado para definir se a solução obtida na iteração deve se tornar a solução corrente (linha 6).

Algoritmo 2: Iterated Local Search (ILS)

```

1  $s_0 = \text{CriaSoluçãoInicial}();$ 
2  $s^* = \text{BuscaLocal}(s_0);$ 
3 enquanto critério de parada não atingido faça
4    $s' = \text{Perturbação}(s^*, \text{Historico});$ 
5    $s'^* = \text{BuscaLocal}(s');$ 
6    $s^* = \text{Critério de Aceitação}(s^*, s'^*, \text{Historico});$ 

```

Na etapa de busca local uma ou mais estruturas de vizinhança são usadas para explorar soluções próximas à atual com o objetivo de encontrar soluções melhores. Esse processo é repetido até encontrar um mínimo local. Esse processo é fundamental na otimização, porém existe a possibilidade do algoritmo estagnar em uma bacia de atração. Então, a perturbação é usada para tentar transferir a busca para outra região, permitindo que outros pontos mínimos sejam encontrados. É importante que a perturbação seja forte o suficiente para transferir a busca para outra região, mas não pode ser excessivamente forte, pois existe o risco de ocorrer uma reinicialização aleatória do método. O critério de aceitação é responsável por determinar quando uma solução perturbada deve ser aceita ou não.

3.5.2 Variable Neighborhood Descent

A metaheurística *Variable Neighborhood Descent* (VND) é um algoritmo que estabelece uma forma determinística de trocar entre estruturas de vizinhança (GEN-DREAU; POTVIN, 2019), como mostra o Algoritmo 3.

O VND testa combinações da primeira estrutura em busca de uma solução com menor custo. Enquanto houver melhora, essa estrutura continua sendo usada. Quando há estagnação, o algoritmo passa a buscar melhora na segunda estrutura de vizinhança. As combinações são testadas novamente e, se houver melhora, a solução de menor custo é adotada e o algoritmo volta a otimizar usando a estrutura inicial. O processo é repetido até que o algoritmo passe por todas as estruturas de vizinhança sem encontrar nenhuma melhora.

Existem algumas estratégias de busca em cada estrutura, sendo as mais comuns a *first improvement*, em que uma nova solução candidata é adotada sempre que há avanço, e a *best improvement*, em que a solução candidata com maior avanço é adotada após testar todas as possibilidades. Estratégias intermediárias também podem ser usadas.

Algoritmo 3: *Variable Neighborhood Descent* (VND)

Entrada: Solução s , Conjunto de k Estruturas de Vizinhança EV
Saida: Mínimo local s^*

```

1  $i = 0$ ;
2 enquanto  $i < k$  faça
3    $s^* = EV_i(s)$ ; // Busca local com i-ésima estrutura
4   se  $f(s^*) < f(s)$  então
5      $s = s^*$ ;
6      $i = 0$ ;
7   senão
8      $i = i + 1$ ;

```

3.5.3 Non-dominated Sorting Genetic Algorithm II

O *Non-dominated Sorting Genetic Algorithm II* (NSGA II), proposto por K. Deb et al. (2002), é um algoritmo evolutivo desenvolvido para resolver problemas multiobjetivo. Esse algoritmo usa o conceito de dominância Pareto para ranquear as melhores soluções candidatas. O funcionamento do algoritmo é apresentado no Algoritmo 4.

O algoritmo mantém a cada geração dois conjuntos de população. A população P_t contém os indivíduos sobreviventes da geração anterior e Q_t armazena os indivíduos criados na última geração. Um método de criação de indivíduos é usado para inicializar P e o conjunto Q começa vazio (linhas 1 e 2). Após a inicialização das duas populações

Algoritmo 4: Non-dominated Sorting Genetic Algorithm II (NSGA II)

```

1  $P_0 = \text{CriaPopula\c{c}\~{a}oInicial}();$ 
2  $Q_0 = \emptyset;$ 
3 enquanto critério de parada não atingido faça
4    $R_t = P_t \cup Q_t;$ 
5    $F = \text{Classifica\c{c}\~{a}oN\~{a}oDominadaR\~{a}pida}(R_t);$ 
6    $P_{t+1} = \emptyset;$ 
7    $i = 1;$ 
8   enquanto  $|P_{t+1}| + |F_i| \leq N$  faça
9      $\text{CalculaDist\~{a}nciaAglomerac\~{a}o}(F_i);$ 
10     $P_t = P_{t+1} \cup F_i;$ 
11     $i = i + 1;$ 
12   $\text{OrdenaDist\~{a}nciaAglomerac\~{a}o}(F_i);$ 
13   $P_t = P_{t+1} \cup F_i[1 : (N - |P_{t+1}|)];$ 
14   $Q_{t+1} = \text{CriaNovaPopulacao}(P_{t+1});$ 
15   $t = t + 1;$ 

```

o algoritmo realiza o *loop* principal até que o critério de parada seja atingido (linhas 3 a 14).

O processo iterativo do algoritmo começa unindo as populações P_t e Q_t para formar R_t (linha 4). O método *Classificação Não Dominada Rápida (Fast Non-Dominated Sort)* faz o ranqueamento dos indivíduos em grupos (*fronts*) de acordo com o critério de dominância (linha 5). Os indivíduos no primeiro *front* não são dominados por nenhuma outra solução. Os do segundo são dominados apenas pelo do primeiro, e assim em diante.

Após a definição dos *fronts* a seleção dos sobreviventes é feita. A população P_{t+1} é inicializada sem indivíduos e então os *fronts* são adicionados até que a adição de mais um *front* em P_{t+1} exceda N indivíduos (linhas 6 a 10), conforme Figura 4. A cada etapa de adição é calculado a distância de aglomeração (*Crowding Distance*) para cada indivíduo do *front* ativo para medir a diversidade de cada solução (linha 8). Por fim a população P_{t+1} é completada com os indivíduos de maior *Crowding Distance* do melhor *front* não adicionado (linhas 11 e 12).

A etapa final do algoritmo consiste em criar a população Q_{t+1} usando um método que faz a seleção de pais, cruzamento e mutação dentre os indivíduos de P_{t+1} (linha 13). Os métodos usados no algoritmo são descritos nas seções seguintes.

3.5.3.1 Classificação Não Dominada Rápida

O algoritmo *Classificação Não Dominada Rápida (Fast Non-Dominated Sort)* é responsável por definir os *fronts* da população do algoritmo. Os indivíduos que não são dominados fazem parte do primeiro *front*, os que são dominados apenas pelos

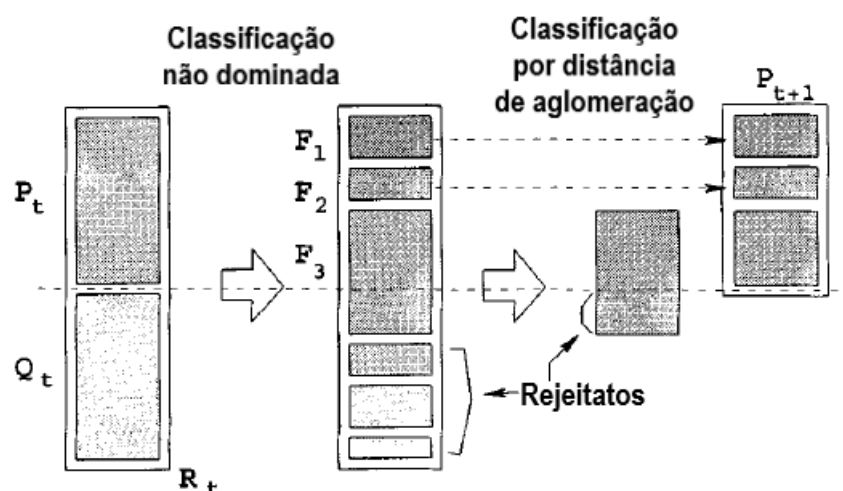


FIGURA 4 – Procedimentos do NSGA-II. Figura adaptada de K. Deb et al. (2002).

indivíduos do primeiro *front* compõem o segundo, e assim em diante. Isso é feito ranqueando cada indivíduo de acordo com o número de indivíduos dominados por ele e que o dominam. O funcionamento geral do método é apresentado do Algoritmo 5.

Inicialmente todos os pares de indivíduos são comparados (linhas 1 a 8) e são armazenados o conjunto de indivíduos dominados por cada solução p (linhas 5 e 6) e o contador n_p com a quantidade de soluções que dominam p (linhas 7 e 8). Os indivíduos que não são dominados são colocados no primeiro *front* F_1 (linhas 9 a 11). Então, o *front* dos demais indivíduos é calculado (linhas 12 a 22). Para cada solução p em F_1 , subtrai-se 1 do contador de soluções que dominam cada solução q dominada por p (linha 17). Após essas subtrações, as soluções que são dominadas apenas por soluções do primeiro *front* ficam com o contador n_q zerado, então podem ser colocadas no segundo *front* (linhas 18 a 22). O processo é repetido até que todos os indivíduos sejam atribuídos a algum *front*.

3.5.3.2 Distância de Aglomeração

O algoritmo *Classificação por Distância de Aglomeração (Crowding Distance)* tem o objetivo de medir a diversidade de soluções dentro de cada *front*. Um valor é atribuído a cada solução da fronteira correspondente a aproximação do maior cuboide que contenha a solução sem incluir nenhuma outra, como mostra a Figura 5 para um problema de dois objetivos.

O valor do *crowding distance* é calculado conforme o Algoritmo 6. Todas as soluções da fronteira são inicializadas com o valor zero (linha 2). Então, o cálculo do perímetro de cada cuboide para cada solução é feito a partir da soma dos lados de cada objetivo m (linhas 3-10). Primeiramente as soluções são ordenadas de acordo

Algoritmo 5: Classificação Não Dominada Rápida (Fast Non-Dominated Sort)

Entrada: P
Saida: F

```

1 para cada  $p \in P$  faça
2    $S_p = \emptyset$ ;
3    $n_p = 0$ ;
4   para cada  $q \in P$  faça
5     se  $p \prec q$  então
6        $S_p = S_p \cup q$ ;
7     senão se  $q \prec p$  então
8        $n_p = n_p + 1$ ;
9   se  $n_p == 0$  então
10     $p_{rank} = 1$ ;
11     $F_1 = F_1 \cup p$ ;
12  $i = 1$ ;
13 enquanto  $F_i \neq \emptyset$  faça
14    $Q = \emptyset$ ;
15   para cada  $p \in F_i$  faça
16     para cada  $q \in S_p$  faça
17        $n_q = n_q - 1$ ;
18       se  $n_q == 0$  então
19          $q_{rank} = i + 1$ ;
20          $Q = Q \cup q$ ;
21    $i = i + 1$ ;
22    $F_i = Q$ 
23

```

com a função objetivo m (linha 4) e os valores máximo e mínimo são armazenados (linhas 5 e 6). As soluções extremas assumem *crowding distance* sempre infinito para sempre serem escolhidas pelo NSGA-II (linhas 7 e 8). Isso é feito para que o melhor valor conhecido de cada objetivo tenha maior probabilidade de sobrevivência durante a execução do algoritmo. Por fim, o tamanho do lado do cuboide relativo ao objetivo m é calculado para as demais soluções do *front* (linha 10). Esse valor é ponderado pelos valores máximo e mínimo do objetivo para compensar o efeito de funções objetivo com diferentes ordens de grandeza.

3.5.3.3 Criação da Nova População

A criação dos indivíduos da geração seguinte Q_{t+1} ocorre em três etapas. Inicialmente um par de pais da população P_{t+1} é escolhido usando o método de seleção. Um processo de cruzamento é realizado entre esse par de pais e um par de filhos é gerado combinando características dos pais. Por fim, cada filho tem uma

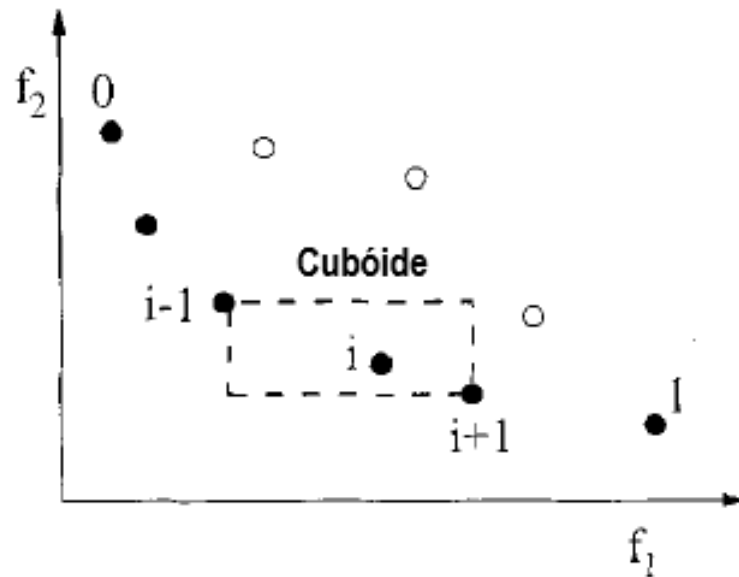


FIGURA 5 – Crowding Distance. Figura adaptada de K. Deb et al. (2002)

Algoritmo 6: Calcula Crowding Distance

Entrada: F_i
Saida: CD

- 1 $l = |F_i|;$
- 2 $CD[k] = 0 \quad \forall k \in [1, \dots, l];$
- 3 **para cada função objetivo m faça**
- 4 $F_i = \text{Ordena}(F_i, m);$
- 5 $f_{max} = \max(F_i.m);$
- 6 $f_{min} = \min(F_i.m);$
- 7 $CD[1] = \infty;$
- 8 $CD[l] = \infty;$
- 9 **para $k = 2$ até $l - 1$ faça**
- 10 $CD[k] = CD[k] + (F_i.m[k + 1] - F_i.m[k - 1]) / (f_{max} - f_{min});$

probabilidade de sofrer mutação. Esse processo é repetido até que N filhos sejam gerados.

A seleção dos pais para cruzamento é feita por meio de um torneio binário. Um par de indivíduos é escolhido aleatoriamente da população P_{t+1} . Então, o de melhor ranking é selecionado para o cruzamento. Caso ambos tenham o mesmo ranking, aquele com o maior *crowding distance* é escolhido com o intuito de aumentar a diversidade da população. Persistindo o empate, um indivíduo aleatório é escolhido.

3.5.4 Métricas de Qualidade

Métricas de qualidade são usadas para avaliar a performance de algoritmos multiobjetivo (JIANG et al., 2014). De acordo com Riquelme, Lucken e Baran (2015), há três fatores de interesse que podem ser avaliados na medição da qualidade de uma fronteira Pareto estimada:

- *Cardinalidade*: mede a quantidade de soluções na fronteira;
- *Convergência*: mede a evolução da fronteira Pareto estimada e a sua proximidade em relação à fronteira ótima teórica. Quanto essa fronteira é desconhecida um ponto ou conjunto de referência é usado;
- *Diversidade*: mede a distribuição de soluções no espaço de objetivos, ou seja, avalia se as soluções encontradas se concentram em uma única região.

Existem métricas de qualidade que avaliam um dos fatores isoladamente e outras que os consideram uma combinação.

3.5.4.1 Ratio of Nondominated Individuals

O Ratio of Nondominated Individuals (RNI) é uma das métricas mais simples de cardinalidade. Ele foi proposto por Tan, Lee e Khor (2002) e calcula a proporção de indivíduos da população que estão na fronteira Pareto ótima, ou seja, qual fração da população de tamanho N é composta com indivíduos não dominados conforme Equação 3.3.

$$RNI = \frac{|F_1|}{N} \quad (3.3)$$

onde $|X|$ é o conjunto de soluções não dominadas.

3.5.4.2 Delta

O medidor Delta (Δ), proposto por Kalyanmoy Deb et al. (2000), é uma métrica de diversidade que mede a distância Euclidiana entre duas soluções adjacentes na fronteira Pareto estimada em relação à distância média. A Equação 3.4 mostra como calculá-la.

$$\Delta = \sum_{i=1}^{F_1} \frac{|d_i - \bar{d}|}{F_1} \quad (3.4)$$

3.5.4.3 Hipervolume

O Hipervolume é uma das métricas mais usadas na avaliação de algoritmos multiobjetivo. Ele foi proposto por Zitzler e Thiele (1998) e avalia convergência e diversidade simultaneamente.

O método pode considerar um limitante superior e um inferior, conforme Figura 6. Caso a fronteira ótima teórica A^* seja conhecida, ela é usada como limitante superior e caso contrário é usado um ponto utópico melhor que qualquer solução possível em todos os objetivos. O limitante inferior é chamado *nadir* e equivale a um ponto pior que qualquer solução em todos os objetivos.

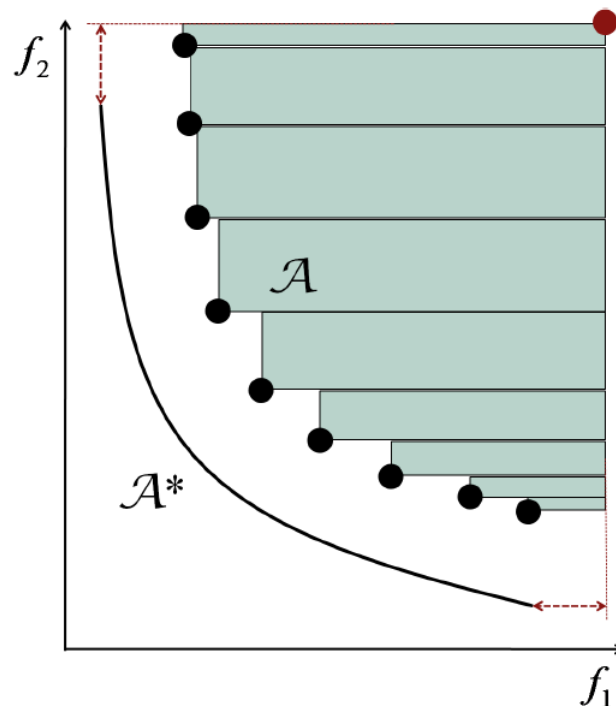


FIGURA 6 – Hipervolume para problema com dois objetivos.

O valor do hipervolume é dado pela área total A apresentada na Figura 6, que considera a soma de volumes S_i entre duas soluções adjacentes da fronteira Pareto estimada F_1 e o ponto *nadir*. O valor normalizado também pode ser usado ao dividi-lo pelo volume total entre A^* ou o ponto utópico e *nadir*, conforme Equação 3.5.

$$HV(F_1, A^*) = \frac{\cup vol(S_i)}{vol(A^*)} \quad (3.5)$$

Dessa forma, o valor da métrica se aproxima de 1 quanto mais espalhados e próximos os valores encontrados estiverem da fronteira teórica.

