

GIOVANNI CESAR MEIRA BARBOZA

**ESTUDO DE HEURÍSTICAS PARA O PROBLEMA  
DE ROTEIRIZAÇÃO DE VEÍCULOS COM FROTA  
HETEROGÊNEA FIXA, JANELAS DE TEMPO,  
ENTREGAS FRACIONADAS E LIMITAÇÕES DE  
ACESSO**

São Paulo  
2024



GIOVANNI CESAR MEIRA BARBOZA

**ESTUDO DE HEURÍSTICAS PARA O PROBLEMA  
DE ROTEIRIZAÇÃO DE VEÍCULOS COM FROTA  
HETEROGÊNEA FIXA, JANELAS DE TEMPO,  
ENTREGAS FRACIONADAS E LIMITAÇÕES DE  
ACESSO**

Trabalho apresentada à Escola Politécnica  
da Universidade de São Paulo para obten-  
ção do Diploma de Engenheiro de Produção.

São Paulo  
2024

GIOVANNI CESAR MEIRA BARBOZA

**ESTUDO DE HEURÍSTICAS PARA O PROBLEMA  
DE ROTEIRIZAÇÃO DE VEÍCULOS COM FROTA  
HETEROGÊNEA FIXA, JANELAS DE TEMPO,  
ENTREGAS FRACIONADAS E LIMITAÇÕES DE  
ACESSO**

Versão Final

Trabalho apresentada à Escola Politécnica  
da Universidade de São Paulo para obten-  
ção do Diploma de Engenheiro de Produção.

Área de Concentração:

Engenharia de Produção

Orientadora:

Débora Pretti Ronconi

São Paulo  
2024

Autorizo a reprodução e divulgação total ou parcial deste trabalho, por qualquer meio convencional ou eletrônico, para fins de estudo e pesquisa, desde que citada a fonte.

#### Catálogo-na-publicação

Barboza, Giovanni

Estudo de heurísticas para o problema de roteirização de veículos com frota heterogênea fixa, janelas de tempo, entregas fracionadas e limitações de acesso / G. Barboza -- São Paulo, 2024.

53 p.

Trabalho de Formatura - Escola Politécnica da Universidade de São Paulo. Departamento de Engenharia de Produção.

1.Roteirização de Veículos 2.Heurísticas 3.Algoritmos Genéticos  
4.Otimização I.Universidade de São Paulo. Escola Politécnica. Departamento de Engenharia de Produção II.t.



## AGRADECIMENTOS

Agradeço, em primeiro lugar, à minha professora orientadora Dra. Débora Pretti Ronconi que, com paciência, dedicação e sabedoria, me guiou ao longo de cada etapa deste trabalho. Seu apoio foi essencial para que eu pudesse enfrentar os desafios e amadurecer academicamente.

Aos meus pais, Janete e Fernando, expresso minha mais profunda gratidão pelo amor, carinho e suporte incondicional que sempre me proporcionaram. Eles foram a base sólida que sustentou toda a minha jornada na graduação, me motivando a perseverar mesmo nos momentos mais difíceis.

Aos meus amigos, sou grato pela parceria e pelas risadas que tornaram esta etapa tão mais leve e memorável.

Por fim, agradeço especialmente ao Quinhones Dutra, cuja generosidade em compartilhar as instâncias que deram origem ao problema abordado neste trabalho foi fundamental para o desenvolvimento desta pesquisa.

*"Não é o mais forte das espécies que sobrevive, nem o mais inteligente, mas o que melhor se adapta às mudanças."*

*– Charles Darwin*



# RESUMO

Este trabalho explora o problema de roteirização de veículos com uma frota heterogênea fixa, considerando restrições de janelas de tempo, entregas fracionadas e limitações de acesso (SDHFFVRPTWSD), proposto inicialmente por Dutra et al. (2023) para melhorar a logística nas entregas da indústria de cimento. O problema foi investigado mais a fundo com o objetivo de gerar soluções alternativas à heurística de inserção originalmente proposta, que poderiam resultar em uma redução dos custos operacionais.

Uma heurística construtiva, baseada na heurística de economias de Clarke-Wright, e um Algoritmo Genético foram implementados para resolver o problema. Foram feitas diversas adaptações no método proposto por Clarke e Wright (1964) para acomodar as várias restrições do problema. No entanto, a ideia central de ordenação dos clientes com base no cálculo das economias foi preservada, resultando em um algoritmo que se mantém fiel ao conceito original, mas capaz de construir rotas viáveis dentro das restrições do SDHFFVRPTWSD.

A heurística de Clarke-Wright desenvolvida mostrou-se flexível o suficiente para gerar, de forma eficiente, uma variedade de soluções distintas. Essas soluções foram cruciais para povoar a população inicial da meta-heurística escolhida, o Algoritmo Genético (AG). O AG incorporou várias soluções da heurística construtiva para lidar com as restrições do SDHFFVRPTWSD. Além disso, operadores de diferentes trabalhos da literatura foram combinados e selecionados com base em sua adequação ao problema proposto.

Ao testar as instâncias propostas por Dutra et al. (2023), utilizando dados reais da operação, foi possível concluir que a abordagem proposta é capaz de gerar soluções viáveis e de boa qualidade, considerando as diversas restrições envolvidas. A heurística de Clarke-Wright, apesar de não superar a heurística de inserção em desempenho direto, mostrou-se altamente útil para o contexto do Algoritmo Genético (AG). Sua principal contribuição foi fornecer diversidade nas soluções iniciais, gerando o espaço de busca explorado pelo AG. Esse aumento na diversidade permitiu ao AG encontrar soluções de maior qualidade em muitos casos. A melhoria foi especialmente evidente ao analisar a redução na média do *gap* percentual entre os custos das heurísticas e as soluções ótimas obtidas por métodos exatos. Enquanto a heurística de inserção apresentou um *gap* médio de 21%, o AG conseguiu reduzir esse valor para 17%. Essa redução demonstra o potencial do AG em refinar soluções iniciais variadas, otimizando o resultado final e oferecendo soluções mais próximas do ótimo em problemas de roteirização complexos em tempo de processamento competitivo.

**Palavras-Chave** –roteirização de veículos, heurísticas construtivas, algoritmos genéticos, janelas de tempo, entregas fracionadas.

# ABSTRACT

This work explores the vehicle routing problem with a fixed heterogeneous fleet, considering time window constraints, split deliveries, and access limitations (SDHFFVRPTWSD), initially proposed by Dutra et al. (2023) to improve delivery logistics in the cement industry. The problem was further investigated to generate alternative solutions to the originally proposed insertion heuristic, aiming to reduce operational costs.

A constructive heuristic based on the Clarke-Wright savings heuristic and a Genetic Algorithm were implemented to solve the problem. Several adaptations were made to the method proposed by Clarke e Wright (1964) to accommodate the problem's various constraints. However, the central idea of sorting customers based on calculated savings was preserved, resulting in an algorithm that remains faithful to the original concept while being capable of constructing feasible routes within the SDHFFVRPTWSD constraints.

The developed Clarke-Wright heuristic proved flexible enough to efficiently generate a variety of distinct solutions. These solutions were crucial for populating the initial population of the chosen metaheuristic, the Genetic Algorithm (GA). The GA incorporated several solutions from the constructive heuristic to handle the SDHFFVRPTWSD constraints. Furthermore, operators from various works in the literature were combined and selected based on their suitability for the proposed problem.

By testing the instances proposed by Dutra et al. (2023), using real operational data, it was possible to conclude that the proposed approach can generate viable and high-quality solutions, considering the diverse constraints involved. The Clarke-Wright heuristic, despite not outperforming the insertion heuristic in direct performance, proved highly useful in the context of the Genetic Algorithm. Its main contribution was providing diversity in the initial solutions, expanding the search space explored by the GA. This increase in diversity allowed the GA to find higher-quality solutions in many cases. The improvement was particularly evident when analyzing the reduction in the average percentage gap between the heuristic costs and the optimal solutions obtained by exact methods. While the insertion heuristic presented an average gap of 21%, the GA reduced this value to 17%. This reduction demonstrates the potential of the GA to refine varied initial solutions, optimizing the final result and offering solutions closer to the optimum for complex routing problems with competitive processing times.

**Keywords** – vehicle routing, constructive heuristics, genetic algorithms, time-windows, split deliveries.

# LISTA DE FIGURAS

|   |  |    |
|---|--|----|
| 1 | Procedimento de correção de concomitâncias. Etapas: (1) identificação;<br>(2) correção (3) troca para corrigir janelas de tempo. . . . . | 32 |
| 2 | Cruzamento de Troca de Rotas (CTR). . . . .  | 38 |

## LISTA DE TABELAS

|   |  |    |
|---|--|----|
| 1 | Resultados da solução das instâncias do problema por métodos exatos . . .                                  | 27 |
| 2 | Parâmetros dos clientes e depósito para a instância-exemplo. . . . .                                       | 29 |
| 3 | Matriz de limitação de acesso da instância-exemplo. . . . .  | 29 |
| 4 | Matrizes das distâncias (km) e do tempo de percurso (horas) da instância-exemplo, respectivamente. . . . . | 29 |
| 5 | Resultado dos algoritmos nas instâncias de teste. . . . .  | 42 |
| 6 | Resultados dos testes Kolmogorov-Smirnov fornecidos pelo <i>Scipy</i> . . . . .                            | 45 |

## LISTA DE ABREVIATURAS E SIGLAS

|            |  |
|------------|--|
| 1M         | Mutação de troca de 1 nível  |
| 2M         | Mutação de troca de 2 níveis   |
| AG         | Algoritmo Genético   |
| ANOVA      | Análise de Variância   |
| CTR        | Cruzamento de Troca de Rotas   |
| CVRP       | <i>Capacitated Vehicle Routing Problem</i>   |
| CW         | Clarke-Wright  |
| HI         | Heurística de Inserção   |
| HFVRPTWSD  | <i>Heterogeneous Fleet Vehicle Routing Problem with Time Windows and Split Deliveries</i>                      |
| KS         | Kolmogorov-Smirnov   |
| LSM        | <i>Local Search Mutation</i>   |
| MDE        | Mínima Distância Extra   |
| Op         | Ótimo  |
| OX         | <i>Ordered Crossover</i>   |
| RBX        | <i>Route Based Crossover</i>   |
| SDHFFVRPTW | <i>Site-Dependent Heterogeneous Fixed Fleet Vehicle Routing Problem with Time Windows and Split Deliveries</i> |
| SBX        | <i>Sequence Based Crossover</i>  |
| SREX       | <i>Selective Route Crossover</i>   |
| TSP        | <i>Travelling Salesman Problem</i>   |
| VRP        | <i>Vehicle Routing Problem</i>   |
| VRPSD      | <i>Vehicle Routing Problem with Split Deliveries</i>   |
| VRPTW      | <i>Vehicle Routing Problem with Time Windows</i>   |

# SUMÁRIO

|          |   |           |
|----------|---|-----------|
| <b>1</b> | <b>Introdução</b>   | <b>12</b> |
| 1.1      | Motivação . . . . .   | 12        |
| 1.2      | Objetivos . . . . .   | 13        |
| <b>2</b> | <b>Revisão bibliográfica</b>                                | <b>15</b> |
| 2.1      | Abordagens na literatura . . . . .                          | 15        |
| 2.2      | Adaptações de Clarke-Wright . . . . .                       | 18        |
| 2.3      | Algoritmo Genético . . . . .                                | 19        |
| <b>3</b> | <b>Formulação matemática</b>                                | <b>22</b> |
| 3.1      | Elementos da formulação . . . . .                           | 22        |
| 3.1.1    | Índices . . . . .   | 22        |
| 3.1.2    | Parâmetros . . . . .  | 22        |
| 3.1.3    | Variáveis de decisão . . . . .                              | 23        |
| 3.2      | Modelo de Programação Linear Inteira Mista (PLIM) . . . . . | 24        |
| 3.3      | Solução exata . . . . .                                     | 26        |
| <b>4</b> | <b>Métodos heurísticos</b>                                  | <b>28</b> |
| 4.1      | Algoritmo de economias de Clarke-Wright . . . . .           | 30        |
| 4.1.1    | Etapas do procedimento . . . . .                            | 30        |
| 4.1.2    | Exemplo de solução . . . . .                                | 33        |
| 4.2      | Algoritmo genético . . . . .                                | 34        |
| 4.2.1    | Inicialização . . . . .                                     | 34        |
| 4.2.2    | Seleção . . . . .   | 36        |
| 4.2.3    | Recombinação . . . . .                                      | 36        |

|          |   |           |
|----------|---|-----------|
| 4.2.4    | Mutação . . . . .                                 | 38        |
| 4.2.5    | Sobrevivência . . . . .                           | 38        |
| 4.2.6    | Algoritmo geral . . . . .                         | 39        |
| 4.2.7    | Exemplo de solução . . . . .                      | 40        |
| <b>5</b> | <b>Resultados</b>                                 | <b>41</b> |
| 5.1      | Comparativo de resultados . . . . .               | 41        |
| 5.2      | Análise estatística . . . . .                     | 44        |
| 5.2.1    | Teste de normalidade Kolmogorov-Smirnov . . . . . | 44        |
| 5.2.2    | ANOVA . . . . .                                   | 45        |
| <b>6</b> | <b>Conclusão</b>                                  | <b>46</b> |
|          | <b>Referências</b>                                | <b>48</b> |

# 1 INTRODUÇÃO

## 1.1 Motivação

Diariamente, variadas técnicas matemáticas são aplicadas com o intuito de otimizar a operação logística de distribuição de mercadorias; a atividade de transporte não agrega valor ao produto e, portanto, há grande interesse em minimizar os desperdícios nessa área. Almejando esse objetivo, problemas de roteirização de veículos têm um papel fundamental em limitar os custos fixos e variáveis da logística de distribuição.