4 METODOLOGIA

4.1 MODELAGEM DO PROBLEMA DE ROTEAMENTO DE VEÍCULOS COM JANELAS DE ATENDIMENTO

O PRV-JA consiste em determinar um conjunto de rotas, com menor custo, para que uma frota de veículos atenda um conjunto de clientes respeitando suas janelas de disponibilidade. Cada rota começa e termina no depósito e cada cliente é visitado exatamente uma vez por um único veículo. Esse problema pode ser modelado como um grafo $G = (V, A)$, onde $V = \{v_0, \dots, v_n\}$ é o conjunto de vértices e $A = \{i, j\} : v_i, v_j \in V$ é o conjunto de arestas. Cada aresta tem um custo associado $c_{i,j}$ correspondente ao custo de viagem de v_i para v_j equivalente à distância entre os pontos. O problema contém um depósito, representado por v_0 , onde a frota se encontra inicialmente. Os demais vértices contêm os clientes, representados por $N = V \setminus v_0$, que possuem uma demanda d_i , um tempo de serviço s_i e uma janela para atendimento $[a_i, b_i]$. Cada veículo $k \in K$ possui uma capacidade Q_k . Um exemplo de solução do PRV-JA pode ser visto na Figura 7.

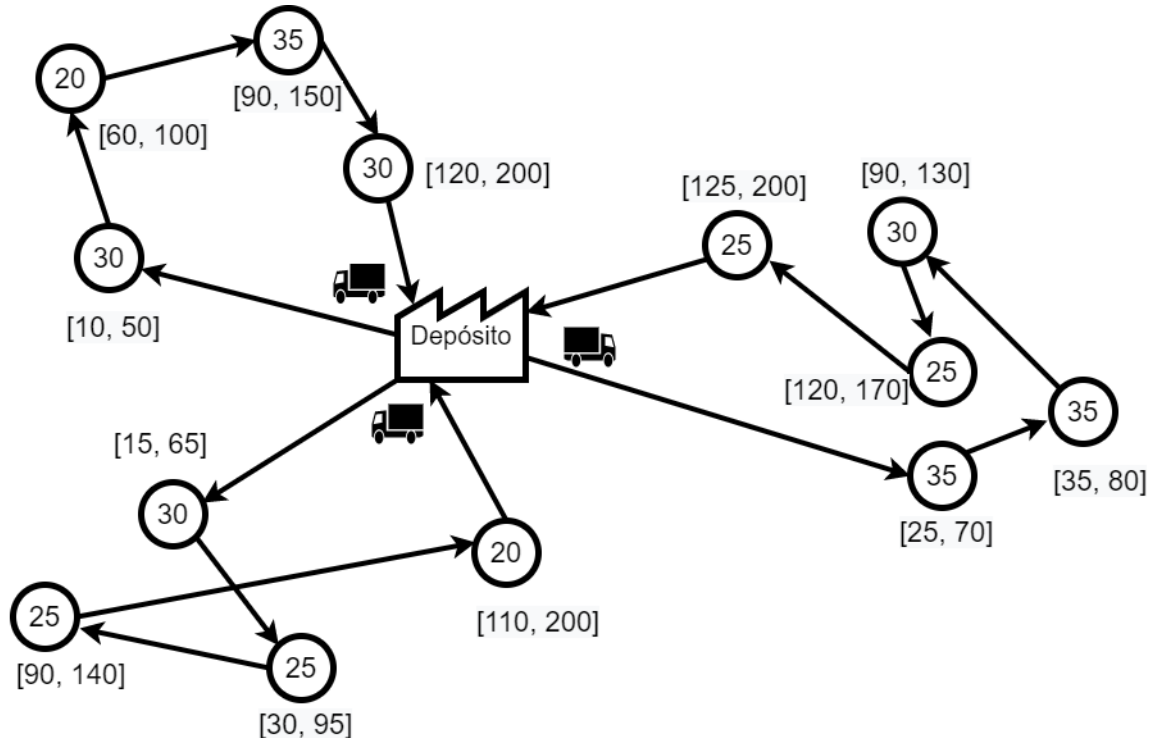


FIGURA 7 – Exemplo de solução do PRV-JA com 3 veículos e 13 clientes. Os valores no interior de cada cliente representam seu tempo de atendimento e os valores entre colchetes o início e fim das janelas de atendimento.

Os valores dados pela janela de atendimento indicam os instantes de tempo limites para o início do serviço. Isso significa que se um veículo chegar no cliente i no instante b_i ele pode realizar o serviço normalmente e sairá desse cliente no instante $b_i + s_i$. Um veículo é penalizado caso ele chegue em um cliente após o fechamento de sua janela de atendimento. Se ele chegar antes do início da janela, a_i , o veículo fica ocioso aguardando até o horário de abertura. Além das restrições de janela de tempo, o problema ainda considera que cada veículo tem uma capacidade máxima Q e uma jornada de trabalho máxima T . Com isso, as rotas estabelecidas para cada veículo devem terminar em um tempo inferior a T e o somatório das demandas d_i dos clientes atendidos por cada veículo não pode ultrapassar Q .

Esse problema pode ser formulado como um modelo de fluxo de rede com dois tipos de variáveis (TOTH; VIGO, 2014): $x_{i,j,k}$ é uma variável binária que assume valor 1 se o arco (i, j) é percorrido pelo veículo k , e 0 caso contrário; e a variável de tempo $\tau_{i,k}$ especifica o instante de tempo em que o veículo k inicia o atendimento em v_i . A formulação matemática é dada pela Equação 4.1:

$$\begin{aligned}
 \text{min:} & \sum_{k \in K} \sum_{(i,j) \in A} c_{i,j} x_{i,j,k} & (1) \\
 \text{sujeito a:} & \sum_{k \in K} \sum_{j \in \delta^+(i)} x_{i,j,k} = 1 & \forall i \in N & (2) \\
 & \sum_{j \in \delta^+(0)} x_{0,j,k} = 1 & \forall k \in K & (3) \\
 & \sum_{i \in \delta^-(j)} x_{i,j,k} - \sum_{i \in \delta^+(j)} x_{j,i,k} = 0 & \forall k \in K, j \in N & (4) \\
 & \sum_{i \in \delta^-(0)} x_{i,0,k} = 1 & \forall k \in K & (5) \\
 & \tau_{i,k} + s_i + t_{i,j} - \tau_{j,k} \leq (1 - x_{i,j,k}) M_{i,j} & \forall k \in K, (i, j) \in A & (6) \\
 & a_i \leq \tau_{i,k} \leq b_i & \forall k \in K, i \in V & (7) \\
 & \sum_{i \in N} \sum_{j \in \delta^+(i)} d_i x_{i,j,k} \leq Q_k & \forall k \in K & (8) \\
 & x_{i,j,k} \in \{0, 1\} & \forall k \in K, (i, j) \in A & (9)
 \end{aligned} \tag{4.1}$$

onde $M_{i,j}$ são constantes grandes, que podem assumir o valor de $\max(b_i + s_i + t_{i,j} - a_j, 0)$. $\delta^+(i) \in A$ é o subconjunto de arestas com uma das pontas no vértice $i \in V$, sendo $\delta^+(i)$ e $\delta^-(i)$ os subconjuntos de arcos que chegam em i e saem de i , respectivamente.

A função objetivo (1) minimiza o custo total. As restrições (2) garantem que cada cliente é atendido por um único veículo. As restrições (3-5) definem os fluxos de cada veículo. As restrições (6) garantem a ordem de atendimento. É importante ressaltar que o valor de $\tau_{i,k}$ é irrelevante quando o cliente v_i não é atendido pelo veículo k . Em seguida, as restrições (7) garantem que os atendimentos ocorrem dentro das janelas de atendimento e as restrições (8) que a capacidade de cada veículo é

respeitada. Por fim, as restrições (9) indicam que as variáveis $x_{i,j,k}$ devem assumir valores binários.

A solução desse modelo por meio de algoritmos exatos requer muito esforço computacional e só é viável em instâncias pequenas. Por isso, metaheurísticas foram usadas para resolver o problema.

4.2 AVALIAÇÃO DO EFEITO DA INCERTEZA NA SOLUÇÃO DO PROBLEMA DE ROTEAMENTO DE VEÍCULOS COM JANELA DE ATENDIMENTO

A primeira etapa do trabalho consistiu em resolver o PRV-JA determinístico e posteriormente avaliar como incertezas nos tempos de viagem e atendimento influenciam na solução.

Para isso, um ILS foi usado para resolver o problema e então a Simulação de Monte Carlo foi aplicada na solução final para avaliar quantos clientes seriam violados, ou seja, teriam o atendimento atrasado, como mostra a Figura 8.

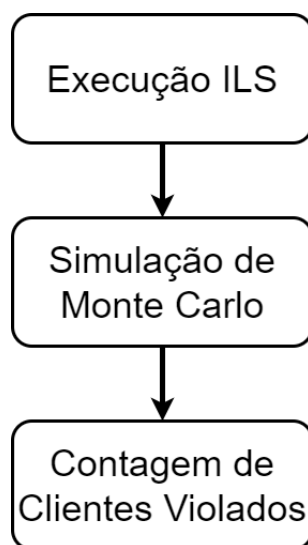


FIGURA 8 – Fluxograma da avaliação do efeito da incerteza.

4.2.1 Representação da Solução

Uma solução candidata foi representada como uma lista de rotas, cada uma contendo a trajetória de um veículo. Uma rota contém uma lista com a sequência de clientes atendidos por aquele veículo, como mostra a Figura 9. Uma solução pode conter qualquer número de rotas com qualquer número de clientes, desde que todos os clientes estejam na solução e que cada um esteja atribuído a um único veículo.

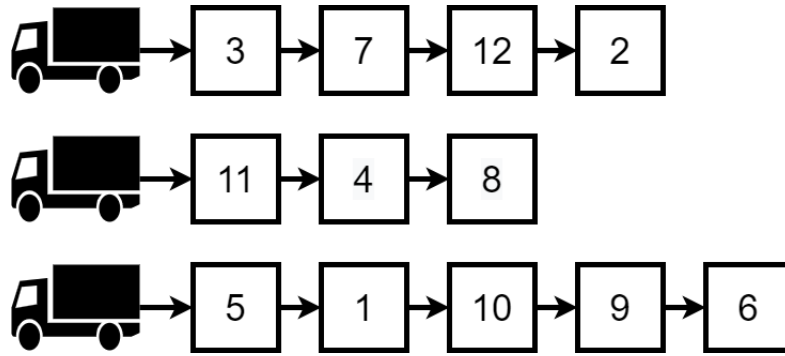


FIGURA 9 – Exemplo de solução candidata com três veículos e doze clientes.

O custo da solução é dado pela Equação 4.2. Ela considera a distância total percorrida pelos veículos e penalidades são adicionadas sempre que alguma restrição é violada, ou seja, quando a capacidade dos veículos ou jornada de trabalho é excedida ou caso algum serviço inicie após o fechamento da janela de atendimento. Quando um veículo fica ocioso esperando a abertura da janela de um cliente não há penalidade, pois isso reduz o número de clientes atendidos por veículo, aumenta a quantidade de veículos necessários e indiretamente aumenta o custo da solução.

$$\begin{aligned}
 f(S) = & \sum_{(i,j) \in A} c_{i,j} x_{i,j} + \sum_{R \in S} (P_1 \max(0, \sum_{i \in R} d_i - Q) + P_2 \max(0, t_0 - T)) \\
 & + P_3 \sum_{i \in R} \max(0, t_i - b_i) + P_4 \sum_{i \in R} d_{i[t_i > b_i]}
 \end{aligned} \tag{4.2}$$

onde:

S : solução composta por um conjunto de rotas;

R : uma rota pertencente a S ;

$c_{i,j}$: custo de viagem da aresta $(i, j) \in A$, dado pela distância entre os nós.

$x_{i,j}$: variável que indica se a aresta $(i, j) \in A$ é usada ($x_{i,j} = 1$) ou não ($x_{i,j} = 0$);

t_i : instante de tempo em que o veículo chega no cliente i . t_0 indica o instante de tempo para estar de volta no depósito. Esse valor é calculado com base nos tempos de viagem, serviço e espera;

P_1 : coeficiente de penalidade por violação de capacidade do veículo;

P_2 : coeficiente de penalidade por violação do tempo de rota;

P_3 : coeficiente de penalidade por tempo de violação das janelas de atendimento;

P_4 : coeficiente de penalidade por demanda com violação das janelas de atendimento dos clientes violados.

Valores arbitrários foram atribuídos para as penalidades. P_1 e P_2 assumiram valor 100 para cada unidade de demanda e instante de tempo, respectivamente. O valor usado para as penalidades P_3 e P_4 foi 10 para cada instante de tempo de atraso e unidade de demanda dos clientes com atraso. Os clientes com atraso possuem dois fatores de penalidade para priorizar o atendimento daqueles com maior demanda, mas evitar grandes atrasos em qualquer tipo de cliente.

4.2.2 Operadores do ILS

4.2.2.1 Criação da População Inicial

Os indivíduos da população inicial são criados a partir de uma adaptação da Heurística de Inserção Sequencial (*Sequential Insertion Heuristic* - SIH) proposta por Solomon (1987) especificamente para o PRV-JA.

O método cria uma rota por vez e insere os clientes de forma gulosa. Um dos clientes é escolhido para ser usado como semente da primeira rota. O custo de inserção de cada cliente não roteado κ em cada posição da rota em construção é calculado de acordo com a Equação 4.3 e então o cliente com menor custo é inserido na melhor posição. Esse processo é repetido até que a inserção de qualquer cliente torne a rota inactível. Quando isso ocorre, um novo cliente não roteado é escolhido como semente e uma nova rota é construída. O processo completo é repetido até que todos clientes sejam atribuídos a alguma rota.

$$e_{i,j}^{\kappa} = \alpha_1(c_{i,\kappa} + c_{\kappa,j} - w \times c_{i,j}) + \alpha_2(\tau_{j_{\kappa}} - \tau_j) + \alpha_3(b_{\kappa} - \tau_{\kappa}) \quad (4.3)$$

A primeira parcela da equação avalia o acréscimo de custo ao inserir κ entre i e j . Os valores τ_j e $\tau_{j_{\kappa}}$ na segunda parcela representam o instante em que o atendimento inicia no cliente j e qual seria esse instante caso o cliente κ seja inserido imediatamente antes de j , respectivamente. Dessa forma, a segunda parcela calcula o acréscimo de tempo na rota caso o cliente κ seja inserido antes de j e a última parcela visa minimizar o tempo de espera.

Dois critérios são usados para definir o cliente semente de cada rota. Pode ser escolhido o cliente não roteado mais distante do depósito ou o cliente com a janela de atendimento mais próxima de terminar. A cada nova rota criada um dos dois critérios é adotado aleatoriamente com igual probabilidade.

Uma alteração foi feita em relação ao algoritmo original para acrescentar variabilidade às soluções iniciais. Ao invés de selecionar sempre o cliente com menor custo, uma lista de candidatos é criada dentre aqueles com os menores custos de inserção e um cliente aleatório dessa lista é selecionado.

Seguindo as recomendações de Solomon (1987), os valores dos parâmetros w , α_1 , α_2 e α_3 usados na Heurística de Inserção Sequencial são 1, 0.4, 0.4 e 0.2, respectivamente.

4.2.2.2 Busca Local

O VND é usado na etapa de busca local do ILS. As estruturas de vizinhança usadas são mostradas a partir do exemplo da Figura 10. A configuração inicial (a) consiste de uma solução do PRV com três rotas e 12 clientes. Os movimentos aplicados por cada operador são explicados e a configuração resultante é apresentada nas outras figuras.

- *Or-Opt*: Estrutura intra-rota que remove um cliente da rota e insere novamente na posição de menor custo. Na Figura 10 (b) o cliente 5 é colocado antes do cliente 4;
- *Two-Opt*: Estrutura intra-rota que inverte a ordem de atendimento dos clientes posicionados entre dois nós específicos. Na Figura 10 (c) a rota é invertida entre os clientes 8 e 11;
- *Shift 1-0*: Estrutura inter-rota que transfere um cliente de uma rota aleatória para outra. Na Figura 10 (d) o cliente 5 é transferido para outra rota.

Testes preliminares mostraram que a melhor ordem para ser usada pelo VND é: *Shift 1-0*, *Or-opt*, *Two-opt*. A estratégia *first improvement* é usada na busca para cada estrutura.

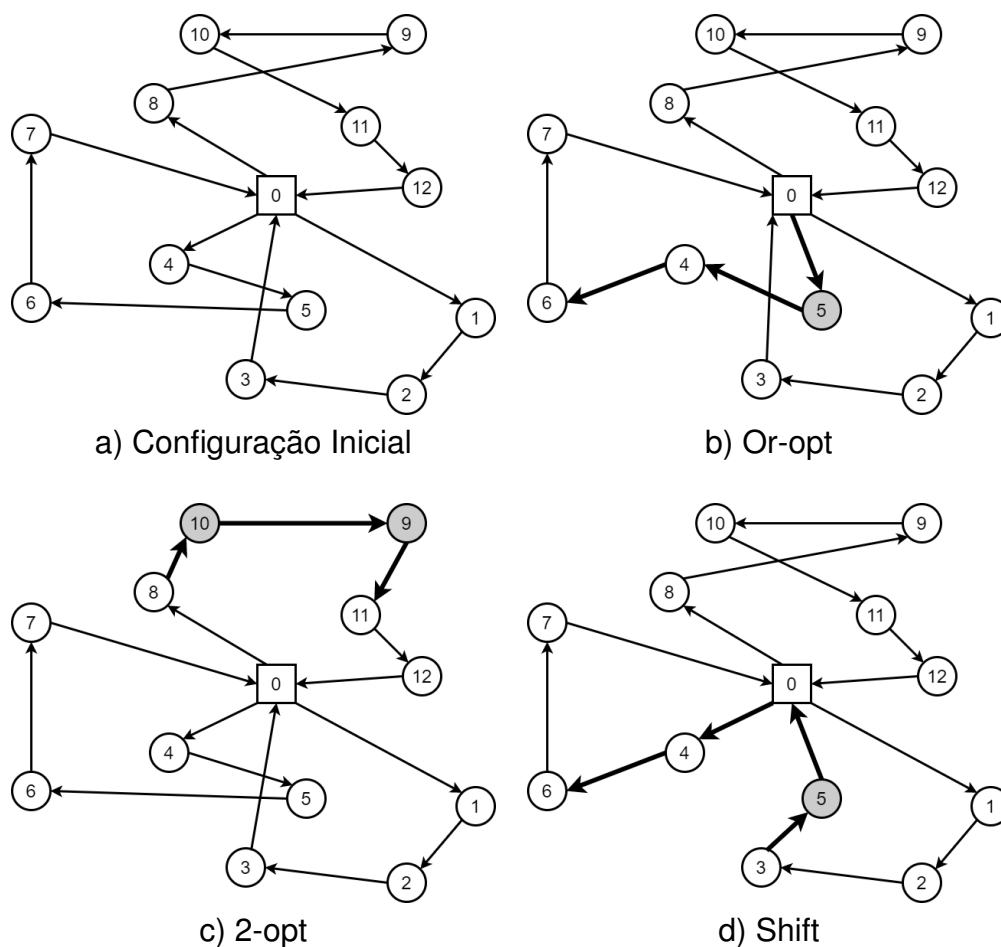


FIGURA 10 – Exemplo dos operadores de busca local aplicados na configuração inicial (a)

4.2.2.3 Heurísticas de Perturbação

As estruturas de perturbação usadas pelo ILS no trabalho foram:

- *Multi-Shift*: Realiza n vezes o movimento *Shift 1-0*, onde um cliente aleatório é transferido de uma rota para outra;
- *Multi-Swap*: Realiza n vezes o movimento *Swap 1-1*, onde duas rotas são escolhidas aleatoriamente e um cliente de cada são trocados. Na Figura 11 (b) as rotas dos clientes 3 e 5 são trocadas;
- *Empty Route*: Remove um veículo ao transferir todos os seus clientes para outras rotas. Na Figura 11 (c) a rota dos clientes 1, 2 e 3 é removida e seus clientes são distribuídos entre as demais rotas;
- *Split Route*: Divide uma rota em duas, adicionando um veículo à solução. Na Figura 11 (d) os clientes da rota do topo são divididos entre duas novas rotas.