O problema de roteirização de veículos (*Vehicle Routing Problem - VRP*) envolve o projeto de rotas de entrega e/ou coleta de custo mínimo, partindo de um ou mais depósitos, para um número de clientes e sujeito a restrições adicionais (ARENALES et al., 2007). Os estudos na área de gerenciamento da distribuição logística buscam formular restrições adicionais ao problema clássico de roteirização para melhor aproximá-lo à realidade – contudo, a maioria dos estudos se concentra em instâncias simplificadas do problema, as quais não acomodam restrições ou objetivos comumente encontrados na prática (KRITIKOS; IOANNOU, 2013). Portanto, é de grande interesse a abordagem de modelos que, considerando uma gama maior de condições reais da operação logística, obtêm resultados mais refinados.

O principal desafio acompanha a formulação de VRPs mais complexos é sua solução; assim como o problema do caixeiro viajante (*Travelling Salesman Problem - TSP*) – do qual o VRP é uma ramificação – o VRP é um problema *NP-hard* (problemas de classe não-determinística que são de difícil resolução), mas de um ponto de vista prático, VRPs são, em geral, muito mais difíceis de resolver do que TSPs de tamanho similar, de forma que enquanto muitos trabalhos focam em obter a solução exata dos TSPs, o uso de heurísticas para resolver VRPs é mais recorrente. (LAPORTE; NOBERT, 1987).

Assim, um dos primeiros estudos no qual foram propostas heurísticas para resolver o VRP se preocupou em desenvolver a variação do problema considerando a capacidade dos veículos: Clarke e Wright (1964) formularam o chamado CVRP e propuseram a heurística



de economias para resolvê-lo de forma sequencial. A partir disso, uma série de heurísticas foram desenvolvidas para lidar com complicações desse modelo, aproveitando detalhes de formulação e implementação contemplados nos estudos anteriores e adicionado as técnicas necessárias para lidar com nuances adicionais.

## 1.2 Objetivos

O problema de roteirização de veículos com frota heterogênea fixa, janelas de tempo, entregas fracionadas e limitações de acesso (em inglês, *Site-Dependent Heterogeneous Fixed Fleet Vehicle Routing Problem with Time Windows and Split Deliveries* - SDHFFVRPTWSD) apresenta diversas particularidades que tornam sua solução desafiadora e próxima de casos reais. Destacam-se a heterogeneidade da frota, com variações de capacidade e custos de frete devido ao uso de diferentes tipos de veículos no mesmo problema; as janelas de tempo que devem ser respeitadas para as entregas; a localização geográfica dos clientes, que influencia a escolha dos veículos, o custo variável por distância percorrida e o tempo de entrega; e o fracionamento das entregas, ou seja, a possibilidade de um veículo complementar uma entrega iniciada por outro.

O trabalho desenvolvido por Dutra et al. (2023) aborda essas nuances ao propor uma heurística de inserção sequencial para resolver o problema no contexto de uma indústria cimenteira localizada no interior do estado de São Paulo. O procedimento heurístico demonstrou ser consideravelmente mais rápido que a solução exata, comprovando sua aplicabilidade em situações reais de roteirização em *software*. No entanto, enquanto o método exato alcançou, em média, uma redução de 24% nos custos em relação aos dados reais, o método heurístico proposto obteve uma média de 9%, indicando que há um *gap* considerável para o aprimoramento do procedimento heurístico.

Meta-heurísticas são estratégias gerais para resolver problemas baseadas em técnicas de busca inteligente (WINSTON, 2003); elas poderiam ser aplicadas para obter melhores resultados ao SDHFFVRPTWSD. Além disso, há espaço para criação de novos procedimentos heurísticos que podem trazer melhores soluções ao problema. Portanto, o presente trabalho busca aprimorar a solução do problema abordado por (DUTRA et al., 2023), muito relevante em problemas reais, com uma série de restrições pouco abordadas em conjunto em problemas da literatura. Isso será feito por meio do estudo de procedimentos heurísticos alternativos e de meta-heurísticas, além da análise de possíveis mudanças na formulação a fim de melhor contemplar a realidade da operação e melhorar a performance da roteirização.

O trabalho está dividido em seis capítulos. No Capítulo 2 é feita uma revisão bibliográfica dos estudos relevantes para a formulação e resolução do problema. No Capítulo 3 é feita a formulação matemática do problema e resolução por meio de métodos exatos. No Capítulo 4 é discutida a implementação da heurística e da meta-heurística escolhida para solucionar o problema. No Capítulo 5 ocorre a discussão dos resultados da aplicação dos algoritmos heurísticos com os dados do problema e comparação da performance com a solução exata. Por fim, o Capítulo 6 traz as conclusões do trabalho.

## 2 REVISÃO BIBLIOGRÁFICA

Neste capítulo, são apresentadas as principais abordagens da literatura para problemas de roteirização de veículos (VRP), abrangendo desde as formulações clássicas até suas variações mais complexas, como o VRP com janelas de tempo (VRPTW), frota heterogênea (HFVRP), entregas fracionadas (*Split Deliveries* VRP) e dependência de local (*Site Dependent* VRP), culminando na análise do problema que combina todas essas características. Além disso, são discutidas adaptações do algoritmo de Clarke-Wright, com foco em sua aplicação ao VRPTW, e as contribuições de algoritmos genéticos, abordando suas implementações, desafios e aplicações específicas em diferentes variantes do VRP.

### 2.1 Abordagens na literatura

Os problemas de roteirização de veículos (VRP, do inglês *Vehicle Routing Problems*) têm sido amplamente estudados na literatura devido à sua relevância em otimizar operações logísticas, especialmente em cenários com múltiplas restrições e alta complexidade computacional. O trabalho de Dantzig e Ramser (1959), *The truck dispatching problem* é amplamente reconhecido como o ponto de partida para o estudo do Problema de Roteamento de Veículos (VRP). A partir do problema do caixeiro viajante (TSP), os autores foram um dos primeiros a construir um modelo matemático para minimizar os custos de distribuição de combustíveis para postos de gasolina usando uma frota limitada de veículos.