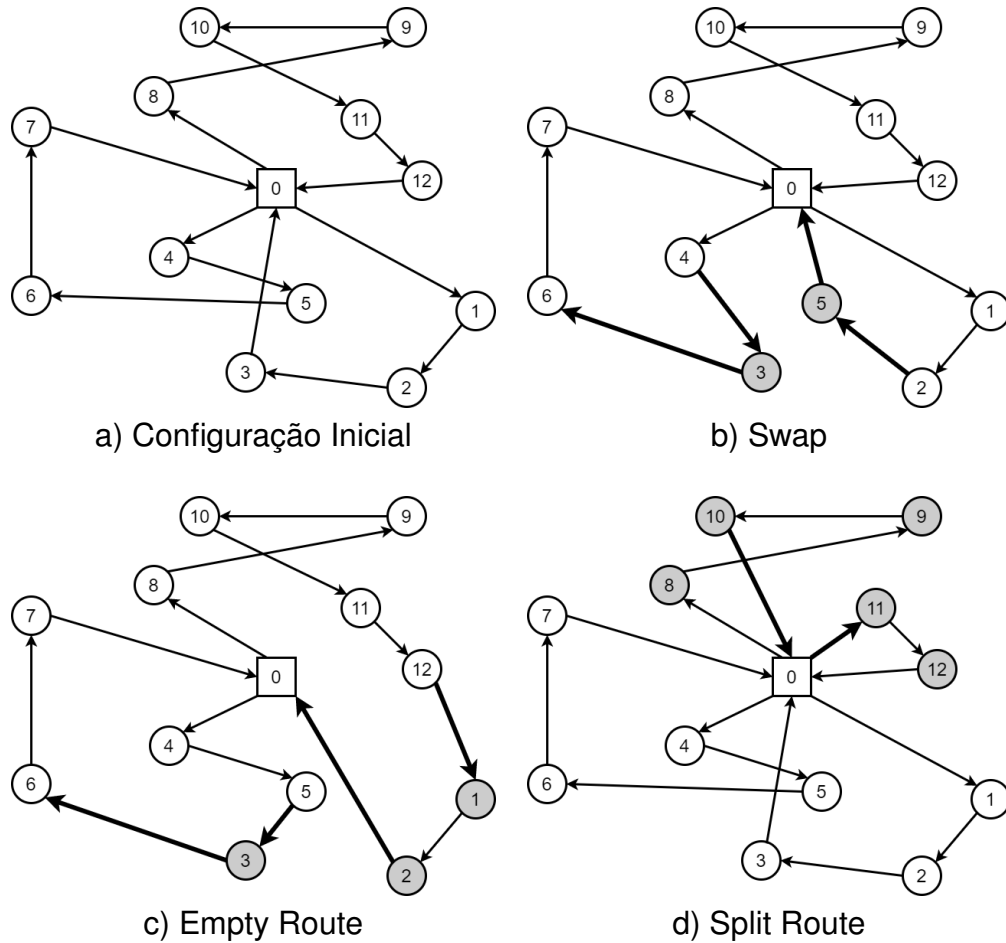


FIGURA 11 – Exemplo dos operadores de perturbação aplicados na configuração inicial (a)

A cada iteração um dos operadores é escolhido aleatoriamente para ser aplicado. As estruturas *Multi-Shift* e *Multi-Swap* na perturbação do ILS fazem dois movimentos cada vez que são executadas. Como as alterações que realizam na solução são suaves, foram usadas com probabilidade de 40% cada. Os operadores *Empty Route* e *Split Route* realizam alterações mais drásticas e são executados com probabilidade 10% cada. Um mínimo local se torna solução corrente sempre que seu custo é o menor conhecido ou quando foram executadas 15 buscas locais sem melhora. Todos os parâmetros foram escolhidos de forma arbitrária. A solução ótima é dada pelo mínimo local com menor custo. Um limite de 1000 iterações é usado como critério de parada.

4.2.3 Simulação de Monte Carlo

A simulação de Monte Carlo é usada para simular variações de tempo de viagem e serviço. Após a execução da otimização, os tempos de viagem e atendimento determinísticos são substituídos por FDPs. Então, os tempos de chegada em cada cliente de cada rota da solução final são calculados a partir de amostras aleatórias das FDPs. Uma falha é contabilizada sempre que um veículo chega em um cliente após o

término de sua janela de atendimento ou quando a jornada de trabalho é excedida. A simulação é realizada diversas vezes para possibilitar a observação do comportamento das soluções ao se considerar as incertezas, conforme o Algoritmo 7.

Algoritmo 7: Simulação de Monte Carlo

Entrada: Solução s , Número de repetições N
Saida: Tempo Médio Violado tmv , Quantidade Média de Clientes Violados mcv

```

1  $tv = 0$  // Tempo total violado
2  $cv = 0$  // Número de clientes violados
3 para  $i = 1$  até  $N$  faça
4   para cada  $rota \in s$  faça
5      $t = 0$  // Tempo de rota
6      $ca = v_0$  // Cliente anterior inicia no depósito
7     para cada  $Cliente\ c \in rota$  faça
8        $t = \max(a_c, t + \text{AmostraTempoViagem}(ca, c));$ 
9       se  $t > b_c$  então
10         $tv = tv + t - b_c;$ 
11         $cv = cv + 1;$ 
12         $t = t + \text{AmostraTempoAtendimento}(c);$ 
13         $ca = c;$ 
14       $t = t + \text{AmostraTempoViagem}(c, v_0);$ 
15      se  $t > T$  então
16         $tv = tv + t - T;$ 
17         $cv = cv + 1;$ 
18  $tmc = tv/N;$ 
19  $mcv = cv/N;$ 

```

As distribuições de probabilidade adotadas foram a *Lognormal* para os tempos de viagem e a *Gamma* para os tempos de serviço, assim como Gutierrez et al. (2018). Essas distribuições são adequadas para representar os parâmetros do problema por não apresentarem valores negativos e possuírem 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. A distribuição *Gamma* modela dados assimétricos à direita, a maioria dos casos apresenta um valor semelhante, mas alguns apresentam valores maiores.

Os parâmetros das distribuições foram ajustados de tal forma que o valor esperado de cada uma delas fosse igual ao valor determinístico de cada tempo, mantendo assim similaridade com os resultados originais. Dessa forma, as distribuições foram criadas considerando a média igual ao valor determinístico e a variância ajustada para

obter um Coeficiente de Variação (CV) igual a 0.2, de forma similar a Gutierrez et al. (2018). O CV é uma métrica de dispersão dada pela divisão entre o desvio padrão σ_x e a média μ_x de uma população, conforme Equação 4.4

$$CV = \frac{\sigma_x}{\mu_x} \quad (4.4)$$

Espera-se que haja estabilidade nos parâmetros de saída após diversas simulações. O número de simulações N necessário depende da complexidade do sistema e da variabilidade das entradas (BILLINTON; LI, 1994). Para a avaliação do efeito de incerteza cada solução foi simulada 1000 vezes após a otimização.

4.2.4 Base de dados

As instâncias de teste criadas por Solomon (1987) foram usadas para testar o modelo proposto. Essa base é composta por 56 problemas do PRV-JA com tempos determinísticos divididos em 6 conjuntos. Cada problema é composto por um depósito com frota homogênea e 100 clientes. Os grupos R1 e R2 possuem clientes com posições geográficas geradas aleatoriamente (do inglês *randomized*), os grupos C1 e C2 possuem os clientes agrupados (dos inglês *clustered*) e os grupos RC1 e RC2 contém metade dos clientes agrupados e metade gerados aleatoriamente. Os clientes dos grupos C1, R1 e RC1 possuem janela de atendimento estreita e os veículos possuem jornada de trabalho curta, o que faz com que poucos clientes sejam atendidos por veículo. Os grupos C2, R2 e RC2 possuem clientes com janelas folgadas e os veículos possuem jornada de trabalho longa, permitindo que um veículo atenda vários clientes.

Apesar de ser uma base de dados antiga, ela ainda é usada atualmente para testar modelos propostos para resolver o PRV-JA. Além disso, ela pode ser facilmente adaptada para incluir os parâmetros de incerteza de tempo e avaliar seu efeito.

Os testes foram realizados em um Ubuntu 19 com processador Intel Xeon E5-2640 2.5GHz com 8 núcleos e 16GB de memória RAM. Todos os problemas considerando os 50 primeiros clientes de cada instância para obter soluções com menor esforço computacional. Como uma heurística foi usada na otimização, cada problema foi resolvido 30 vezes para garantir a consistência dos resultados.

4.3 MODELO MULTIOBJETIVO PARA O PROBLEMA DE ROTEAMENTO DE VEÍCULOS COM JANELA DE ATENDIMENTO E TEMPOS ESTOCÁSTICOS

Após a análise do efeito de incerteza, um modelo multiobjetivo usando *sim-heuristics* que considera a minimização de custos e falhas foi proposto.

Algoritmos que minimizam apenas custo tendem a gerar soluções que exploram os limites das janelas de atendimento de alguns clientes. Entretanto, ao adicionar incerteza nos tempos, falhas (atrasos no atendimento) podem ocorrer para clientes cujo atendimento inicia próximo ao fim da janela. Um algoritmo multiobjetivo que considera a minimização de custo e falhas tende a gerar soluções com diferentes custo-benefício entre custo e risco.

O NSGA-II é usado para resolver esse problema. A simulação de Monte Carlo é usada para simular o risco associado a cada solução. Essa abordagem traz maior custo para o algoritmo, mas permite que quaisquer funções de probabilidade sejam adotadas em cada rota.

4.3.1 Operadores NSGA II

4.3.1.1 Criação da população Inicial

A Heurística de Inserção Sequencial usada no ILS é usada para criar a população inicial no NSGA-II. Entretanto, é feita uma segunda alteração em relação ao algoritmo original visando reduzir a probabilidade de atraso ao inserir incerteza e aumentar a diversidade da população inicial. Cada indivíduo criado considera uma parcela diferente da janela de atendimento de cada cliente. Cada janela é reduzida de um valor aleatório entre 0% e 50%, deixando os clientes mais restritos nessa etapa do algoritmo. Entretanto, essa alteração é aplicada apenas durante a criação da população inicial. No restante da execução do NSGA-II são usadas as janelas de atendimento originais, de forma que os clientes tendam a ser menos impactados por atrasos.

4.3.1.2 Operador de Cruzamento

O método usado no cruzamento é o *Best Cost Route Crossover* (BCRC), proposto por Ombuki, Ross e Hanshar (2006). Esse operador foi desenvolvido especificamente para o PRV-JA.

O funcionamento do operador é exemplificado na Figura 12. Os pais P_1 e P_2 são copiados para gerar o par de filhos C_1 e C_2 . A primeira etapa consiste em escolher uma das rotas RP_i de cada pai aleatoriamente. No exemplo da Figura 12 a), a rota com clientes 5 e 6 é escolhida em RP_1 e em RP_2 escolhe-se a rota com os clientes 7 e 3. Então, os clientes escolhidos são removidos no outro filho. Ainda na Figura 12 a), os clientes 5 e 6 são removidos de C_2 e os clientes 7 e 3 removidos de C_1 . Por fim, cada

cliente removido é reinserido na rota e posição de menor custo (Figuras 12 b) e c)). A ordem em que os clientes são escolhidos para reinserção é aleatória. O processo é encerrado após a reinserção de todos os clientes em cada filho C_i (Figura 12 d)).

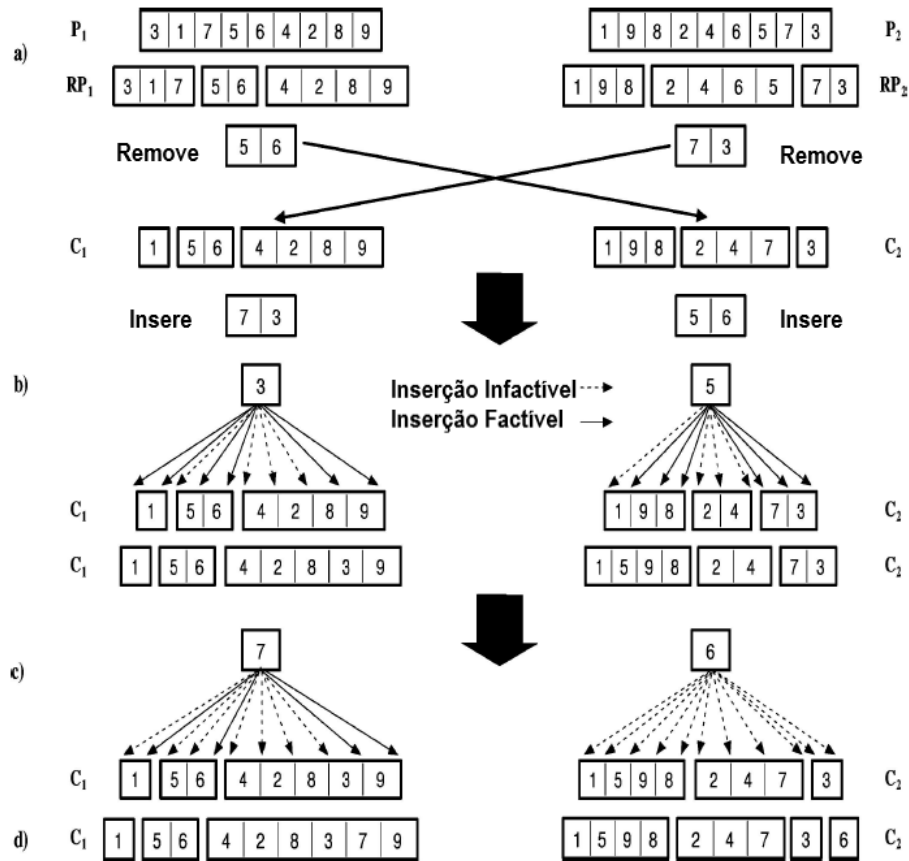


FIGURA 12 – Exemplo de cruzamento usando o BCRC. Figura adaptada de Ombuki, Ross e Hanshar (2006).

4.3.1.3 Operadores de Mutação

As estruturas de perturbação do ILS apresentados em 4.2.2.3 são usados para realizar a mutação no NSGA-II.

4.3.1.4 Hipervolume

O hipervolume da fronteira Pareto estimada é calculado a cada iteração para avaliar a evolução do algoritmo. Ele requer a definição de dois pontos de referência:

- Ponto utópico: As distâncias mínimas conhecidas para cada problema são usadas e considera-se que nenhum cliente tem a janela de tempo violada;

- Ponto *nadir*: O limitante inferior de cada problema é construído considerando que cada cliente é atendido em uma viagem separada e que todas as janelas de atendimento são violadas.

4.3.1.5 Parâmetros do Algoritmo

Os parâmetros do NSGA-II usados no experimento foram:

- Tamanho da população: 50
- Tempo de execução: 2h
- Probabilidade de mutação: 20%
- Número de simulações durante a otimização: 1.000
- Número de simulações após a otimização: 10.000

4.3.2 Planejamento Experimental

A base de dados de Solomon usada no ILS também foi usada para o teste do modelo multiobjetivo. Os testes foram realizados com três problemas de cada classe considerando todos os 100 clientes de cada instância. Cada problema foi executado 10 vezes para garantir a consistência dos resultados.

Além do NSGA-II original, o experimento também foi realizado considerando duas alterações, conforme Figura 13:

1. **NSGAIIL_ILS**: o algoritmo original é usado durante o processo de otimização. Entretanto, após a criação da população inicial o ILS usado no teste anterior é aplicado em um dos indivíduos considerando os tempos determinísticos para tentar encontrar uma solução com menor distância e com isso guiar o processo de busca em direção às soluções de menor custo. Para evitar que o processo iterativo do NSGA-II demorasse muito a começar, o tempo de execução do ILS foi limitado em 30 minutos.
2. **NSGAIIL_VND**: a criação da população inicial original é usada. Durante o processo de otimização, uma busca local com tempos determinísticos é feita em cada indivíduo da nova população ao final de cada iteração para tentar encontrar soluções com menores distâncias. O VND usado na busca local do ILS do teste anterior é usado na busca local.

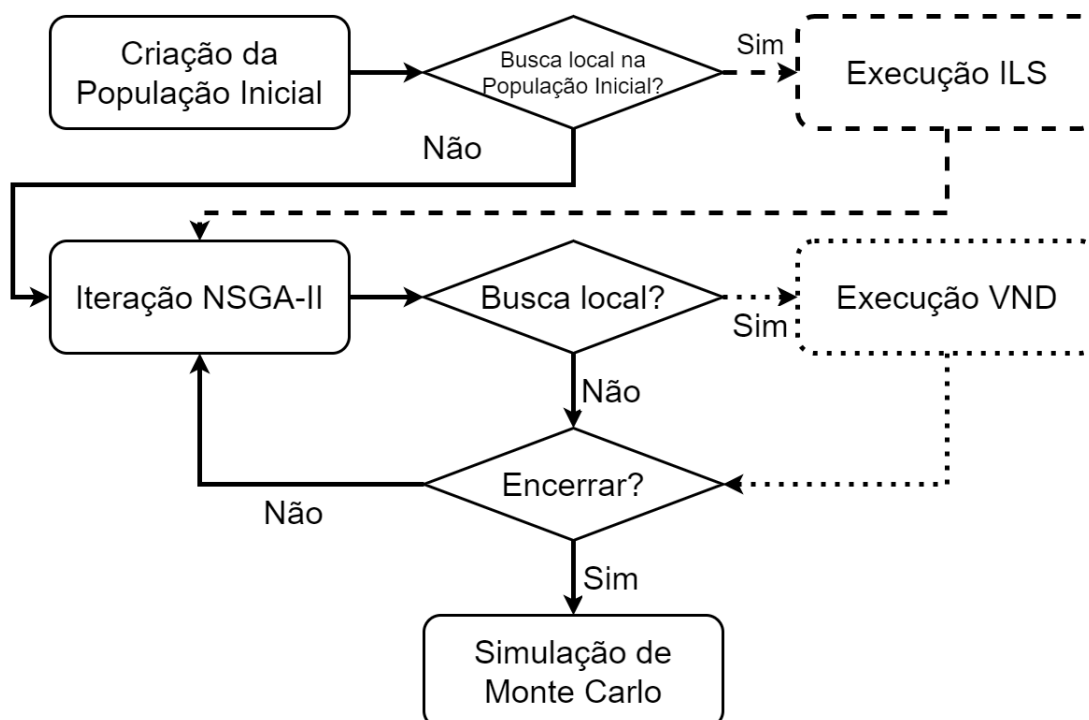


FIGURA 13 – Fluxograma das versões do NSGA-II. As setas contínuas ligam os blocos do algoritmo original. Os blocos tracejados são executados apenas quando o ILS é usado e os pontilhados apenas no algoritmo Memético.

As buscas locais com ILS e VND consideram apenas o objetivo custo na minimização. O risco não é considerando pois seu cálculo é muito caro computacionalmente. Como ele é obtido após a execução da simulação de Monte Carlo, seria necessário realizar uma simulação a cada operação de busca local.

Todos os testes foram executados para todas as versões do NSGA-II. Os experimentos foram executados em um Ubuntu 19 com processador Intel Xeon E5-2640 2.5GHz com 8 núcleos e 16GB de memória RAM.

4.4 COMPARAÇÃO ENTRE ALGORITMOS

Para comparar as três abordagens propostas as métricas de desempenho Hipervolume, Delta e RNI são calculadas para cada fronteira Pareto estimada obtida em cada execução de cada problema para cada algoritmo. Então, uma blocagem é feita para isolar o efeito das diferentes instâncias nos resultados de cada métrica e permitir uma melhor comparação.

Blocagem é uma técnica que visa reduzir o efeito de fatores indesejados ou desconhecidos em análises estatísticas (MONTGOMERY, 2013). Ela é recomendada quando sabe-se desde o princípio do experimento que alguns fatores podem influenciar no valor de algumas variáveis.

Neste trabalho, os diferentes grupos de problemas e instâncias possuem diferentes soluções ótimas, o que gera diferentes fronteiras Pareto e conseqüentemente diferentes resultados para cada métrica de desempenho. A blocagem isola os efeitos dos grupos e instâncias e permite comparar de forma mais justa a performance de cada abordagem. Entretanto, a blocagem só é efetiva se as seguintes premissas forem validadas:

- Uma replicação por bloco;
- Independência entre blocos;
- Aleatoriedade dentro de cada bloco;
- Normalidade dos resíduos;
- Homogeneidade de variância dos resíduos (homocedasticidade dos resíduos).

Por fim, o teste comparações múltiplas de Tukey é usado para identificar se há diferença significativa entre o desempenho de cada par de algoritmos para cada métrica. Esse teste compara os algoritmos par a par e usa a correção de significância de Bonferroni para identificar em quais combinações há diferença significativa. A hipótese nula assume que não há diferença significativa entre os métodos e a hipótese alternativa que há.

5 RESULTADOS

Os resultados estão divididos em duas partes. Primeiro são apresentados os resultados referentes a avaliação do efeito da incerteza, obtidos resolvendo um modelo com um único objetivo. Em seguida, são mostradas as diferentes abordagens para a resolução do problema multiobjetivo considerando a minimização simultânea de custo e risco.

5.1 AVALIAÇÃO DO EFEITO DE INCERTEZA

Os resultados obtidos nos testes computacionais da análise do efeito da incerteza para cada problema são divididos em duas etapas. A primeira consiste na análise da qualidade da otimização, ou seja, na capacidade do método proposto encontrar boas soluções. A segunda etapa analisa o efeito da incerteza em si, avaliando como as soluções se comportam após a simulação de Monte Carlo considerando as incertezas nos tempos.

5.1.1 Resultados da otimização

Os resultados médios encontrados para cada problema são mostrados na Tabela 3 localizada no Apêndice A, juntamente com aos valores de referência das melhores soluções conhecidas apresentadas em VRPTW BENCHMARK PROBLEMS. . . (2005). Os valores da referência são compostos pela menor distância encontrada d e a quantidade de veículos usados v . Os resultados da otimização apresentam a distância percorrida média \bar{d} , sua variância $var(d)$, valor mínimo $min(d)$, diferença de distância em relação à referência Δd , a média de veículos usados \bar{v} e o tempo médio de viagem \overline{tmv} . Os problemas R207, R208 e RC208 não possuem informação de melhor solução conhecida disponível, então a análise dos resultados desses três problemas foi removida.

Os resultados da Tabela 3 mostram que o algoritmo foi capaz de encontrar resultados próximos do ótimo para todas as classes de problema, sendo que a média da distância entre os valores obtidos e o melhor conhecido foi 3.43%. A Figura 14 mostra que os problemas do tipo RC tiveram piores resultados e os C os melhores, especialmente os problemas C2, cujas soluções encontradas foram apenas 0.73% em média piores que a referência. Dentre todas as classes, os problemas do tipo C1 apresentaram a maior variabilidade, e o tipo C2 a menor.

Ao analisar o efeito da duração das janelas de atendimento, observa-se que os problemas dos tipos C e R apresentaram melhores resultados quando as janelas

de atendimento eram maiores, ou seja, as instâncias das classes R2 e C2 tiveram resultados melhores que R1 e C1, respectivamente. Para a classe RC observa-se o inverso, os problemas com janelas de atendimento mais curtas apresentaram os resultados melhores.

O algoritmo encontrou resultados muito piores que a referência apenas para os problemas C104, RC106 e RC203, com soluções aproximadamente 10%, 13% e 12% piores, respectivamente.

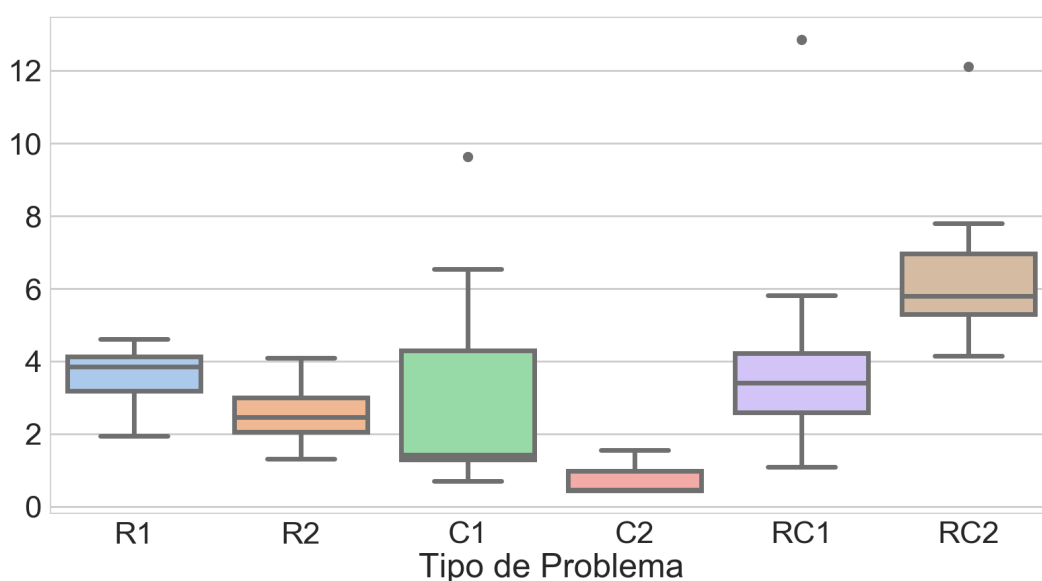


FIGURA 14 – Diferença percentual média de distância total entre resultados encontrados e valores de referência.

A Figura 15 mostra que o número de veículos usados em cada caso também foi muito próximo da quantidade usada na melhor solução conhecida. Os resultados mais próximos da referência foram nos problemas tipo C e os mais distantes no tipo R. A classe C2 usou exatamente a mesma quantidade de veículos da referência em todas as execuções de todos os problemas. Apenas o problema R106 usou uma quantidade de veículos consideravelmente diferente (9.1 em média contra 5 da referência).

Os problemas com janela de atendimento longas usam menos veículos e apresentaram menor variabilidade nas soluções, resultado que condiz com as características dessas instâncias.

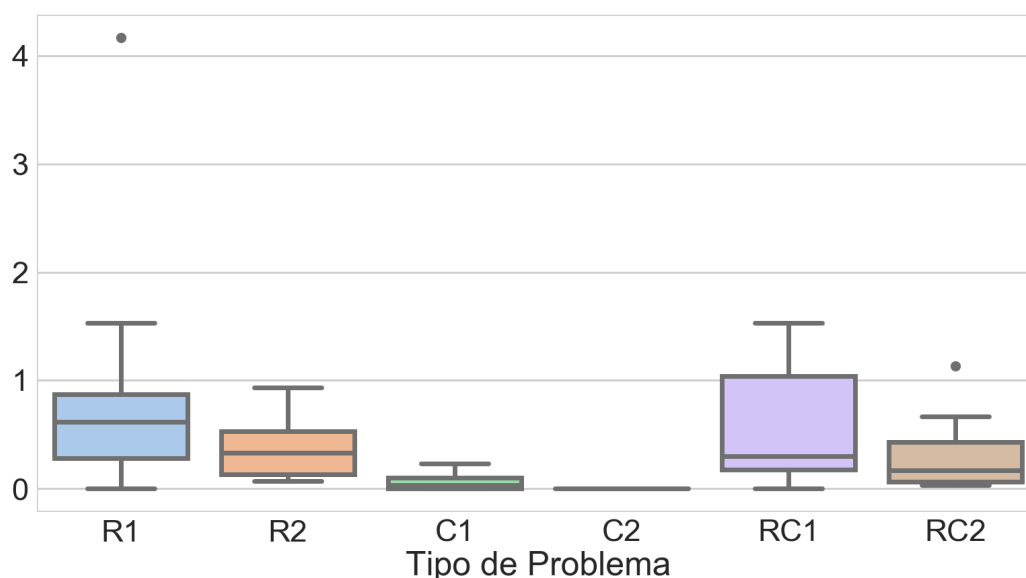


FIGURA 15 – Diferença média de veículos usados entre resultados encontrados e valores de referência.

É importante ressaltar que as restrições foram respeitadas em todas as execuções de todos dos problemas, seja nas janelas de atendimento, jornada de trabalho ou capacidade dos veículos.

5.1.2 Análise do efeito da incerteza nas soluções

Os resultados da simulação são apresentados na Tabela 4 localizada no Apêndice A. O tempo médio de viagem total da solução determinística \overline{tmv} é apresentado na tabela e usado para comparar com os valores obtidos pela simulação. Além dessa informação, as outras colunas da tabela mostram o tempo médio de viagem simulado \overline{tmvs} , a diferença em relação ao tempo determinístico Δtmv , a média de clientes com atraso no atendimento \overline{cv} e a probabilidade de observar pelo menos um atraso na simulação \overline{pf} .

A similaridade dos tempos médios de viagem determinísticos e os estimados pela simulação de Monte Carlo mostram que as distribuições de probabilidade foram adequadamente parametrizadas para o problema.

Os problemas com clientes agrupados apresentaram menor média de clientes violados (0.2 para a classe C1 e apenas 0.002 para a C2), conforme Figura 16. Isso indica que a incerteza nos tempos tem impacto menor quando os tempos de viagem entre clientes são menores, o que tende a ocorrer quando há vários clientes concentrados em um espaço próximo.

A média de clientes violados por problema é de aproximadamente 2 (4%) para as classes R1 e RC1, e 0.3 (0.6%) para R2 e RC2. Esses resultados sugerem que os problemas com janelas mais estreitas (R101, C101 e RC101) foram mais afetados pela incerteza que os problemas com janela de atendimento folgadas. Esse comportamento era esperado, já que quanto menor a janela de atendimento, maior a probabilidade de um atraso gerar uma falha.

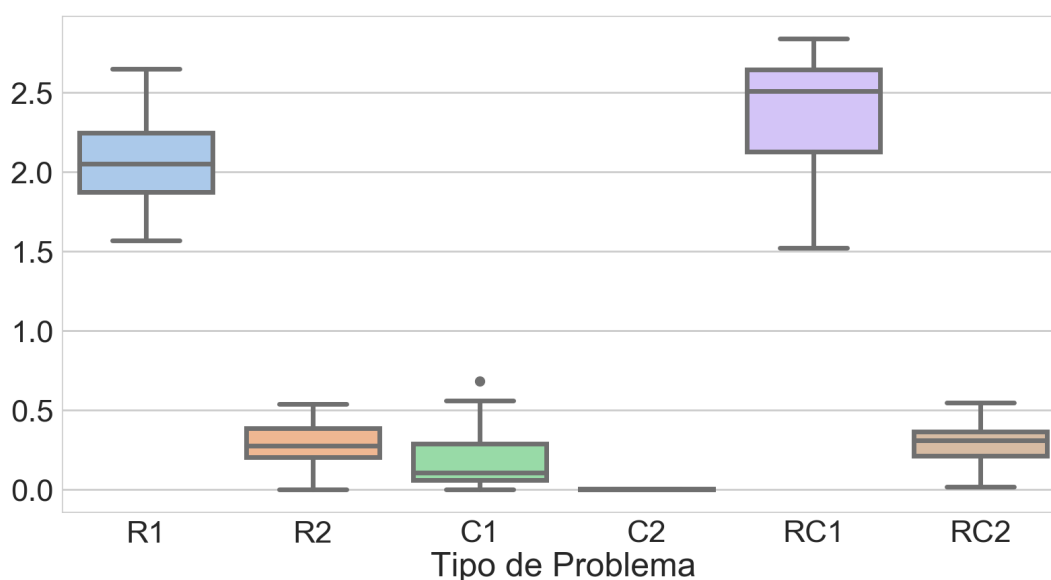


FIGURA 16 – Média de clientes violados por classe de problema.

A probabilidade de observar uma falha na simulação também é maior nos problemas com janela estreita. A Figura 17 mostra que a média de simulações com pelo menos uma falha dos problemas dos tipos R1 e RC1 foi superior a 80%. Os problemas das classes R2 e RC2 apresentaram falhas em aproximadamente 25% das simulações. Novamente, as falhas foram menos presentes nos problema com clientes agrupados, sendo que as simulações na classe C1 apresentaram falhas em 12.6% das execuções e na classe C2 apenas 0.1%.

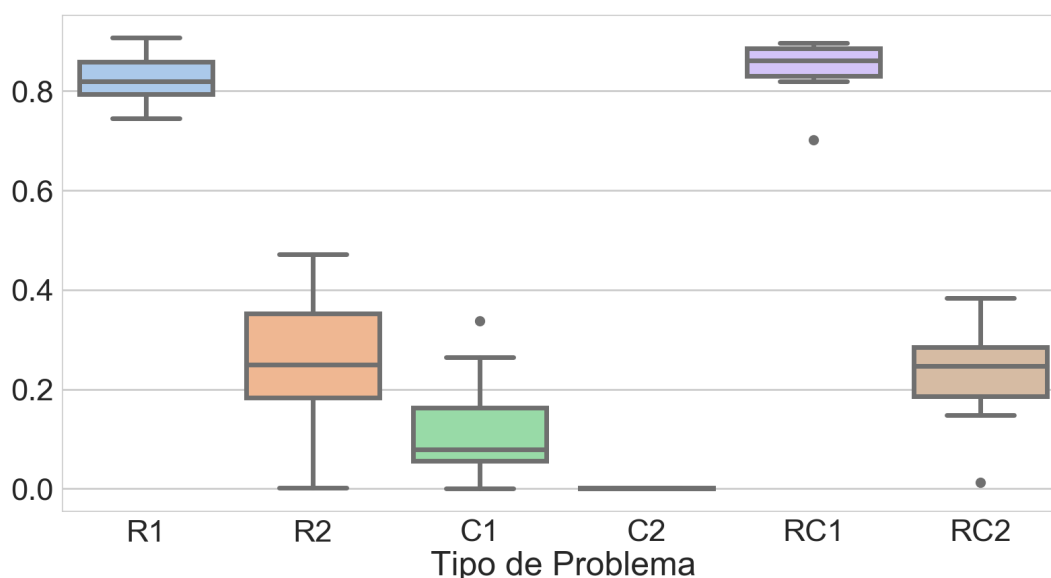


FIGURA 17 – Probabilidade de falha média por classe de problema.

Esses resultados sugerem que nem sempre é necessário considerar incerteza na otimização do PRV-JA. A configuração geográfica e a duração das janelas de tempo dos clientes influenciam no impacto da incerteza. Quando os clientes estão agrupados e principalmente quando a duração das janelas é longa a probabilidade de observar uma falha é baixa, portanto dependendo do nível de aversão ao risco o esforço computacional da inclusão da incerteza no modelo de otimização pode não ser justificável.

5.2 OTIMIZAÇÃO MULTI OBJETIVO

A segunda etapa do trabalho comparou três abordagens multiobjetivo para solução do PRV-JA com incertezas nos tempos de viagem de atendimento usando o algoritmo NSGA-II. A primeira abordagem considera o método puro (denominado NSGAI), a segunda aplica o ILS a um indivíduo da população inicial antes de iniciar o processo de busca (denominado NSGAI_ILS) e a terceira aplica uma busca local usando o VND em cada indivíduo gerado a cada geração (denominado NSGAI_VND). Foram selecionados aleatoriamente três problemas de cada classe e os três algoritmos foram executados dez vezes em cada problema para a comparação dos resultados. As instâncias usadas no experimento foram: C104, C105, C108, C202, C206, C208, R101, R105, R107, R201, R203, R206, RC103, RC105, RC107, RC201, RC202 e RC205.

5.2.1 Resultados da Otimização

As Figuras 18 a 23 mostram as fronteiras Pareto estimadas da execução com maior hipervolume final para cada abordagem em cada problema. Com uma inspeção visual é possível perceber que, em comparação com os algoritmos NSGAII_VND e NSGAII_ILS, o NSGAII encontrou soluções com maior distância para a maioria dos problemas. Em relação à máxima violação de clientes, a inspeção visual não permite perceber uma superioridade ou inferioridade clara de nenhum dos algoritmos.

Assim como nos testes com um objetivo, os problemas da classe C2 apresentaram as menores violações, sendo que em alguns problemas não houve atraso em nenhuma das replicações em pelo menos um dos algoritmos. Por isso, os algoritmos encontraram poucas soluções nos problemas C202, C206 e C208, havendo casos em que foi obtida uma única solução.