A formulação proposta em Dantzig e Ramser (1959) serviu de base para a maioria dos trabalhos que tratam do VRP, introduzindo elementos básicos de representação que se repetem na maioria dos trabalhos subsequentes, como a matriz de distância ( $d_{ij} \quad \forall i, j$ ) – que, no artigo, determina diretamente o custo da rota, as variáveis  $x_{ij}$  que pareiam a ordem em que dois clientes ( $i$  e  $j$ ) são atendidos e restrições, como  $\sum x_{ij} = 1 \quad \forall i, j$  que garante o funcionamento do modelo de construção de rotas impedindo o atendimento ao mesmo cliente mais de uma vez e restrições que limitam a soma das distâncias de uma

rota. O trabalho já prevê também restrições de capacidade dos veículos ( $\sum q_i \leq C$  para uma rota, onde  $q_i$  é a demanda do cliente  $i$  e  $C$  a capacidade do veículo). Os autores citam alguns métodos que podem ser aplicados a heurísticas construtivas e buscas locais, o que inspirou diversos trabalhos consequentes. Clarke e Wright (1964) desenvolvem em seu trabalho *Scheduling of vehicles from a central depot to a number of delivery points* uma heurística voltada a solucionar o problema de Dantzig e Ramser (1959) (VRP com capacidade) baseada no conceito de economias que obteve bons resultados e, por esse motivo, será tratada em especial na próxima seção desse capítulo. A heurística de Clarke e Wright (1964) introduz o conceito de economias que busca identificar pares de clientes cuja combinação em uma mesma rota resulta em uma economia de distância ou custo em comparação ao atendimento separado e adicioná-los sequencialmente à rota até que se atinja a capacidade do veículo. Essas economias são calculadas pela fórmula:

$$s_{ij} = d_{0i} + d_{0,j} - d_{ij}$$

Onde os índices representam o depósito 0 e os clientes  $i$  e  $j$ ,  $d$  é a distância e  $s$  a economia do par de clientes.

A partir da formulação clássica do VRP, diversos autores buscaram implementar nuances que melhor aproximam o problema de casos reais. O problema com restrições de janelas de tempo (VRPTW - *VRP with Time Windows*) foi uma das primeiras complexidades abordadas. O trabalho de Solomon (1987) teve extrema importância no avanço do estudo em VRPTWs pois, além de propor a formulação com as restrições com janela de tempo, na qual a variável  $b_i$  de tempo de início do atendimento deve garantir que o atendimento do cliente  $i$  seja feito dentro da janela de tempo  $[e_i, l_i]$ , apresenta três principais métodos para solucionar o problema: uma heurística construtiva baseada em economias que remonta o trabalho de Clarke e Wright (1964) para as novas restrições, uma heurística de varredura baseada em agrupamentos por proximidade e uma heurística de inserção.

A última é de grande relevância porque fornece com maior efetividade soluções factíveis no âmbito de janelas de tempo, o que se mostrou um desafio quando se trabalha com heurísticas baseadas na distância. Constrói-se rotas iniciais a partir de um critério de custos e, por meio de dois outros critérios  $c_1$  e  $c_2$  avalia-se entre quais pares de clientes  $i, j$  é mais vantajoso inserir o cliente  $u$  entre todos os espaços possíveis para uma inserção factível. Enquanto  $c_1(i, u, j)$  é um critério de distância semelhante ao conceito de economias,  $c_2(i, u, j)$  avalia o tempo gasto ao adicionar o cliente  $u$  na posição (SOLOMON, 1987). Por meio de um parâmetro  $\mu$  que multiplica a distância extra no critério  $c_1$  é possível regular a importância do custo de distância em relação ao tempo – assim é possível dar

mais importância ao critério  $c_2$  se o programa gera muita infactibilidade de tempo. Os resultados que o algoritmo de Solomon (1987) se mostraram satisfatórios, de modo que ele serviu diversos outros trabalhos que abordam o VRPTW com outras complexidades, como as heurísticas construtivas de Kritikos e Ioannou (2013), Dutra et al. (2023) ou até mesmo para gerar soluções iniciais factíveis para meta-heurísticas como no algoritmo genético de Chiang e Hsu (2014).

Para lidar com uma frota com veículos de diferentes tipos de veículos que podem ter diferentes capacidades de carga, velocidades e custos de operação – chamados de VRPs com frota heterogênea (HFVRP) o artigo de Golden et al. (1984) é frequentemente citado porque propõe uma formulação de capacidades e custos capaz de diferenciar cada veículo, assim como mostra uma análise comparativa dos melhores métodos até então (1984) encontrados para resolver o problema com a complexidade aumentada. Uma nuance relevante para aprimorar o problema de frota heterogênea proposta em Dutra et al. (2023) é a dependência de local (*Site Dependent HFVRP*), que considera certos tipos de veículos incapazes de atender determinadas áreas, o que ocorre na realidade por restrições de tamanho dos veículos e de regulações urbanas de tráfego de carga.

Um relaxamento que amplia as possibilidades de melhoria no planejamento de rotas e no uso da frota é permitir entregas fracionadas (*VRP with Split Deliveries - VRPSD*), isto é, viabilizar a divisão da demanda atendida por um cliente em diferentes rotas por meio de entregas fracionadas. Dror et al. (1994), no artigo *Vehicle Routing Problem with Split Deliveries* foram pioneiros na abordagem do VRPSD, introduzindo uma variável contínua  $y_{ij} \in [0, 1]$ <sup>1</sup> para representar a parcela da demanda que é atendida na rota. Além disso, para adaptar o VRP clássico às entregas fracionadas é necessário fortalecer as restrições de eliminação de subrotas (DROR et al., 1994).

Finalmente, trabalhos que combinam as nuances do VRP citadas até agora são de grande relevância para basear presente trabalho. O VRP de frota heterogênea com janelas de tempo e entregas fracionadas (HFVRPTWSD) é tratado em Belfiore e Yoshizaki (2013). A função objetivo busca minimizar tanto o custo proporcional à distância total percorrida por cada veículo, quanto o custo fixo relacionado à utilização do veículo; a formulação das restrições é baseada em Golden et al. (1984) e Dror et al. (1994) (BELFIORE; YOSHIZAKI, 2013). O trabalho se utiliza heurísticas construtivas de inserção para criar uma solução inicial que é explorada pelo método *Scatter Search*.

O trabalho de Dutra et al. (2023) aborda um problema semelhante ao de Belfiore e

---

<sup>1</sup>Neste trabalho, tal como em Dutra et al. (2023) a demanda fracionada é representada por  $f_{ij}^v$

Yoshizaki (2013), com a adição da dependência de local (HFSDVRPTWSD). A formulação se baseia nos artigos comentados até então; a função objetivo busca minimizar tanto o custo proporcional à distância percorrida quanto o tempo ocioso entre as entregas. Isso ocorre porque em seu problema há restrições de não-concomitância de dois veículos no mesmo cliente (cenário permitido pelo fracionamento de entregas), as quais implicam que haja momentos de ociosidade enquanto um veículo espera o término do serviço do outro para iniciar o próprio. Além disso, a adição da dependência de local faz necessário o uso de restrições de local na formulação:  $\sum x_{ij}^v \leq R_j^v \quad \forall i, j, v$  impede a ida do veículo  $v$  ao cliente  $j$  se na matriz de restrição de local  $R$  a variável binária tem valor zero. As soluções do problema são obtidas por meio de uma heurística de inserção adaptada de Solomon (1987) e, considerando a média do *gap* percentual de custos das instâncias criadas no trabalho com dados reais de problemas de roteamento da indústria cimenteira, chega-se a 21% do valor ótimo de custo, obtido por métodos exatos (DUTRA et al., 2023).