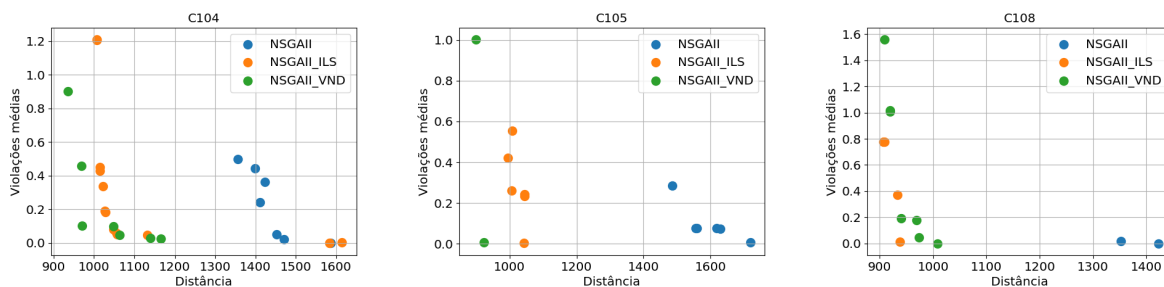


FIGURA 18 – Fronteiras Pareto estimadas dos problemas C1

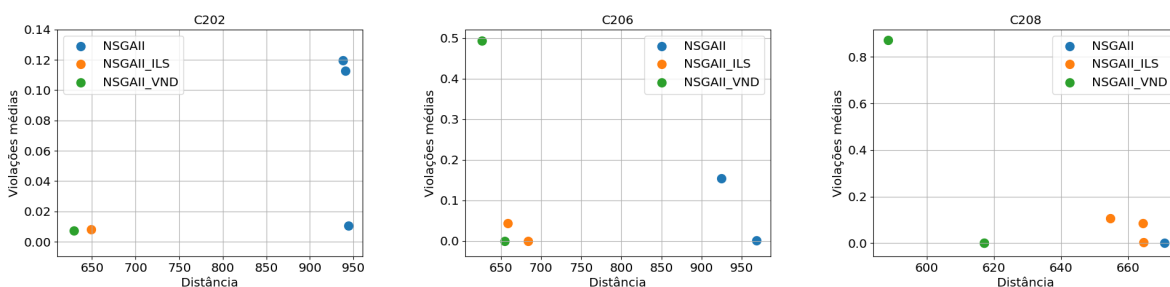


FIGURA 19 – Fronteiras Pareto estimadas dos problemas C2

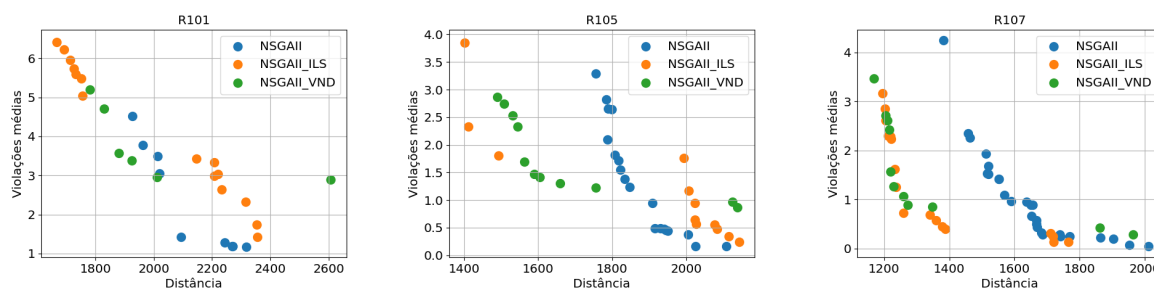
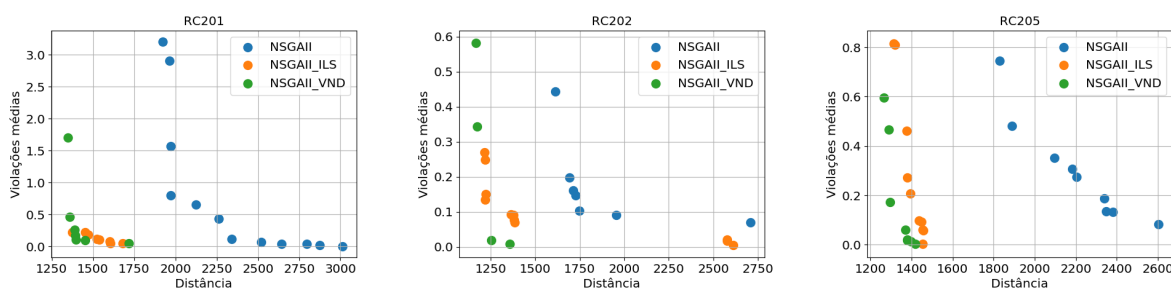
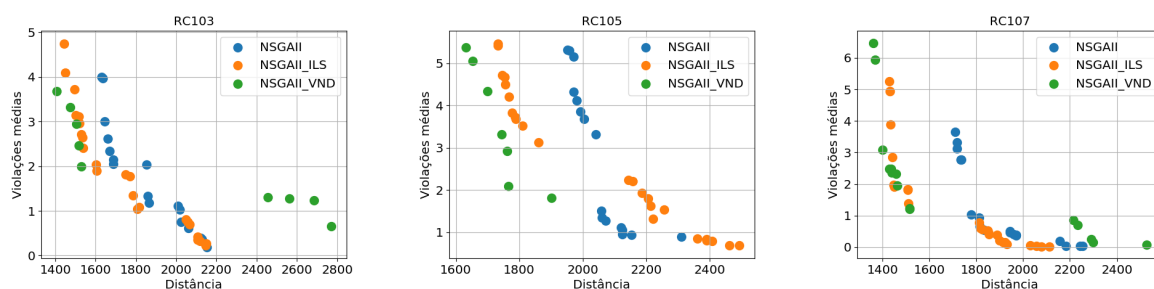
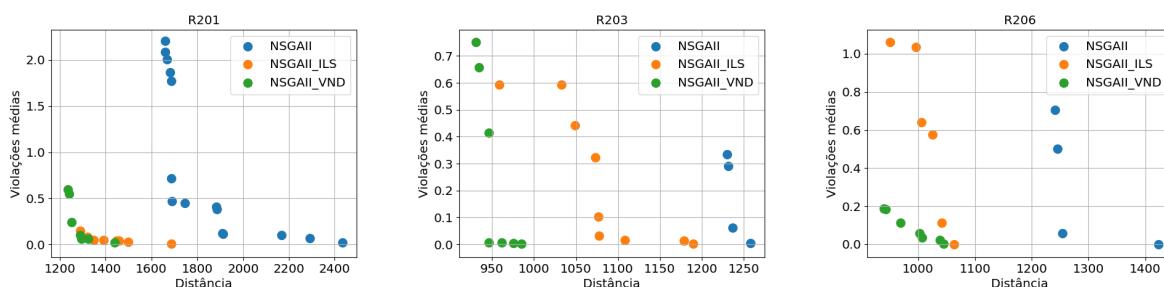


FIGURA 20 – Fronteiras Pareto estimadas dos problemas R1



A Tabela 5 localizada no Apêndice A mostra os resultados médios gerais de cada algoritmo em cada problema após duas horas de execução, como a média da distância mínima encontrada por alguma solução e a violação mínima média encontrada por uma solução.

A Figura 24 mostra quantas gerações em média cada algoritmo fez durante a execução. O algoritmo NSGAI_VND teve uma média bem menor que os outros dois, devido à busca local ao fim de cada geração.

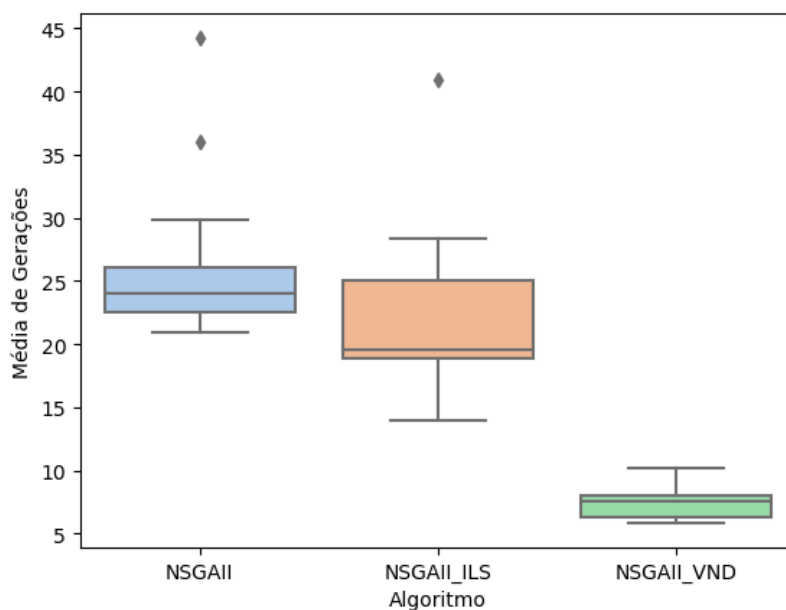


FIGURA 24 – Média de Gerações por algoritmo

A média de soluções encontradas por cada algoritmo em cada problema é mostrada na Figura 25. Assim como o número de gerações, os menores valores foram observados no algoritmo NSGAI_VND. Isso pode ter ocorrido porque a busca local opera apenas visando minimizar a distância, sem considerar as violações, e assim as fronteiras Pareto estimadas podem ter se concentrado em uma região específica do espaço de soluções.

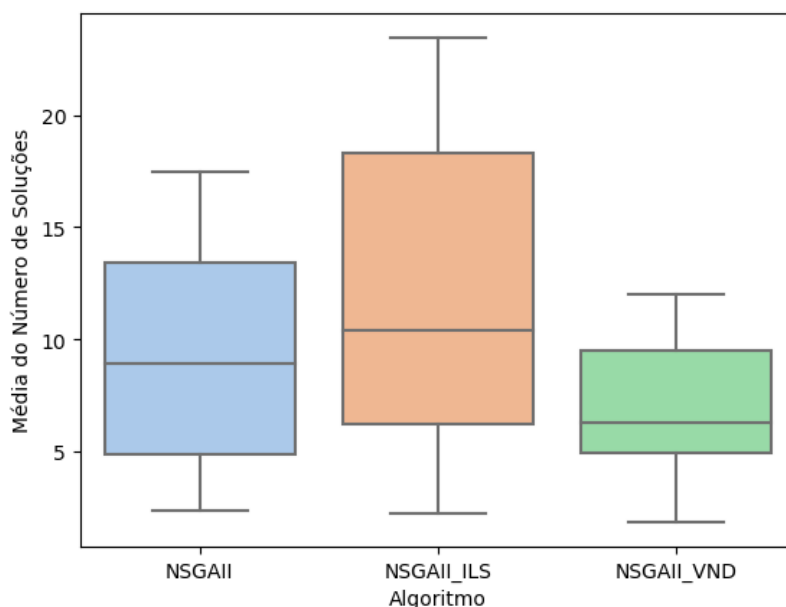


FIGURA 25 – Média do número de soluções por algoritmo

A Figura 26 mostra a diferença entre o Hipervolume médio da população inicial e o valor da métrica ao final da otimização. Diferentemente dos outros resultados, vê-se que o algoritmo NSGAII_VND apresentou a maior progressão de Hipervolume na média. Por outro lado, o NSGAII_ILS mostrou um avanço muito pequeno.

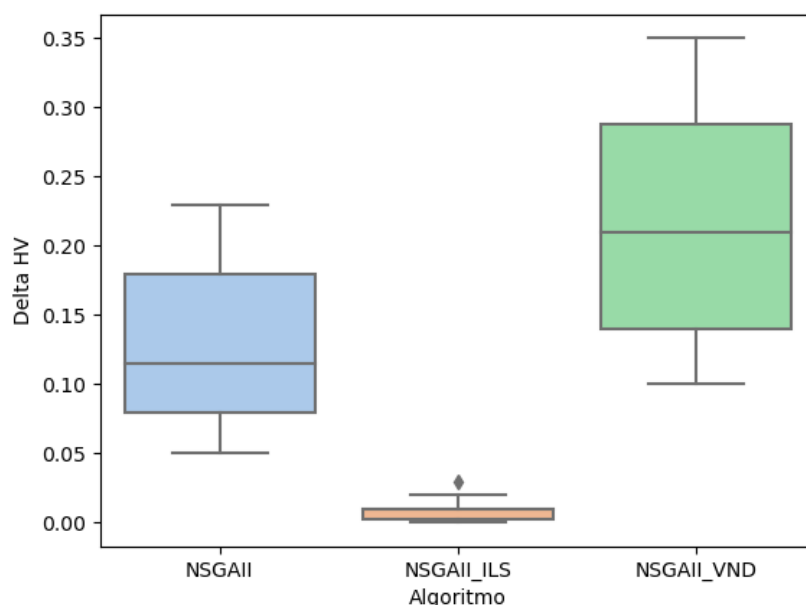


FIGURA 26 – Média do crescimento de HV por algoritmo

O baixo crescimento de Hipervolume do NSGAII_ILS mostra como a solução inicial gerada pelo ILS é capaz de exercer uma boa pressão evolutiva no restante da população. Como ele já inicia o processo evolutivo com uma boa solução, havia a expectativa de que esse método apresentasse o menor crescimento de Hipervolume, porém os resultados mostraram que o aumento foi ainda menor que o esperado, demonstrando a importância de criar uma boa solução inicial na perspectiva de custo. Entretanto, não era possível prever que isso ocorreria antes da execução dos experimentos, e por isso o algoritmo foi executado da mesma forma que os outros. Em testes futuros, faz sentido considerar um critério de parada que resulte em um menor tempo de execução do processo evolutivo.

5.2.2 Comparação entre os algoritmos

Uma comparação estatística é feita em cada métrica de desempenho para analisar melhor a performance das três abordagens propostas. Os problemas usados nos testes tem diferentes características e levam a diferentes soluções independente do método de solução. Por isso, uma análise bloqueando o efeito das instâncias foi

realizada para remover o efeito que diferentes problemas podem gerar e permitir a comparação dos métodos propostos isoladamente.

Os valores de R quadrado da Tabela 1 mostram que os modelos blocados para o Hipervolume e RNI ficaram muito aderentes aos dados originais, mas para o Delta o modelo não ficou tão adequado. Porém, o teste de Shapiro para averiguar a normalidade dos resíduos mostrou que todos os modelos seguem uma distribuição normal considerando um nível de significância de 5%.

TABELA 1 – Resultados da blocagem para comparação de Algoritmos

Métrica	R quadrado	p-valor Shapiro dos resíduos
Hipervolume	0.950566	0.9343
Delta	0.6232795	0.07889
RNI	0.8880327	0.7871

A normalidade dos resíduos também pode ser checada nas Figuras 27 a 29. Em todos os casos quase todos os pontos se encontram próximos da linha de normalidade e apenas um ponto para RNI está fora dos intervalos de confiança.

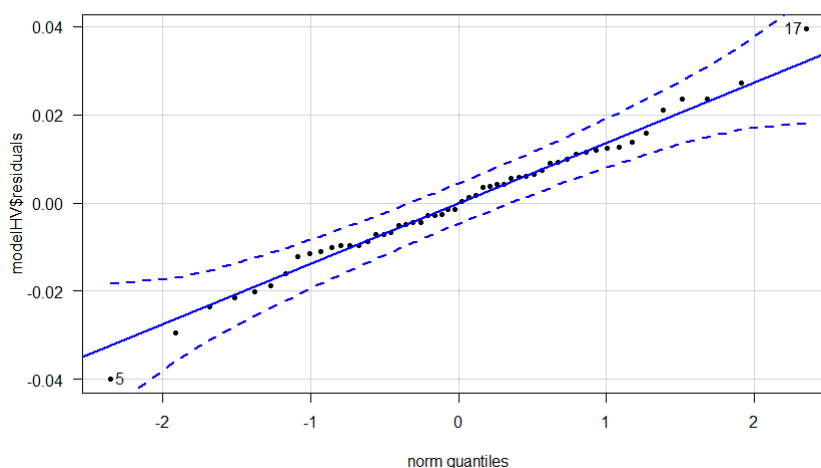


FIGURA 27 – Normalidade dos resíduos para o modelo ajustado com a blocagem para o HV.

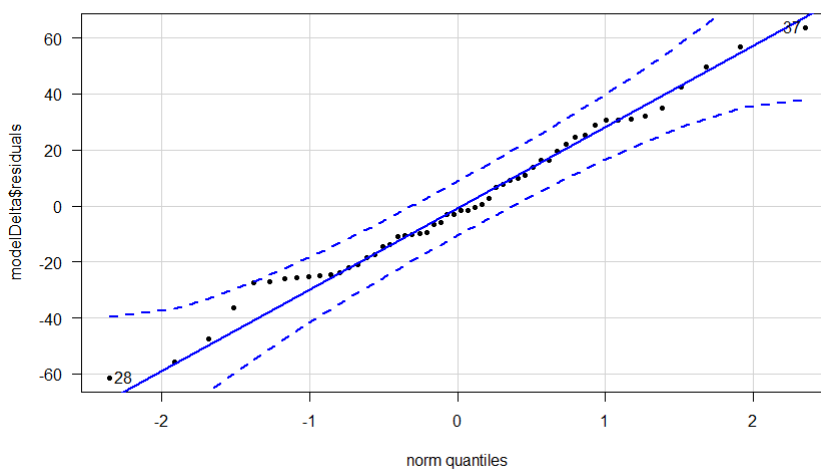


FIGURA 28 – Normalidade dos resíduos para o modelo ajustado com a blocagem para o Delta.

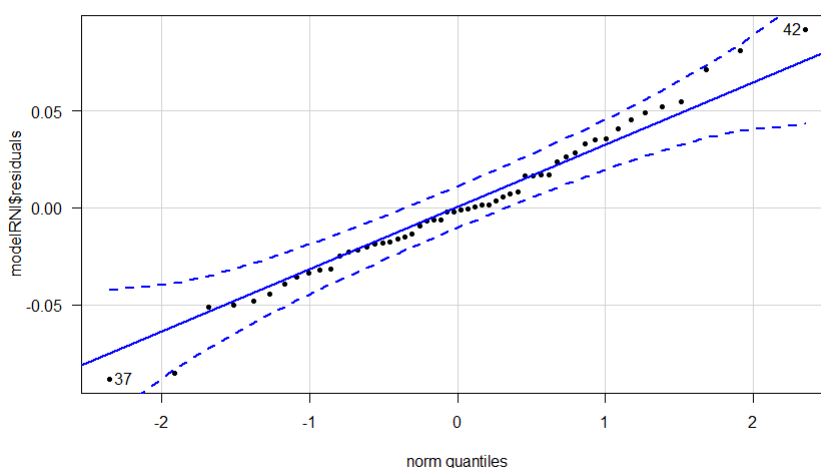


FIGURA 29 – Normalidade dos resíduos para o modelo ajustado com a blocagem para o RNI.

Com isso, optou-se por seguir avaliando as demais premissas da blocagem para usar esses modelos na comparação dos algoritmos. Os gráficos de resíduos por valores ajustados das Figuras 30 a 32 mostram que os resíduos estão homogeneamente distribuídos em torno do zero em todas as métricas. Como não é possível observar nenhuma tendência nos pontos de nenhum dos gráficos, há indícios de que a variância dos resíduos é igual para todos os algoritmos, ou seja, há indícios de que eles sejam homocedásticos.

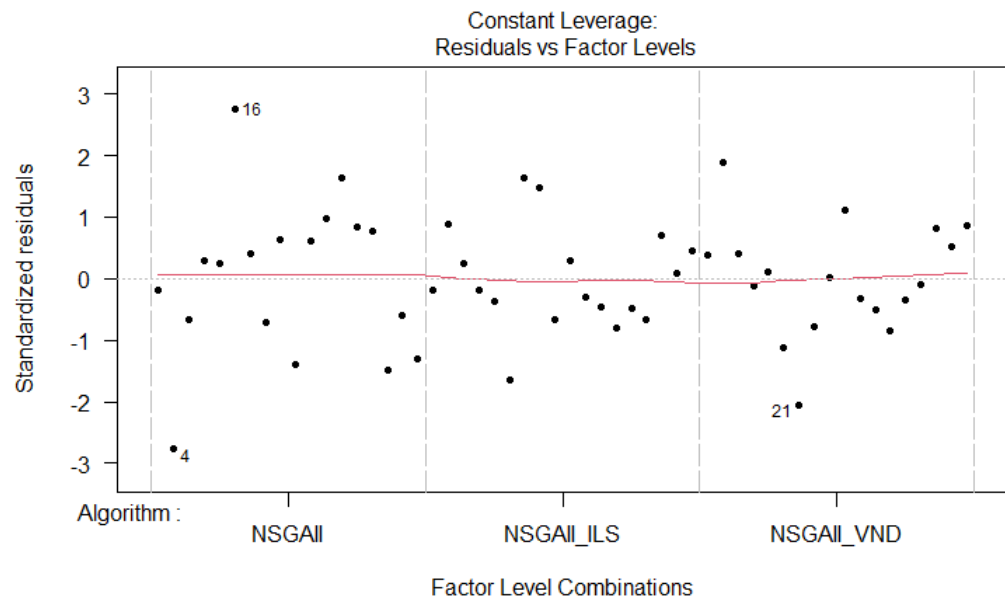


FIGURA 30 – Resíduos por cada algoritmo com blocagem para o HV.

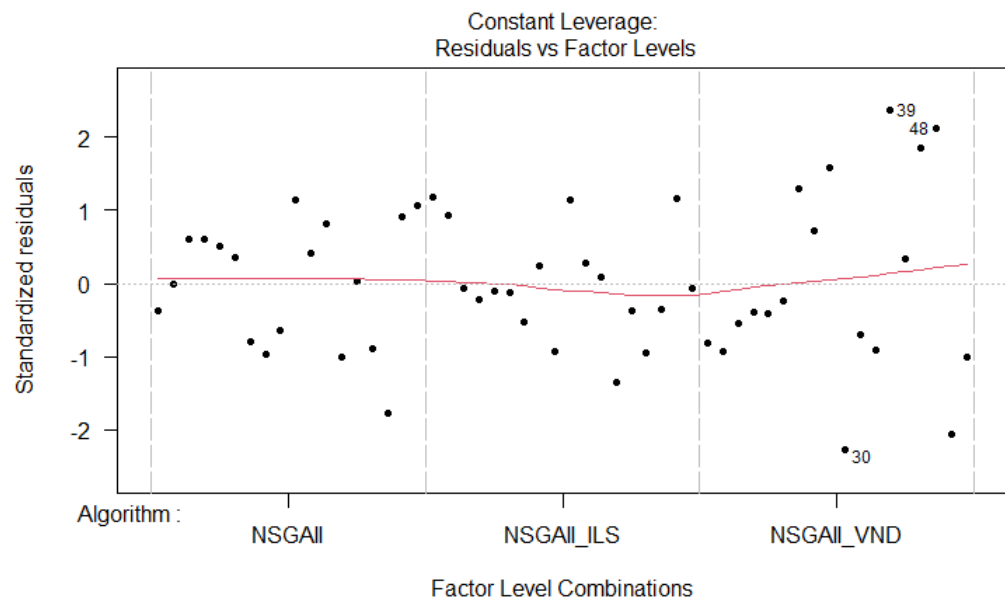


FIGURA 31 – Resíduos por cada algoritmo com blocagem para o Delta.

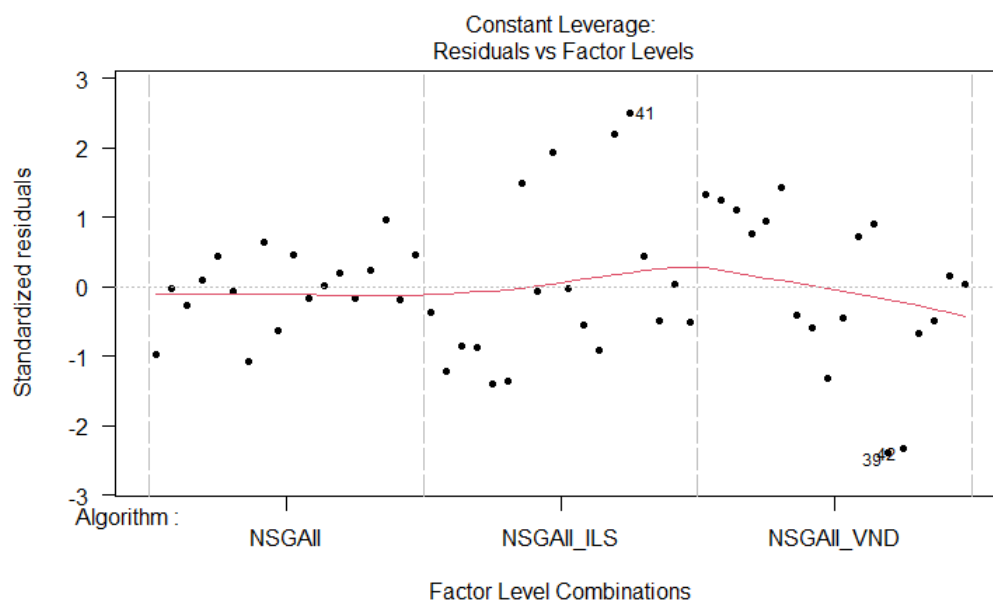


FIGURA 32 – Resíduos por cada algoritmo com bloqueio para o RNI.

As premissas da bloqueio de independência entre blocos e intra-bloco são garantidas pelo projeto experimental. As execuções dos algoritmos são independentes, de forma que as amostras geradas por cada algoritmo em cada instância são produzidas sem que uma observação interfira em outra. A construção do modelo garante que haja apenas uma replicação por bloco.

Com as premissas validadas, o teste de Tukey foi usado para avaliar se há diferença significativa entre os algoritmos. A Tabela 2 resume os resultados para cada par de algoritmos em cada métrica.

TABELA 2 – Probabilidade de não existir diferença entre algoritmos no teste de Tukey com a bloqueio

Algoritmo 1	Algoritmo 2	Hipervolume	Delta	RNI
NSGAI_VND	NSGAI	<1e-04	0.260	0.00698
NSGAI_VND	NSGAI_ILS	0.287	0.884	<0.001
NSGAI	NSGAI_ILS	<1e-04	0.506	0.01495

Os resultados de Tukey para o Hipervolume mostram que o NSGAI é inferior ao demais, mas não há diferença entre o algoritmo NSGAI_VND e o NSGAI_ILS, conforme Figura 33. Isso indica que a qualidade das soluções encontradas pelo NSGAI é inferior, mas que o desempenho dos outros dois algoritmos é semelhante.

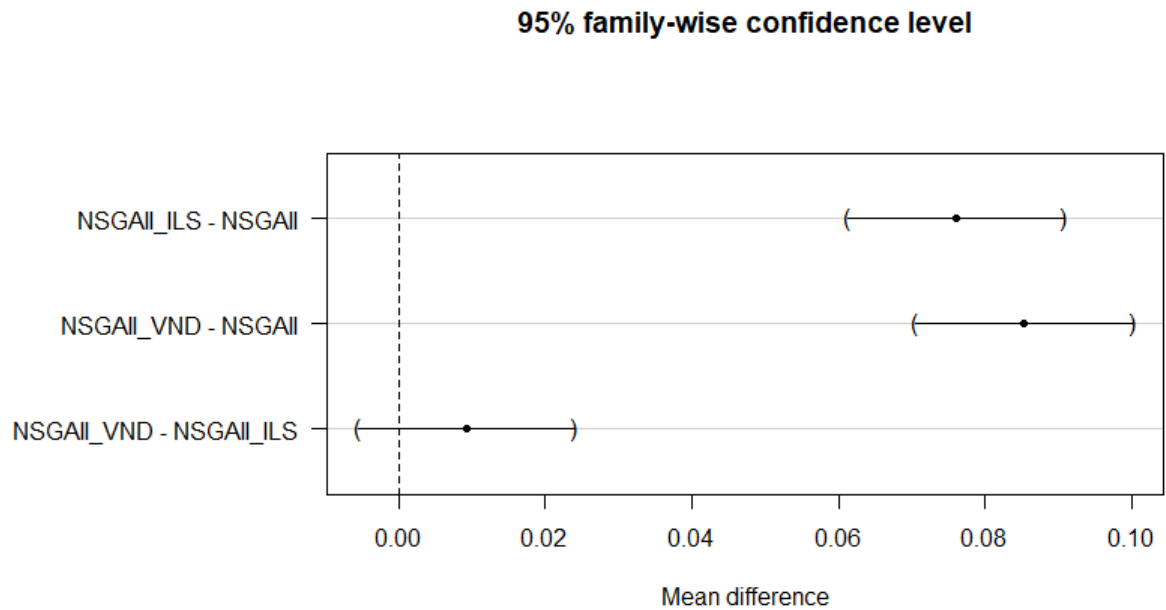


FIGURA 33 – Intervalo de confiança de Tukey para o HV.

Para o indicador de espalhamento, o Delta, os resultados de Tukey não indicaram nenhuma diferença significativa, como mostra a Figura 34. Isso significa que os três algoritmos encontram fronteiras Pareto estimadas com espalhamentos semelhantes.

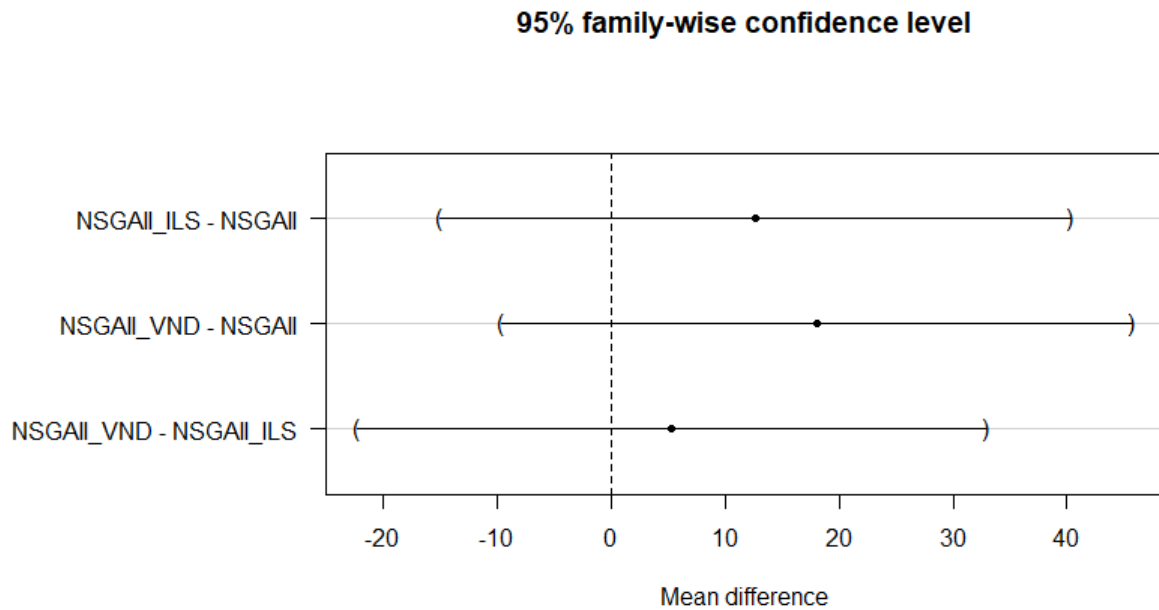


FIGURA 34 – Intervalo de confiança de Tukey para o Delta.

Já para o RNI, a métrica que mede a cardinalidade das fronteiras Pareto estimadas, as comparações múltiplas mostram que há diferenças entre todos os algoritmos. O algoritmo NSGAI_VND se mostrou inferior ao demais, tendo menos soluções na fronteira final. Dentre os outros métodos, o NSGAI_ILS se apresentou superior ao NSGAI, ou seja, o NSGAI_ILS tende a encontrar a maior quantidade de soluções na fronteira final e dar mais alternativas em uma tomada de decisão posterior.

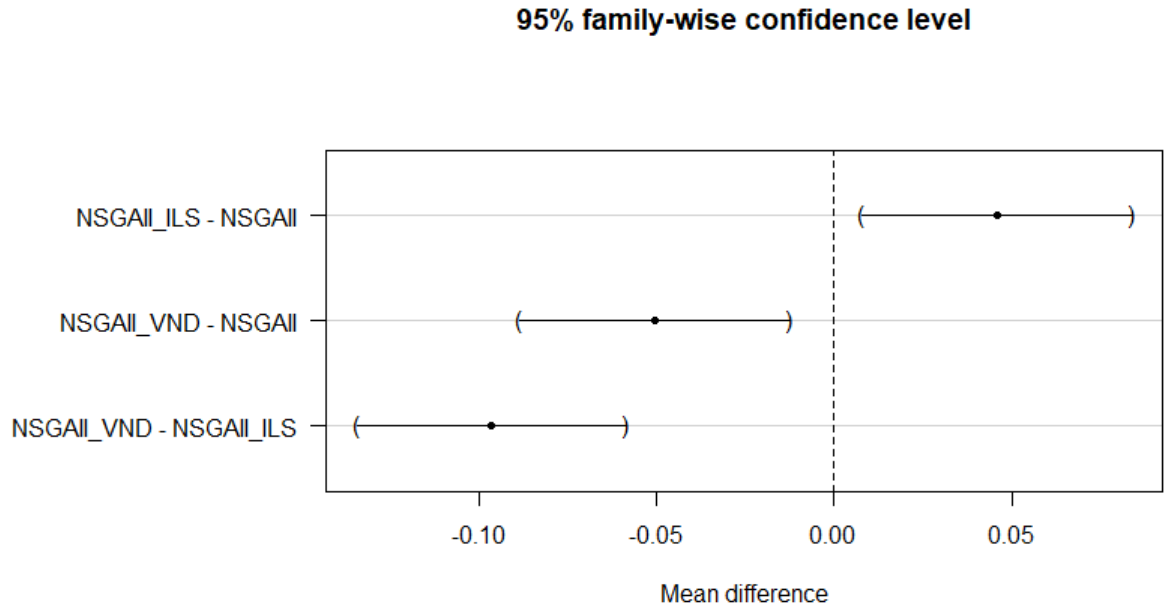


FIGURA 35 – Intervalo de confiança de Tukey para o RNI.

A decisão do melhor algoritmo deve ser feita com base nas comparações de Hipervolume e RNI, pois o Delta não foi capaz de identificar diferenças significativas. O Hipervolume mede convergência e diversidade e o RNI apenas a cardinalidade, então a melhor dentre as três abordagens deve ser aquela com maior HV e o RNI pode ser usado como critério de desempate. Em relação ao Hipervolume, os algoritmos NSGAI_VND e NSGAI_ILS foram superiores ao NSGAI, mas não houve prevalência de nenhum deles. Como o RNI do NSGAI_ILS se mostrou superior ao do NSGAI_VND, o primeiro pode ser considerado o melhor método dentre os três propostos. Resumindo, o algoritmo com melhores resultados foi o NSGAI_ILS, sendo seu grande diferencial a geração de uma boa solução inicial.

6 CONCLUSÃO

Este trabalho analisou e resolveu o Problema de Roteamento de Veículos com Janelas de Atendimento (PRV-JA) com incertezas nos tempos de viagem e atendimento. Em um primeiro momento o problema foi resolvido com tempos determinísticos com o objetivo de minimizar o custo. A inclusão da incerteza ocorreu posteriormente, com o intuito de avaliar seu impacto em uma solução obtida deterministicamente e quantificar quantos clientes são prejudicados por atrasos. Em um segundo momento o problema foi resolvido já considerando as incertezas no processo de busca. Foi proposto um modelo multiobjetivo para minimizar o custo total e a incidência dos atrasos.

O modelo determinístico foi resolvido usando as metaheurísticas ILS e VND, devido à alta complexidade do problema. Uma base de dados bem conhecida foi usada nos testes e os resultados obtidos em cada problema ficaram próximos das melhores soluções conhecidas, o que indica a adequação do método de solução escolhido.

A incerteza nas soluções finais foi inserida ao substituir os tempos de viagem e de atendimento de cada problema por FDPs ajustadas para representar o tempo determinístico como valor médio. Simulações de Monte Carlo foram executadas para avaliar como a inserção de incerteza prejudicaria as soluções. As simulações mostraram que o impacto da incerteza varia de acordo com as características das instâncias. Problemas com clientes agrupados ou com longos períodos de atendimento se mostraram menos propensos a falhas. Os problemas que combinavam essas duas características apresentaram, em média, falhas em menos de 1% das simulações. Entretanto, os testes foram realizados com um valor fixo de CV e uma distribuição de probabilidade para os tempos de viagem e outra para os tempos de atendimento. Diferentes resultados podem ser obtidos caso esses parâmetros mudem.

As instâncias de teste e as distribuições de probabilidade foram mantidas para a segunda parte do trabalho, em que a incerteza já é considerada no processo de busca. Por se tratar de um modelo multiobjetivo, o algoritmo evolutivo NSGA-II foi usado. Foram propostas duas abordagens além do NSGA-II puro, uma que gera uma solução guiada na população inicial usando a combinação do ILS com VND usada na busca determinística (NSGAIILS) e outra que realiza uma busca local somente com um VND em todos os indivíduos gerados ao fim de cada geração (NSGAIIVND). As fronteiras Pareto estimadas de cada algoritmo em cada problema mostraram que cada método levou a soluções com diferentes características. A análise visual dos resultados mostrou que as buscas locais realizadas pelo NSGAIILS e pelo NSGAIIVND levaram a soluções com menores distâncias, porém vários dos problemas concentraram as soluções em regiões específicas do espaço de soluções. Já o NSGA-II puro geralmente

não encontrava soluções com baixa distância, mas era capaz de encontrar soluções melhor distribuídas.

Três métricas de desempenho foram usadas para comparar as abordagens de forma mais metódica: o Hipervolume, que mede convergência e diversidade; o Delta, que mede diversidade; e o RNI, que mede cardinalidade. Uma blocagem foi realizada nos resultados de cada métrica para isolar o efeito de diferentes instâncias antes de comparar os métodos propostos. Os testes do Hipervolume e do RNI apresentaram diferenças significativas entre os algoritmos, mas o Delta não. O NSGA-II puro teve os piores resultados em relação ao Hipervolume, enquanto os outros dois se mostraram semelhantes. Assim, o melhor algoritmo pode ser escolhido dentre aqueles de melhor desempenho que fornece mais alternativas para uma tomada de decisão posterior. Esse critério secundário pode ser medido pelo RNI, e por isso entende-se que essa foi a abordagem mais adequada para a solução do problema.