Finalmente, o livro *Local Search in Combinatorial Optimization* de Aarts e Lenstra (2018) foi extremamente útil para identificar métodos heurísticos que abordam as restrições do problema em estudo. Ele proporcionou uma compreensão aprofundada sobre heurísticas construtivas e meta-heurísticas altamente eficazes para resolver VRPs, com destaque especial para os Algoritmos Genéticos, tema abordado em maior detalhe posteriormente.

## 2.2 Adaptações de Clarke-Wright

Durante a revisão bibliográfica, é possível constatar que a maioria das heurísticas utilizadas para construir soluções de variações do VRPTW se dão por métodos de inserção como o método de (SOLOMON, 1987) em vez de utilizar o conceito de economias de Clarke e Wright (1964) de forma mais direta. Assim, é interessante explorar mais profundamente aplicações do algoritmo de Clarke-Wright no VRPTW visando adaptá-las para o problema com as demais complexidades, uma vez que a abordagem possui características únicas que podem oferecer vantagens significativas, como a simplicidade no cálculo de economias e a natureza iterativa de construção de soluções.

Uma escolha de funcionamento pertinente à aplicação do algoritmo de Clarke-Wright – assim como em outras heurísticas de inserção é se será uma versão em paralelo ou em série. Enquanto a versão em série constrói uma rota por completo antes de partir para a próxima, a versão em paralelo constrói as rotas conforme for mais oportuno; no caso de Clarke-Wright, a construção em paralelo constrói uma nova rota sempre que não

houver como inserir um dos clientes do par de economias em uma das rotas existentes. A princípio algoritmos em série rendem soluções de melhor qualidade nas primeiras rotas construídas mas, em geral, heurísticas paralelas são superiores que heurísticas sequenciais, mas é necessário um processo de geração de *seeds* nas primeiras (LIU; SHEN, 1999). De fato, principalmente quando se trata de janelas de tempo, a heurística de Clarke-Wright paralela produz mais soluções factíveis porque evita-se que clientes com janela de tempo parecida sejam roteados na mesma rota, o que leva a infactibilidades de tempo, pois é desejável que uma rota tenha clientes com janelas de tempo distribuídas ao longo da jornada. O processo de *seeding* será discutido no Capítulo 4.

Uma aplicação do algoritmo de Clarke-Wright para gerar soluções para uma meta-heurística é proposta no artigo de Campos et al. (2006) "Algoritmos Genéticos e Computação Paralela Para Problemas de Roteirização de Veículos com Janela de Tempo e Entregas Fracionadas". Nele adaptações ao algoritmo de Clarke-Wright são feitas: para lidar com lojas cuja demanda supera a capacidade do veículo, é permitido o fracionamento da demanda, isto é, o veículo com demanda ociosa entrega o quanto pode (considerando sua capacidade) para a loja e a demanda remanescente é suprida nas rotas seguintes (CAMPOS et al., 2006). Outra mudança diz respeito ao cálculo das economias: adiciona-se o custo fixo de se utilizar um veículo, pois esse custo é considerado na função objetivo. Finalmente, adaptações relacionadas à frota heterogênea são feitas: "A cada rota, é atribuído o menor caminhão capaz de atender à demanda da respectiva loja e quando duas lojas são avaliadas para se juntar em uma mesma rota, caso a demanda das duas combinadas ultrapasse a capacidade do veículo atual, o menor veículo capaz de atender a essa demanda é designado para a rota"(CAMPOS et al., 2006). Esse trabalho mostra, por meio dos resultados satisfatórios obtidos do algoritmo genético que utiliza a adaptação do algoritmo Clarke-Wright, que é possível utilizar a heurística das economias para construir soluções iniciais factíveis para o problema com janelas de tempo e demandas fracionadas.

## 2.3 Algoritmo Genético

Buscando aprimorar os resultados da heurística construtiva proposta, é interessante adotar uma meta-heurística devido à sua capacidade de explorar espaços de solução mais amplos e encontrar soluções próximas ao ótimo em problemas complexos. Entre as opções, os algoritmos genéticos têm se destacado nas últimas décadas como uma abordagem robusta e flexível, sendo amplamente utilizados para resolver problemas de roteirização, especialmente para VRPs com limitações de capacidade, onde a combinação de algoritmos

genéticos e outras heurísticas (por exemplo, busca-local e busca-tabu) têm rendido bons resultados (VIDAL et al., 2013).

O *framework* geral dos algoritmos genéticos foi desenvolvido em Holland (1975). Nele, utiliza-se operadores genéticos como cruzamento (seleção e recombinação) e mutação para resolver problemas de otimização. Para aplicar o método aos VRPs, é necessário traduzir o significado dos operadores para o âmbito de busca em rotas. O trabalho de Potvin e Bengio (1996) "*The Vehicle Routing Problem with Time Windows - Part II: Genetic Search*" trata de aplicar o algoritmo genético ao VRPTW e, portanto, lista como os operadores genéticos devem ser interpretados. Inicialmente, a representação do cromossomo é feita diretamente sobre as soluções, isto é, a ordem de atendimento em uma rota é o próprio cromossomo (POTVIN; BENGIO, 1996). Outros autores que utilizam esse tipo de representação para múltiplas rotas (como no problema de frota heterogênea) o fazem considerando o cromossomo como o conjunto de rotas com a ordem direta dos atendimentos (CHIANG; HSU, 2014).

O cruzamento se dá pela combinação de atributos dos cromossomos de dois pais e a seleção é feita de maneira estocástica, enviesada para as melhores soluções (POTVIN; BENGIO, 1996). Enquanto Potvin e Bengio (1996) propõe que a seleção seja feita por número de rotas ou menor tempo total de rota é comum na literatura encontrar uma seleção baseada em uma probabilidade baseada na função de "aptidão" dos cromossomos. Essa função em Stehling e Souza (2017) considera tanto o número de rotas quanto a distância total percorrida e a seleção dos pais é feita com o método da roleta (selecionar múltiplos cromossomos igualmente espaçados no intervalo de probabilidades) para garantir a diversidade de soluções.