6.1 PROPOSTA DE TRABALHOS FUTUROS

A primeira parte do trabalho identificou que a inclusão de incerteza afeta a qualidade da solução do PRV-JA. Esse problema foi resolvido na segunda parte usando *sim-heuristics* em um modelo multiobjetivo, um método abrangente que permite considerar qualquer distribuição de probabilidade. Por outro lado, esse modelo tem alto custo computacional e demora até encontrar um bom conjunto de soluções, mesmo em instâncias pequenas. Outras abordagens que estimem o risco sem a necessidade da simulação podem ser exploradas para encontrar soluções em menos tempo.

Todos os experimentos foram realizados considerando uma única distribuição de probabilidade para os tempos de viagem e outra para os tempos de serviço, além de um Coeficiente de Variação fixo em todos os casos. Não há garantia que os resultados obtidos se mantenham caso os tempos incertos sigam outros padrões de distribuição. Dessa forma, seria interessante realizar mais experimentos que considerem outras formas de variar os tempos de viagem e atendimento. Isso pode ser feito explorando a heterogeneidade da incerteza, com Coeficientes de Variação maiores, considerando que cada trecho e cliente possui um CV diferente ou até mesmo que sejam modelados distribuições de probabilidade distintas. Outra alternativa interessante é realizar a modelagem para dados reais de tráfego.

A segunda parte do estudo mostrou que o NSGA-II com ILS apresentou os melhores resultados para o modelo multiobjetivo, apesar de não ter mostrado grandes avanços no HV durante a execução. Assim, estudos futuros podem avaliar se a redução do tempo de execução ou do limite de gerações impactam no seu resultado, e se ele ainda se mantém como melhor algoritmo. A busca local também pode ser aplicada de outras formas, que não sejam executadas apenas na criação da população inicial ou

em todos os indivíduos em todas as gerações. Abordagens intermediárias, que aplicam o VND em apenas uma parcela da população ou apenas em algumas gerações, podem apresentar boa convergência com menor custo computacional e maior diversidade.

O trabalho foi desenvolvido considerando apenas incertezas nos tempos de viagem e atendimento. Entretanto, existem outras variações de Problemas de Roteamento de Veículos Estocásticos. Os mesmos princípios que combinam otimização e simulação também podem ser aplicados a outras variantes do problema, como no caso de demandas estocásticas.

REFERÊNCIAS

- AMARAN, Satyajith et al. Simulation optimization: a review of algorithms and applications. **Annals of Operations Research**, Springer Science e Business Media LLC, v. 240, n. 1, p. 351–380, set. 2015. DOI: 10.1007/s10479-015-2019-x. Disponível em: <<https://doi.org/10.1007/s10479-015-2019-x>>. Citado 1 vez na página 14.
- ANDERLUH, Alexandra et al. Impact of travel time uncertainties on the solution cost of a two-echelon vehicle routing problem with synchronization. **Flexible Services and Manufacturing Journal**, Springer Science e Business Media LLC, v. 32, n. 4, p. 806–828, abr. 2019. DOI: 10.1007/s10696-019-09351-w. Disponível em: <<https://doi.org/10.1007/s10696-019-09351-w>>. Citado 1 vez na página 9.
- ASSIS, Luciana P. et al. Multiobjective vehicle routing problem with fixed delivery and optional collections. **Optimization Letters**, Springer Science e Business Media LLC, v. 7, n. 7, p. 1419–1431, set. 2012. DOI: 10.1007/s11590-012-0551-z. Disponível em: <<https://doi.org/10.1007/s11590-012-0551-z>>. Citado 2 vezes nas páginas 6, 10.
- BARADARAN, Vahid; SHAFAEI, Amir; HOSSEINIAN, Amir Hossein. Stochastic vehicle routing problem with heterogeneous vehicles and multiple prioritized time windows: Mathematical modeling and solution approach. **Computers & Industrial Engineering**, Elsevier BV, v. 131, p. 187–199, mai. 2019. DOI: 10.1016/j.cie.2019.03.047. Disponível em: <<https://doi.org/10.1016/j.cie.2019.03.047>>. Citado 1 vez na página 10.
- BERALDI, P. et al. The risk-averse traveling repairman problem with profits. **Soft Computing**, Springer Science e Business Media LLC, v. 23, n. 9, p. 2979–2993, nov. 2018. DOI: 10.1007/s00500-018-3660-5. Disponível em: <<https://doi.org/10.1007/s00500-018-3660-5>>. Citado 1 vez na página 9.
- BERALDI, Patrizia et al. The mixed capacitated general routing problem under uncertainty. **European Journal of Operational Research**, Elsevier BV, v. 240, n. 2, p. 382–392, jan. 2015. DOI: 10.1016/j.ejor.2014.07.023. Disponível em: <<https://doi.org/10.1016/j.ejor.2014.07.023>>. Citado 2 vezes nas páginas 6, 7.
- BILLINTON, Roy; LI, Wenyuan. **Reliability Assessment of Electric Power Systems Using Monte Carlo Methods**. [S.l.]: Springer US, 1994. (Physics of Solids and Liquids). ISBN 9780306447815. Disponível em: <<https://books.google.com.br/books?id=YN1JRn9-02oC>>. Citado 2 vezes nas páginas 13, 32.

BRAEKERS, Kris; RAMAEKERS, Katrien; NIEUWENHUYSE, Inneke Van. The vehicle routing problem: State of the art classification and review. **Computers & Industrial Engineering**, Elsevier BV, v. 99, p. 300–313, set. 2016. DOI:

10.1016/j.cie.2015.12.007. Disponível em:

<<https://doi.org/10.1016/j.cie.2015.12.007>>. Citado 4 vezes nas páginas 2, 5, 6, 8.

CHEN, Huey-Kuo; HSUEH, Che-Fu; CHANG, Mei-Shiang. Production scheduling and vehicle routing with time windows for perishable food products. **Computers & Operations Research**, Elsevier BV, v. 36, n. 7, p. 2311–2319, jul. 2009. DOI:

10.1016/j.cor.2008.09.010. Disponível em:

<<https://doi.org/10.1016/j.cor.2008.09.010>>. Citado 1 vez na página 5.

COLLETTE, Y.; SIARRY, P. **Multiobjective Optimization: Principles and Case Studies**. [S.l.]: Springer Berlin Heidelberg, 2004. (Decision Engineering). ISBN 9783540401827. Disponível em:

<<https://books.google.com.br/books?id=XNYF4h1toF0C>>. Citado 1 vez na página 13.

DANTZIG, G. B.; RAMSER, J. H. The Truck Dispatching Problem. **Management Science**, Institute for Operations Research e the Management Sciences (INFORMS), v. 6, n. 1, p. 80–91, out. 1959. DOI: 10.1287/mnsc.6.1.80. Disponível em:

<<https://doi.org/10.1287/mnsc.6.1.80>>. Citado 2 vezes nas páginas 1, 5.

DEB, K. et al. A fast and elitist multiobjective genetic algorithm: NSGA-II. **IEEE Transactions on Evolutionary Computation**, Institute of Electrical e Electronics Engineers (IEEE), v. 6, n. 2, p. 182–197, abr. 2002. DOI: 10.1109/4235.996017.

Disponível em: <<https://doi.org/10.1109/4235.996017>>. Citado 1 vezes nas páginas 16, 18, 20.

DEB, Kalyanmoy et al. A Fast Elitist Non-dominated Sorting Genetic Algorithm for Multi-objective Optimization: NSGA-II. In: **PARALLEL Problem Solving from Nature PPSN VI**. [S.l.]: Springer Berlin Heidelberg, 2000. P. 849–858. DOI:

10.1007/3-540-45356-3_83. Disponível em:

<https://doi.org/10.1007/3-540-45356-3_83>. Citado 1 vez na página 21.

EKSIOGLU, Burak; VURAL, Arif Volkan; REISMAN, Arnold. The vehicle routing problem: A taxonomic review. **Computers & Industrial Engineering**, Elsevier BV, v. 57, n. 4, p. 1472–1483, nov. 2009. DOI: 10.1016/j.cie.2009.05.009. Disponível em: <<https://doi.org/10.1016/j.cie.2009.05.009>>. Citado 1 vez na página 5.

GARCIA, Vinicius Jacques et al. A computational intelligence approach to improve the efficiency of repair services in the smart grid context. **Computers & Electrical Engineering**, Elsevier BV, v. 70, p. 37–52, ago. 2018. DOI:

- 10.1016/j.compeleceng.2018.05.016. Disponível em:
<<https://doi.org/10.1016/j.compeleceng.2018.05.016>>. Citado 1 vez na página 5.
- GAUVIN, Charles; DESAULNIERS, Guy; GENDREAU, Michel. A branch-cut-and-price algorithm for the vehicle routing problem with stochastic demands. **Computers & Operations Research**, Elsevier BV, v. 50, p. 141–153, out. 2014. DOI:
10.1016/j.cor.2014.03.028. Disponível em:
<<https://doi.org/10.1016/j.cor.2014.03.028>>. Citado 2 vezes nas páginas 7, 8.
- GEE, Sen Bong et al. Decomposition-based multi-objective evolutionary algorithm for vehicle routing problem with stochastic demands. **Soft Computing**, Springer Science e Business Media LLC, v. 20, n. 9, p. 3443–3453, ago. 2015. DOI:
10.1007/s00500-015-1830-2. Disponível em:
<<https://doi.org/10.1007/s00500-015-1830-2>>. Citado 1 vez na página 10.
- GENDREAU, Michel; JABALI, Ola; REI, Walter. 50th Anniversary Invited Article—Future Research Directions in Stochastic Vehicle Routing. **Transportation Science**, Institute for Operations Research e the Management Sciences (INFORMS), v. 50, n. 4, p. 1163–1173, jun. 2016. DOI: 10.1287/trsc.2016.0709. Disponível em:
<<https://doi.org/10.1287/trsc.2016.0709>>. Citado 1 vez na página 7.
- GENDREAU, Michel; LAPORTE, Gilbert; SÉGUIN, René. Stochastic vehicle routing. **European Journal of Operational Research**, Elsevier BV, v. 88, n. 1, p. 3–12, jan. 1996. DOI: 10.1016/0377-2217(95)00050-x. Disponível em:
<[https://doi.org/10.1016/0377-2217\(95\)00050-x](https://doi.org/10.1016/0377-2217(95)00050-x)>. Citado 3 vezes nas páginas 2, 7, 8.
- GENDREAU, Michel; POTVIN, Jean-Yves (Ed.). **Handbook of Metaheuristics**. [S.l.]: Springer International Publishing, 2019. DOI: 10.1007/978-3-319-91086-4. Disponível em: <<https://doi.org/10.1007/978-3-319-91086-4>>. Citado 2 vezes nas páginas 15, 16.
- GÓMEZ, Andrés et al. On Modeling Stochastic Travel and Service Times in Vehicle Routing. **Transportation Science**, Institute for Operations Research e the Management Sciences (INFORMS), v. 50, n. 2, p. 627–641, mai. 2016. DOI:
10.1287/trsc.2015.0601. Disponível em:
<<https://doi.org/10.1287/trsc.2015.0601>>. Citado 2 vezes nas páginas 7, 9.
- GOODSON, Justin C. A priori policy evaluation and cyclic-order-based simulated annealing for the multi-compartment vehicle routing problem with stochastic demands. **European Journal of Operational Research**, Elsevier BV, v. 241, n. 2, p. 361–369, mar. 2015. DOI: 10.1016/j.ejor.2014.09.031. Disponível em:
<<https://doi.org/10.1016/j.ejor.2014.09.031>>. Citado 2 vezes nas páginas 7, 8.

GUIMARANS, Daniel et al. A simheuristic approach for the two-dimensional vehicle routing problem with stochastic travel times. **Simulation Modelling Practice and Theory**, Elsevier BV, v. 89, p. 1–14, dez. 2018. DOI: 10.1016/j.simpat.2018.09.004. Disponível em: <<https://doi.org/10.1016/j.simpat.2018.09.004>>. Citado 1 vez na página 9.

GUTIERREZ, Andres et al. A hybrid metaheuristic algorithm for the vehicle routing problem with stochastic demands. **Computers & Operations Research**, Elsevier BV, v. 99, p. 135–147, nov. 2018. DOI: 10.1016/j.cor.2018.06.012. Disponível em: <<https://doi.org/10.1016/j.cor.2018.06.012>>. Citado 6 vezes nas páginas 3, 6–8, 31, 32.

HARRISON, Robert L.; GRANJA, Carlos; LEROY, Claude. Introduction to Monte Carlo Simulation. In: DOI: 10.1063/1.3295638. Disponível em: <<https://doi.org/10.1063/1.3295638>>. Citado 1 vez na página 14.

IRAWAN, Chandra Ade et al. Simulation-based optimisation for stochastic maintenance routing in an offshore wind farm. **European Journal of Operational Research**, Elsevier BV, v. 289, n. 3, p. 912–926, mar. 2021. DOI: 10.1016/j.ejor.2019.08.032. Disponível em: <<https://doi.org/10.1016/j.ejor.2019.08.032>>. Citado 1 vez na página 5.

JIANG, Siwei et al. Consistencies and Contradictions of Performance Metrics in Multiobjective Optimization. **IEEE Transactions on Cybernetics**, Institute of Electrical e Electronics Engineers (IEEE), v. 44, n. 12, p. 2391–2404, dez. 2014. DOI: 10.1109/tcyb.2014.2307319. Disponível em: <<https://doi.org/10.1109/tcyb.2014.2307319>>. Citado 1 vez na página 21.

JUAN, A. et al. Using safety stocks and simulation to solve the vehicle routing problem with stochastic demands. **Transportation Research Part C: Emerging Technologies**, Elsevier BV, v. 19, n. 5, p. 751–765, ago. 2011. DOI: 10.1016/j.trc.2010.09.007. Disponível em: <<https://doi.org/10.1016/j.trc.2010.09.007>>. Citado 1 vez na página 10.

JUAN, Angel A. et al. A review of simheuristics: Extending metaheuristics to deal with stochastic combinatorial optimization problems. **Operations Research Perspectives**, Elsevier BV, v. 2, p. 62–72, dez. 2015. DOI: 10.1016/j.orp.2015.03.001. Disponível em: <<https://doi.org/10.1016/j.orp.2015.03.001>>. Citado 2 vezes nas páginas 9, 14.

LAHYANI, Rahma; KHEMAKHEM, Mahdi; SEMET, Frédéric. **Rich vehicle routing problems: From a taxonomy to a definition**. v. 241. [S.l.]: Elsevier BV, fev. 2015. P. 1–14. DOI: 10.1016/j.ejor.2014.07.048. Disponível em: <<https://doi.org/10.1016/j.ejor.2014.07.048>>. Citado 2 vezes nas páginas 2, 6.

LAPORTE, Gilbert; LOUVEAUX, François; MERCURE, Hélène. The Vehicle Routing Problem with Stochastic Travel Times. **Transportation Science**, Institute for Operations Research e the Management Sciences (INFORMS), v. 26, n. 3, p. 161–170, ago. 1992. DOI: 10.1287/trsc.26.3.161. Disponível em:

<<https://doi.org/10.1287/trsc.26.3.161>>. Citado 1 vez na página 8.

LEI, Hongtao; LAPORTE, Gilbert; GUO, Bo. A generalized variable neighborhood search heuristic for the capacitated vehicle routing problem with stochastic service times. **TOP**, Springer Science e Business Media LLC, v. 20, n. 1, p. 99–118, abr. 2011. DOI: 10.1007/s11750-011-0188-6. Disponível em:

<<https://doi.org/10.1007/s11750-011-0188-6>>. Citado 1 vez na página 8.

MAIA, Pedro M; BESSANI, Michel. Análise do Efeito de Incertezas de Tempo Aplicadas ao Problema de Roteamento de Veículos com Janelas de Atendimento. In: ANAIS do LII Simpósio Brasileiro de Pesquisa Operacional. [S.l.: s.n.], 2020. Disponível em:

<<https://proceedings.science/sbpo-2020/papers/analise-do-efeito-de-incertezas-de-tempo-aplicadas-ao-problema-de-roteamento-de-veiculos-com-janelas-de-atendimento>>. Citado 1 vez na página 4.

MALANDRAKI, Chryssi; DASKIN, Mark S. Time Dependent Vehicle Routing Problems: Formulations, Properties and Heuristic Algorithms. **Transportation Science**, Institute for Operations Research e the Management Sciences (INFORMS), v. 26, n. 3, p. 185–200, ago. 1992. DOI: 10.1287/trsc.26.3.185. Disponível em:

<<https://doi.org/10.1287/trsc.26.3.185>>. Citado 1 vez na página 6.

MENDOZA, Jorge E.; ROUSSEAU, Louis-Martin; VILLEGAS, Juan G. A hybrid metaheuristic for the vehicle routing problem with stochastic demand and duration constraints. **Journal of Heuristics**, Springer Science e Business Media LLC, v. 22, n. 4, p. 539–566, jan. 2015. DOI: 10.1007/s10732-015-9281-6. Disponível em:

<<https://doi.org/10.1007/s10732-015-9281-6>>. Citado 1 vez na página 8.

MIRANDA, Douglas M.; BRANKE, Juergen; CONCEIÇÃO, Samuel V. Algorithms for the multi-objective vehicle routing problem with hard time windows and stochastic travel time and service time. **Applied Soft Computing**, Elsevier BV, v. 70, p. 66–79, set. 2018. DOI: 10.1016/j.asoc.2018.05.026. Disponível em:

<<https://doi.org/10.1016/j.asoc.2018.05.026>>. Citado 1 vez na página 10.

MIRANDA, Douglas M.; CONCEIÇÃO, Samuel V. Um problema de roteamento dinâmico, com tempos de viagem estocásticos, múltiplos veículos e janela de tempo. In: XLII Simpósio Brasileiro de Pesquisa Operacional. Bento Gonçalves - RS: Anais do XLII Simpósio Brasileiro de Pesquisa Operacional, 2010. Citado 1 vez na página 8.

MIRANDA, Douglas Moura; CONCEIÇÃO, Samuel Vieira. The vehicle routing problem with hard time windows and stochastic travel and service time. **Expert Systems with Applications**, Elsevier BV, v. 64, p. 104–116, dez. 2016. DOI:

- 10.1016/j.eswa.2016.07.022. Disponível em:
<<https://doi.org/10.1016/j.eswa.2016.07.022>>. Citado 3 vezes nas páginas 7–9.
- MONTGOMERY, D.C. **Design and Analysis of Experiments**. [S.l.]: John Wiley & Sons Incorporated, 2013. ISBN 9781621982272. Disponível em:
<<https://books.google.com.br/books?id=IGyyswEACAAJ>>. Citado 1 vez na página 36.
- O'CONNOR, Patrick D. T.; KLEYNER, Andre. **Practical Reliability Engineering**. [S.l.]: John Wiley & Sons, Ltd, dez. 2011. DOI: 10.1002/9781119961260. Disponível em:
<<https://doi.org/10.1002/9781119961260>>. Citado 0 vez na página 14.
- OMBUKI, Beatrice; ROSS, Brian J.; HANSHAR, Franklin. Multi-Objective Genetic Algorithms for Vehicle Routing Problem with Time Windows. **Applied Intelligence**, Springer Science e Business Media LLC, v. 24, n. 1, p. 17–30, fev. 2006. DOI: 10.1007/s10489-006-6926-z. Disponível em:
<<https://doi.org/10.1007/s10489-006-6926-z>>. Citado 1 vezes nas páginas 33, 34.
- OYOLA, Jorge; ARNTZEN, Halvard; WOODRUFF, David L. The stochastic vehicle routing problem, a literature review, Part II: solution methods. **EURO Journal on Transportation and Logistics**, Elsevier BV, v. 6, n. 4, p. 349–388, nov. 2016. DOI: 10.1007/s13676-016-0099-7. Disponível em:
<<https://doi.org/10.1007/s13676-016-0099-7>>. Citado 1 vez na página 8.
- OYOLA, Jorge; WOODRUFF, David L.; ARNTZEN, Halvard. The stochastic vehicle routing problem, a literature review, part I: models. **EURO Journal on Transportation and Logistics**, Elsevier BV, v. 7, n. 3, p. 193–221, out. 2016. DOI: 10.1007/s13676-016-0100-5. Disponível em:
<<https://doi.org/10.1007/s13676-016-0100-5>>. Citado 2 vezes nas páginas 7, 8.
- PANG, Nansheng et al. Emergency rush repair task scheduling of distribution networks in large-scale blackouts. **International Journal of Electrical Power & Energy Systems**, Elsevier BV, v. 82, p. 373–381, nov. 2016. DOI: 10.1016/j.ijepes.2016.03.014. Disponível em:
<<https://doi.org/10.1016/j.ijepes.2016.03.014>>. Citado 1 vez na página 5.
- RIQUELME, Nery; LUCKEN, Christian Von; BARAN, Benjamin. Performance metrics in multi-objective optimization. In: 2015 Latin American Computing Conference (CLEI). [S.l.]: IEEE, out. 2015. DOI: 10.1109/clei.2015.7360024. Disponível em:
<<https://doi.org/10.1109/clei.2015.7360024>>. Citado 1 vez na página 21.
- SALEHIPOUR, Amir et al. Efficient GRASPVND and GRASPVNS metaheuristics for the traveling repairman problem. **4OR**, Springer Science e Business Media LLC, v. 9,

- n. 2, p. 189–209, jan. 2011. DOI: 10.1007/s10288-011-0153-0. Disponível em: <<https://doi.org/10.1007/s10288-011-0153-0>>. Citado 2 vezes nas páginas 3, 5.
- SOLOMON, Marius M. Algorithms for the Vehicle Routing and Scheduling Problems with Time Window Constraints. **Operations Research**, INFORMS, v. 35, n. 2, p. 254–265, 1987. ISSN 0030364X, 15265463. Disponível em: <<http://www.jstor.org/stable/170697>>. Citado 3 vezes nas páginas 27, 28, 32.
- SUN, W.; YUAN, Y.X. **Optimization Theory and Methods: Nonlinear Programming**. [S.l.]: Springer US, 2006. (Springer Optimization and Its Applications). ISBN 9780387249766. Disponível em: <<https://books.google.com.br/books?id=o0BYHLhhPJMC>>. Citado 1 vez na página 12.
- TAN, K.C.; CHEONG, C.Y.; GOH, C.K. Solving multiobjective vehicle routing problem with stochastic demand via evolutionary computation. **European Journal of Operational Research**, Elsevier BV, v. 177, n. 2, p. 813–839, mar. 2007. DOI: 10.1016/j.ejor.2005.12.029. Disponível em: <<https://doi.org/10.1016/j.ejor.2005.12.029>>. Citado 4 vezes nas páginas 1, 7, 8, 10.
- TAN, K.C.; LEE, T.H.; KHOR, E.F. **Artificial Intelligence Review**, Springer Science e Business Media LLC, v. 17, n. 4, p. 251–290, 2002. DOI: 10.1023/a:1015516501242. Disponível em: <<https://doi.org/10.1023/a:1015516501242>>. Citado 1 vez na página 21.
- TAŞ, D. et al. Vehicle routing with soft time windows and stochastic travel times: A column generation and branch-and-price solution approach. **European Journal of Operational Research**, Elsevier BV, v. 236, n. 3, p. 789–799, ago. 2014. DOI: 10.1016/j.ejor.2013.05.024. Disponível em: <<https://doi.org/10.1016/j.ejor.2013.05.024>>. Citado 1 vez na página 8.
- TOTH, P.; VIGO, D. **Vehicle Routing: Problems, Methods, and Applications, Second Edition**. [S.l.]: SIAM, 2014. (MOS-SIAM Series on Optimization). ISBN 9781611973594. Disponível em: <<https://books.google.com.br/books?id=AoTTBQAAQBAJ>>. Citado 2 vezes nas páginas 6, 24.
- VRPTW BENCHMARK PROBLEMS. [S.l.: s.n.], mar. 2005. <http://web.cba.neu.edu/~msolomon/problems.htm>. Acessado em: 2019-10-13. Citado 1 vez na página 38.
- ZHANG, Junlong; LAM, William H. K.; CHEN, Bi Yu. A Stochastic Vehicle Routing Problem with Travel Time Uncertainty: Trade-Off Between Cost and Customer Service. **Networks and Spatial Economics**, Springer Science e Business Media LLC, v. 13,

n. 4, p. 471–496, mai. 2013. DOI: 10.1007/s11067-013-9190-x. Disponível em: <<https://doi.org/10.1007/s11067-013-9190-x>>. Citado 1 vez na página 8.

ZHANG, Junlong; LAM, William H.K.; CHEN, Bi Yu. On-time delivery probabilistic models for the vehicle routing problem with stochastic demands and time windows. **European Journal of Operational Research**, Elsevier BV, v. 249, n. 1, p. 144–154, fev. 2016. DOI: 10.1016/j.ejor.2015.08.050. Disponível em: <<https://doi.org/10.1016/j.ejor.2015.08.050>>. Citado 2 vez na página 7.

ZITZLER, E.; THIELE, L. Multiobjective Optimization Using Evolutionary Algorithms - A Comparative Case Study. In: PPSN. [S.l.: s.n.], 1998. Citado 1 vez na página 22.

APÊNDICES

APÊNDICE A – TABELAS DE RESULTADOS

TABELA 3 – Resultados da otimização na análise do efeito da incerteza

Problema	Menor distância(d)	Veículos Usados	\bar{d}	$var(d)$	$min(d)$	Δd	\bar{v}	\overline{tmv}
R101	1044.0	12	1075.2	129.1	1049.5	2.98%	12.9	2313.1
R102	909.0	11	926.7	47.4	913.6	1.94%	11.3	2047.7
R103	772.9	9	793.4	30.4	783.8	2.65%	8.9	1678.4
R104	625.4	6	650.8	49.8	631.6	4.07%	6.2	1266.5
R105	899.3	9	939.3	53.4	928.5	4.44%	10.5	1874.3
R106	793.0	5	820.0	27.4	811.6	3.40%	9.2	1647.2
R107	711.1	7	744.0	71.3	730.9	4.62%	7.6	1411.1
R108	617.7	6	637.8	60.5	623.5	3.25%	6.0	1205.0
R109	786.8	8	819.8	61.5	810.9	4.20%	8.6	1553.3
R110	697.0	7	725.7	40.1	717.1	4.12%	7.7	1473.5
R111	707.2	7	733.8	56.0	722.2	3.76%	7.9	1441.0
R112	630.2	6	655.1	36.5	646.6	3.95%	6.4	1270.9
R201	791.9	6	808.2	31.8	794.3	2.06%	5.7	4114.2
R202	698.5	5	715.8	23.0	700.7	2.47%	4.5	3203.1
R203	605.3	5	618.2	24.6	607.6	2.13%	4.1	2926.8
R204	506.4	2	513.0	3.3	511.0	1.31%	2.1	1569.3
R205	690.1	4	704.2	38.4	692.4	2.05%	4.3	3127.5
R206	632.4	4	652.3	17.3	644.7	3.15%	3.5	2508.1
R209	600.6	4	618.6	61.5	608.1	3.00%	3.9	2724.3
R210	645.6	4	661.7	8.9	657.5	2.49%	3.5	2679.9
R211	535.5	3	557.4	97.8	540.1	4.09%	3.1	1967.0
C101	362.4	5	367.6	67.7	363.2	1.42%	5.0	5078.0
C102	361.4	5	366.2	56.6	362.2	1.32%	5.0	5187.8
C103	361.4	5	385.1	138.7	362.2	6.54%	5.0	5326.3
C104	358.0	5	392.5	54.4	381.1	9.64%	5.0	5432.6
C105	362.4	5	368.8	68.8	363.2	1.76%	5.2	5203.0
C106	362.4	5	367.1	39.0	363.2	1.29%	5.1	5048.0
C107	362.4	5	364.9	5.9	363.2	0.70%	5.0	4965.5
C108	362.4	5	366.9	85.1	363.2	1.23%	5.0	4893.8
C109	362.4	5	378.0	147.7	363.2	4.30%	5.1	4978.6
C201	360.2	3	361.8	0.0	361.8	0.44%	3.0	7498.7
C202	360.2	3	361.8	0.0	361.8	0.44%	3.0	7780.4
C203	359.8	3	363.7	6.9	361.4	1.07%	3.0	8005.4
C204	350.1	2	355.6	5.4	352.3	1.56%	2.0	5938.2
C205	359.8	3	361.4	0.0	361.4	0.45%	3.0	7430.1
C206	359.8	3	361.4	0.0	361.4	0.45%	3.0	7346.1
C207	359.6	3	363.0	12.0	361.2	0.95%	3.0	7546.5
C208	350.5	2	352.1	0.0	352.1	0.46%	2.0	6129.2
RC101	944.0	8	975.1	22.8	962.1	3.29%	9.5	1849.3
RC102	822.5	7	851.7	198.4	828.5	3.54%	7.9	1568.7
RC103	710.9	6	737.1	530.9	713.7	3.69%	6.3	1352.2
RC104	545.8	5	551.8	17.2	546.5	1.09%	5.0	1144.4
RC105	855.3	8	878.7	309.7	860.4	2.74%	8.3	1648.4
RC106	723.2	6	816.1	510.0	768.0	12.85%	7.4	1458.1
RC107	642.7	6	680.1	399.9	653.4	5.82%	6.2	1298.7
RC108	598.1	6	611.1	73.2	599.2	2.18%	6.0	1206.6
RC201	684.8	5	713.2	234.4	686.3	4.15%	4.8	3623.1
RC202	613.6	5	651.2	350.7	615.0	6.13%	4.3	3300.1
RC203	555.3	4	622.6	502.4	580.1	12.12%	3.8	2898.6
RC204	444.2	3	465.7	432.7	445.0	4.84%	3.0	2347.2
RC205	630.2	5	666.6	454.2	633.0	5.77%	4.9	3541.6
RC206	610.0	5	657.5	326.2	625.1	7.79%	3.9	2765.3
RC207	558.6	4	590.9	381.4	565.2	5.79%	4.0	2691.1

TABELA 4 – Resultados da simulação na análise do efeito da incerteza

Problema	Otimização		Simulação		
	tmv	$tmvs$	Δtmv	$c\bar{v}$	pf
R101	2313.1	2315.4	0.10%	2.6	0.907
R102	2047.7	2049.8	0.10%	2.4	0.898
R103	1678.4	1683.8	0.32%	2.2	0.816
R104	1266.5	1270.6	0.32%	1.9	0.745
R105	1874.3	1877.6	0.17%	1.7	0.845
R106	1647.2	1650.5	0.20%	1.9	0.845
R107	1411.1	1415.0	0.27%	2.1	0.824
R108	1205.0	1207.0	0.16%	2.0	0.777
R109	1553.3	1557.1	0.24%	1.6	0.749
R110	1473.5	1475.4	0.13%	1.9	0.803
R111	1441.0	1445.3	0.30%	2.6	0.907
R112	1270.9	1273.8	0.23%	2.2	0.799
R201	4114.2	4114.3	0.00%	0.3	0.250
R202	3203.1	3203.2	0.00%	0.4	0.386
R203	2926.8	2927.0	0.01%	0.5	0.472
R204	1569.3	1569.3	0.00%	0.4	0.282
R205	3127.5	3127.5	0.00%	0.3	0.318
R206	2508.1	2508.4	0.01%	0.2	0.168
R207	1897.9	1900.0	0.11%	0.0	0.039
R208	1410.0	1409.8	-0.01%	0.0	0.001
R209	2724.3	2724.4	0.00%	0.2	0.201
R210	2679.9	2679.8	0.00%	0.2	0.198
R211	1967.0	1968.1	0.06%	0.5	0.401
C101	5078.0	5084.3	0.12%	0.6	0.265
C102	5187.8	5196.3	0.16%	0.3	0.163
C103	5326.3	5333.3	0.13%	0.2	0.156
C104	5432.6	5436.4	0.07%	0.1	0.056
C105	5203.0	5209.6	0.13%	0.1	0.079
C106	5048.0	5054.0	0.12%	0.7	0.338
C107	4965.5	4972.9	0.15%	0.0	0.002
C108	4893.8	4900.2	0.13%	0.1	0.074
C109	4978.6	4985.5	0.14%	0.0	0.000
C201	7498.7	7507.1	0.11%	0.0	0.002
C202	7780.4	7785.2	0.06%	0.0	0.002
C203	8005.4	8011.9	0.08%	0.0	0.001
C204	5938.2	5947.8	0.16%	0.0	0.000
C205	7430.1	7435.5	0.07%	0.0	0.000
C206	7346.1	7348.7	0.04%	0.0	0.000
C207	7546.5	7548.4	0.03%	0.0	0.000
C208	6129.2	6131.0	0.03%	0.0	0.000
RC101	1849.3	1852.7	0.18%	1.5	0.701
RC102	1568.7	1575.8	0.45%	2.8	0.834
RC103	1352.2	1359.1	0.51%	2.6	0.866
RC104	1144.4	1145.0	0.06%	1.8	0.819
RC105	1648.4	1655.3	0.42%	2.4	0.857
RC106	1458.1	1464.9	0.47%	2.6	0.896
RC107	1298.7	1307.9	0.71%	2.7	0.894
RC108	1206.6	1215.4	0.72%	2.2	0.883
RC201	3623.1	3623.4	0.01%	0.4	0.279
RC202	3300.1	3300.3	0.00%	0.5	0.383
RC203	2898.6	2898.7	0.00%	0.3	0.257
RC204	2347.2	2347.2	0.00%	0.0	0.012
RC205	3541.6	3542.5	0.02%	0.2	0.199
RC206	2765.3	2766.0	0.03%	0.3	0.236
RC207	2691.1	2691.7	0.02%	0.4	0.304
RC208	2246.3	2246.4	0.01%	0.2	0.148

TABELA 5 – Resultados médios dos algoritmos multiobjetivo.

Problema	Algoritmo	Gerações	Min. Violação	Min. Distância	Num. Soluções	ΔHV
R101	NSGAI	44.3	1.19	2033.25	12.3	0.09
R101	NSGAI_ILS	27.1	1.36	1695.54	19.3	0.02
R101	NSGAI_VND	10.2	2.47	1813.11	11.0	0.14
R201	NSGAI	24.3	0.02	1712.59	10.0	0.22
R201	NSGAI_ILS	20.1	0.02	1338.98	11.4	0.01
R201	NSGAI_VND	8.0	0.05	1252.18	5.8	0.34
C104	NSGAI	20.9	0.01	1456.16	5.5	0.08
C104	NSGAI_ILS	16.6	0.01	1078.04	8.9	0.01
C104	NSGAI_VND	6.0	0.01	988.39	7.2	0.17
C202	NSGAI	23.0	0.0	1043.26	3.1	0.21
C202	NSGAI_ILS	24.9	0.0	696.15	3.6	0.0
C202	NSGAI_VND	7.9	0.01	636.28	1.8	0.31
RC103	NSGAI	29.8	0.21	1742.64	17.5	0.06
RC103	NSGAI_ILS	28.3	0.2	1523.27	23.5	0.02
RC103	NSGAI_VND	6.8	1.02	1432.57	10.2	0.1
RC201	NSGAI	23.5	0.03	1999.29	11.5	0.18
RC201	NSGAI_ILS	18.7	0.02	1445.41	11.1	0.0
RC201	NSGAI_VND	7.8	0.03	1380.15	6.3	0.3
R105	NSGAI	26.7	0.17	1839.81	15.6	0.06
R105	NSGAI_ILS	13.9	0.33	1441.47	16.6	0.01
R105	NSGAI_VND	8.7	0.69	1511.04	10.8	0.14
R203	NSGAI	22.3	0.01	1297.08	7.1	0.12
R203	NSGAI_ILS	18.9	0.0	997.47	8.7	0.0
R203	NSGAI_VND	6.1	0.0	944.49	6.2	0.21
C105	NSGAI	21.6	0.0	1668.66	4.0	0.11
C105	NSGAI_ILS	17.6	0.0	1062.59	4.1	0.02
C105	NSGAI_VND	7.7	0.01	951.81	3.8	0.26
C206	NSGAI	21.2	0.0	1070.22	3.5	0.23
C206	NSGAI_ILS	19.3	0.0	732.7	2.4	0.01
C206	NSGAI_VND	7.9	0.0	644.44	1.9	0.35
RC105	NSGAI	36.0	0.58	2107.62	13.8	0.08
RC105	NSGAI_ILS	40.9	0.6	1813.99	21.0	0.03
RC105	NSGAI_VND	8.8	1.76	1692.44	7.3	0.15
RC202	NSGAI	24.5	0.02	1751.79	8.6	0.17
RC202	NSGAI_ILS	19.4	0.01	1298.92	11.3	0.0
RC202	NSGAI_VND	7.1	0.01	1206.67	6.7	0.28
R107	NSGAI	26.2	0.08	1501.98	14.9	0.05
R107	NSGAI_ILS	26.9	0.07	1281.05	21.9	0.01
R107	NSGAI_VND	7.1	0.42	1186.07	11.1	0.13
R206	NSGAI	23.8	0.0	1276.19	6.8	0.18
R206	NSGAI_ILS	18.9	0.0	1010.38	7.4	0.0
R206	NSGAI_VND	6.0	0.0	962.07	5.9	0.24
C108	NSGAI	22.7	0.0	1455.76	4.6	0.1
C108	NSGAI_ILS	19.4	0.0	1033.98	5.8	0.01
C108	NSGAI_VND	7.4	0.0	960.8	4.6	0.21
C208	NSGAI	25.9	0.0	781.89	2.3	0.13
C208	NSGAI_ILS	19.9	0.0	718.26	2.2	0.01
C208	NSGAI_VND	8.65	0.0	624.72	2.5	0.13
RC107	NSGAI	25.4	0.03	1789.93	16.2	0.06
RC107	NSGAI_ILS	25.1	0.02	1488.83	18.9	0.01
RC107	NSGAI_VND	5.8	0.1	1381.59	12.0	0.13
RC205	NSGAI	23.0	0.04	1912.05	9.2	0.19
RC205	NSGAI_ILS	19.8	0.02	1387.14	9.7	0.01
RC205	NSGAI_VND	6.0	0.02	1292.59	5.9	0.29