Uma vez selecionados os pais é feito o cruzamento de fato por meio da recombinação. Em Potvin e Bengio (1996) esses operadores são classificados em Recombinações Baseadas em Sequência (SBX), nas quais um ou mais clientes são movimentados entre as rotas ou Recombinações Baseadas em Rotas (RBX), nas quais uma rota inteira é trocada entre os cromossomos. A princípio, as recombinações do tipo RBX são mais adequadas ao VRPTW porque não alteram a ordem das rotas e assim não promovem infactibilidade se os pais forem soluções factíveis. Com efeito, no artigo de Stehling e Souza (2017) "Operadores de Recombinação para um Algoritmo Genético Híbrido Aplicado a Problemas de Roteamento de Veículos com Janela de Tempo" foi feita uma análise comparativa entre quatro operadores de recombinação diferentes, sendo três baseados em sequência e um baseado em rotas (Cruzamento de Troca de Rotas - CTR), e o resultado encontrado foi que o operador CTR do tipo RBX é significativamente superior aos outros. O CTR é



baseado no operador de Chiang e Hsu (2014) e busca intensificar as soluções selecionando as piores rotas dos filhos e inserindo as melhores rotas dos pais em seus lugares, de forma cruzada. Para determinar quais são as melhores rotas de cada cromossomo, é utilizada a razão da distância total da rota pelo número de clientes atendidos (CHIANG; HSU, 2014).

Contudo outros trabalhos, como o artigo de Kool et al. (2022) no qual se baseia o pacote *PyVRP* para resolução de VRPs utilizando Python, alternam entre operadores SBX e RBX <sup>1</sup> para promover maior variabilidade, mas é necessário aplicar uma busca local após o operador SBX para corrigir infactibilidades por janela de tempo.

A mutação é um operador que é aplicado aleatoriamente a uma pequena parcela da população para gerar soluções com uma diversidade mínima, controlada pela taxa de mutação (POTVIN; BENGIO, 1996). Há diversos operadores de mutação; em Potvin e Bengio (1996) foi feita uma análise comparativa entre a troca de um nível (1M) na qual ocorre uma troca aleatória entre dois clientes, a troca de dois níveis (2M) na qual ocorre uma troca aleatória entre dois clientes de duas rotas diferentes e a busca local (LSM) que move clientes por meio de Or-opt até encontrar a melhor rota. Entre as opções analisadas, o operador LSM apresentou melhores resultados, mas seu uso causou convergência prematura da população inicial e por esse motivo é recomendado o uso conjunto de diferentes operadores de mutação. Em Stehling e Souza (2017) é utilizado o operador 1M e as trocas ocorrem de 2 a 3 clientes de "distância" para evitar infactibilidades por janelas de tempo.

---

<sup>1</sup> *Ordered Crossover* (OX) e *Selective Route Crossover* (SREX) são os operadores baseados em sequência e rotas em Kool et al. (2022), respectivamente. O SREX funciona de maneira similar ao CTR

## 3 FORMULAÇÃO MATEMÁTICA

Este capítulo apresenta a formulação matemática do problema SDHFFVRPTWSD, conforme discutido por Dutra et al. (2023). A descrição detalhada do modelo matemático servirá como referência fundamental para a análise subsequente e a implementação das estratégias heurísticas propostas.

### 3.1 Elementos da formulação

Considera-se inicialmente o problema de roteamento, no qual há um grafo completo  $G = (N, A)$  em que cada nó, pertencente ao conjunto  $N = \{0, 1, 2, \dots, n\}$  representa um cliente, com exceção do nó 0 que representa o depósito. Já para cada arco do grafo, disposto entre um par de nós  $(i, j)$  pertencente ao conjunto  $A = \{(i, j) : 0 \leq i, j \leq n, i \neq j\}$ , é atribuído um valor de distância  $d_{ij}$ .

#### 3.1.1 Índices

$i, j, p$  nós, clientes ou depósito  
 $k, v$  veículos

#### 3.1.2 Parâmetros

$n$  Número total de clientes  
 $V$  Número total de veículos  
 $K$  Número de modelos de veículos diferentes  
 $t_{ij}$  Duração da viagem de  $i$  a  $j$  (em horas)  
 $R_j^v$  Matriz que indica se o nó  $j$  pode ser atendido pelo veículo  $v$   
 (Se o atendimento é possível,  $R_j^v = 1$ ; caso contrário,  $R_j^v = 0$ )

|          |   |
|----------|---|
| $q_j$    | Demanda do nó $j$ (em toneladas)  |
| $a_v$    | Capacidade do veículo $v$ (em toneladas)  |
| $cf^v$   | Custo do frete do veículo $v$ (em R\$/km)   |
| $d_{ij}$ | Distância do percurso entre $i$ e $j$ (em km)   |
| $M, L$   | Constantes positivas de valor suficientemente grande  |
| $e_j$    | Limite inferior de tempo para o início do atendimento em $j$<br>(em horas, a partir do início da jornada) |
| $s_j$    | Tempo de serviço em $j$   |
| $l_j$    | Limite superior de tempo para o início do atendimento em $j$<br>(em horas, a partir do início da jornada) |

### 3.1.3 Variáveis de decisão

|            |   |
|------------|---|
| $x_{ij}^v$ | 1 se o nó $j$ é atendido após o nó $i$ pelo veículo $v$ ; 0 caso contrário                                |
| $b_j^v$    | Instante de início de atendimento do nó $j$ pelo veículo $v$<br>(em horas, a partir do início da jornada) |
| $f_j^v$    | Fração da demanda do nó $j$ atendida pelo veículo $v$   |
| $Y_j^{vk}$ | 1 se o veículo $v$ antecede o veículo $k$ no atendimento do nó $j$ ; 0, caso contrário.                   |

### 3.2 Modelo de Programação Linear Inteira Mista (PLIM)

$$\min \sum_{i=0}^n \sum_{j=0}^n \sum_{v=1}^V c f^v d_{ij} x_{ij}^v + \left[ \sum_{v=1}^V \sum_{j=0}^n b_j^v / (V \sum_{j=1}^n l_j) \right] \quad (1)$$

Sujeito a:

$$\sum_{j=1}^n x_{0j}^v \leq 1 \quad v = 1, \dots, V \quad (2)$$

$$\sum_{i=0}^n x_{ij}^v \leq R_j^v \quad j = 1, \dots, n; \quad v = 1, \dots, V \quad (3)$$

$$\sum_{v=1}^v \sum_{i=0}^n x_{ij}^v \leq 2 \quad j = 1, \dots, n \quad (4)$$

$$\sum_{i=0}^n x_{ip}^v - \sum_{j=0}^n x_{pj}^v = 0 \quad p = 0, \dots, n; \quad v = 1, \dots, V \quad (5)$$

$$\sum_{j=1}^n q_j f_j^v \leq a_v \quad v = 1, \dots, V \quad (6)$$

$$\sum_{v=1}^v f_j^v = 1 \quad j = 1, \dots, n \quad (7)$$

$$\sum_{i=0}^n x_{ij}^v \geq f_j^v \quad j = 1, \dots, n; \quad v = 1, \dots, V \quad (8)$$

$$b_i^v + s_i + t_{ij} - M(1 - x_{ij}^v) \leq b_j^v \quad i, j = 1, \dots, n; \quad \forall v \quad (9)$$

$$b_i^v - t_{0i} + M(1 - x_{0j}^v) \geq e_0 \quad i = 1, \dots, n; \quad v = 1, \dots, V \quad (10)$$

$$b_i^v - t_{0i} - M(1 - x_{0j}^v) \leq l_0 \quad i = 1, \dots, n; \quad v = 1, \dots, V \quad (11)$$

$$b_j^v \geq \sum_{i=0}^n x_{ij}^v e_j \quad j = 1, \dots, n; \quad v = 1, \dots, V \quad (12)$$

$$b_j^v \leq \sum_{i=0}^n x_{ij}^v (l_j - s_j) \quad j = 1, \dots, n; \quad v = 1, \dots, V \quad (13)$$

$$b_j^v \geq b_j^k + s_j - L(1 - Y_j^{kv}) \quad j = 1, \dots, n; \quad \forall v, k; k \neq v \quad (14)$$

$$\sum_{\substack{k=1 \\ k \neq v}}^V Y_j^{vk} \leq \sum_{\substack{i=0 \\ i \neq j}}^n x_{ij}^v \quad j = 1, \dots, n; \quad v = 1, \dots, V \quad (15)$$

$$\sum_{\substack{k=1 \\ k \neq v}}^V Y_j^{kv} \leq \sum_{\substack{i=0 \\ i \neq j}}^n x_{ij}^v \quad j = 1, \dots, n; \quad v = 1, \dots, V \quad (16)$$

$$Y_j^{vk} + Y_j^{kv} \geq \sum_{\substack{i=0 \\ i \neq j}} x_{ij}^v + \sum_{\substack{i=0 \\ i \neq j}} x_{ij}^k - 1 \quad j = 1, \dots, n; \quad \forall v, k; k \neq v \quad (17)$$

$$0 \leq f_j^v \leq 1 \quad j = 1, \dots, n; \quad v = 1, \dots, V \quad (18)$$

$$b_j^v \leq 0, b_j^v \in \mathbb{R} \quad j = 1, \dots, n; \quad v = 1, \dots, V \quad (19)$$

$$Y_j^{vk} \in \{0, 1\} \quad j = 1, \dots, n; \quad \forall v, k; k \neq v \quad (20)$$

$$x_{ij}^v \in \{0, 1\} \quad \forall i, j, v \quad (21)$$

A função objetivo (1) busca minimizar duas parcelas: (i) a soma dos custos das viagens feitas, que depende da utilização de um veículo, da distância das viagens e se o percurso é feito ou não e (ii) o tempo ocioso entre entregas, cuja minimização depende de distanciar ao máximo o início do atendimento  $b_j^v$  em um cliente de seu limite de fechamento  $l_j$ .

A restrição (2) permite que os veículos saiam do depósito para atender um cliente no máximo uma vez. A restrição (3) diz respeito à limitação de acesso, garantindo que a matriz  $R$ , que determina se o veículo  $v$  é capaz de atender um cliente  $j$  ( $R_j^v = 1$  se o atendimento é possível, 0 caso contrário), seja respeitada.

A restrição (4) limita o número de entregas fracionadas em um mesmo cliente (consequentemente o número de veículos que visitam o mesmo cliente, já que há uma restrição que impede que um veículo visite um cliente duas vezes) a um máximo de dois clientes.

A restrição (5) garante a conservação de fluxo, isto é, se um veículo entra em um nó ele deve sair dele. A restrição (6) garante que a capacidade do veículo é respeitada, ainda que haja entregas fracionadas. A restrição (7) faz com que a soma das entregas fracionadas em um cliente seja suficiente para atender por completo sua demanda. A restrição (8) liga as variáveis de decisão  $x_{ij}^v$  e  $f_j^v$ , fazendo com que se parte ou toda a demanda de um cliente  $j$  for atendida por um veículo  $v$  (representada pela fração  $f_j^v$ ), isso significa pelo menos um percurso desse veículo deve passar pelo nó  $j$ , representado pela soma  $\sum_{i=0}^n x_{ij}^v$ .

A restrição (9) garante que o atendimento em um nó  $j$  não inicie antes que o atendimento no nó  $i$  que o precede tenha terminado e que o tempo de percurso entre  $i$  e  $j$  seja considerado. Essa restrição faz uso de uma constante grande  $M$  que desativa a restrição caso  $i$  não preceda  $j$ ; para cumprir sua função,  $M$  deve ser maior que  $l_j + s_i + t_{ij} - e_j, \forall i, j \in N$ . As restrições (10) e (11) garantem que as viagens estejam condicionadas à janela de tempo do depósito (a restrição (10) garante que a partida dos veículos deve ser no mínimo no momento de abertura do depósito, enquanto a restrição (11) condiciona a partida de todos os clientes no máximo ao momento de fechamento do depósito), e também fazem uso da constante  $M$  para desativá-las quando não há relação de precedência entre o depósito e o

cliente.

As restrições (12) e (13) condicionam o início do atendimento dos clientes à janela de tempo dos últimos. A restrição (14) garante a não concomitância de dois veículos em um cliente. Ela faz uso de outra constante grande  $L$ , que desativa a restrição se somente um veículo atende a totalidade da demanda no cliente. Para cumprir esse objetivo,  $L$  deve ser maior que  $l_j + s + j, \forall j \in N$ .

As restrições (15) e (16) ligam as variáveis de decisão  $Y_j^{vk}$  e  $x_{ij}^v$ , garantindo que um veículo  $v$  só pode preceder um outro veículo  $k$  no nó  $j$  (representado pela variável  $Y_j^{vk}$ ) se o percurso  $ij$  é percorrido pelo veículo  $v$  (representado pela variável  $x_{ij}^v$ ), e vice-versa. A restrição (17) garante a relação de precedência expressa pela variável de decisão  $Y_j^{vk}$ , pois se um veículo precede o outro, um trecho  $i - j$  de cada veículo é percorrido, de forma que o lado direito da inequação se torna igual a 1, o que implica que pelo menos uma das variáveis binárias  $Y_j$  deve ter valor positivo para satisfazer a relação.

Por fim, as restrições (18), (19), (20) e (21) são de domínio das variáveis:  $f_j^v$  é um número real positivo menor que 1,  $b_j^v$  é um número real positivo e ambos  $Y_j^{vk}$  e  $x_{ij}^v$  são variáveis binárias (iguais a 1 ou 0).

### 3.3 Solução exata

Em seguida, 30 instâncias com dados reais do problema SDHFFVRPTWSD proposto por Dutra et al. (2023) foram resolvidos com a relaxação do número máximo de entregas fracionadas.

Neste trabalho, foram utilizadas 32 instâncias de problemas reais de uma empresa da indústria de cimentos, propostos em Dutra et al. (2023), cuja heurística servirá de *benchmark* para os métodos propostos. Assim, para validar a solução apresentada no trabalho original como para familiarizar com a formulação do problema, o modelo de programação linear inteira foi redigido com o *software* IBM ILOG CPLEX Optimization Studio cada uma das instâncias foi executada em um computador pessoal com processador *1,8 GHz Dual-Core Intel Core i5*. Os resultados do método exato tal como os tempos de processamento estão disponíveis na tabela abaixo. Da mesma forma que foi feito em Dutra et al. (2023), a otimização possui um limite de tempo de 4 horas, depois do qual ela é interrompida e a melhor solução encontrada tomada. Assim, para as instâncias I.29, I.30, I.31 e I.32, não há garantia de optimalidade, apenas que essa é a melhor solução encontrada pelo método exato dada a restrição de tempo computacional.

Tabela 1: Resultados da solução das instâncias do problema por métodos exatos

| Instância | Clientes | Veículos | Custo ótimo | Tempo (s) |
|-----------|----------|----------|-------------|-----------|
| I.1       | 5        | 2        | 994         | 0.13      |
| I.2       | 5        | 3        | 1296        | 0.24      |
| I.3       | 5        | 1        | 178         | 0.25      |
| I.4       | 5        | 2        | 935         | 0.14      |
| I.5       | 5        | 4        | 1916        | 2.00      |
| I.6       | 5        | 3        | 1253        | 0.21      |
| I.7       | 5        | 2        | 665         | 0.17      |
| I.8       | 6        | 2        | 1106        | 0.16      |
| I.9       | 6        | 4        | 1304        | 2.28      |
| I.10      | 6        | 5        | 1869        | 0.90      |
| I.11      | 6        | 3        | 1042        | 0.24      |
| I.12      | 6        | 4        | 1993        | 116.20    |
| I.13      | 6        | 6        | 3452        | 1981.54   |
| I.14      | 7        | 6        | 2898        | 56.44     |
| I.15      | 7        | 3        | 938         | 1.27      |
| I.16      | 7        | 4        | 1893        | 1.15      |
| I.17      | 7        | 6        | 1948        | 29.30     |
| I.18      | 7        | 5        | 1418        | 1.54      |
| I.19      | 7        | 6        | 1756        | 34.19     |
| I.20      | 7        | 3        | 1312        | 1.18      |
| I.21      | 8        | 2        | 581         | 0.18      |
| I.22      | 8        | 4        | 2765        | 17.31     |
| I.23      | 8        | 4        | 939         | 1.21      |
| I.24      | 8        | 2        | 707         | 122.51    |
| I.25      | 8        | 5        | 1662        | 176.17    |
| I.26      | 9        | 4        | 2081        | 157.40    |
| I.27      | 9        | 4        | 1079        | 0.60      |
| I.28      | 10       | 4        | 1300        | 5266.41   |
| I.29      | 10       | 4        | 2370        | >4h       |
| I.30      | 10       | 5        | 1707        | >4h       |
| I.31      | 15       | 3        | 1782        | >4h       |
| I.32      | 20       | 4        | 2330        | >4h       |

Fonte: elaboração própria, baseado em Dutra et al. (2023).

## 4 MÉTODOS HEURÍSTICOS

Neste capítulo, são apresentadas e detalhadas as heurísticas utilizadas para a solução do problema de SDHFFVRPTWSD. Primeiramente, é abordada uma heurística construtiva baseada no conceito de economias de Clarke-Wright. A seguir é explicado o funcionamento de um algoritmo genético, que utiliza essa heurística como parte do processo de geração de soluções iniciais. Em ambos os casos, é apresentado um exemplo ilustrativo de como as soluções são geradas e melhoradas. Os algoritmos foram escritos em Python e os códigos-fonte podem ser encontrados no seguinte repositório: <https://github.com/GBarbo/Heuristics-SDHFFVRPTWSD>.

O exemplo utilizado é a mesma instância-exemplo apresentada em Dutra et al. (2023), visto que sua solução aborda todos os desafios do SDHFFVRPTWSD: dependência de local entre os clientes, frota heterogênea, janelas de tempo diferenciadas para os clientes e um cliente com grande demanda que força entregas fracionadas. A instância-exemplo possui os seguintes valores:

Número de clientes ( $n$ ) = 5;

Número de veículos ( $V$ ) = 2;

Demanda dos veículos em toneladas:  $a_1 = 14$ ,  $a_2 = 16$ ;

Custo de frete dos veículos, em R\$/km:  $cf_1 = 4,54$ ,  $cf_2 = 3,13$ .

Os valores de demanda  $q_j$ , janelas de tempo (limites de início  $e_j$  e término  $l_j$ ) e tempo de serviço  $t_j$  de cada nó são expressos na Tabela 2. A matriz de limitação de acesso  $R$  ( $V \times n$ ) de cada veículo está representada na Tabela 3. Já as distâncias e os tempos de todos os percursos possíveis entre dois nós são expressos na Tabela 4



Tabela 2: Parâmetros dos clientes e depósito para a instância-exemplo.

| j | q (ton) | e (h) | s (h) | l (h) |
|---|---------|-------|-------|-------|
| 0 | 0       | 8     | 0.0   | 18    |
| 1 | 18      | 12    | 2.5   | 20    |
| 2 | 0.8     | 10    | 1.0   | 20    |
| 3 | 0.8     | 8     | 1.5   | 14    |
| 4 | 6       | 8     | 1.5   | 12    |
| 5 | 4       | 8     | 1.0   | 13    |

Fonte: adaptado de Dutra et al. (2023).

Tabela 3: Matriz de limitação de acesso da instância-exemplo.

| v \ j | 1 | 2 | 3 | 4 | 5 |
|-------|---|---|---|---|---|
| $v_1$ | 1 | 0 | 0 | 1 | 1 |
| $v_2$ | 1 | 1 | 1 | 1 | 0 |

Fonte: adaptado de Dutra et al. (2023).

Tabela 4: Matrizes das distâncias (km) e do tempo de percurso (horas) da instância-exemplo, respectivamente.

|       |     | Distâncias (km) |     |     |     |     |  |
|-------|-----|-----------------|-----|-----|-----|-----|--|
| i \ j | 0   | 1               | 2   | 3   | 4   | 5   |  |
| 0     | -   | 106             | 116 | 47  | 57  | 58  |  |
| 1     | 106 | -               | 21  | 117 | 127 | 127 |  |
| 2     | 116 | 21              | -   | 119 | 128 | 128 |  |
| 3     | 47  | 117             | 119 | -   | 11  | 12  |  |
| 4     | 57  | 127             | 128 | 11  | -   | 2   |  |
| 5     | 58  | 127             | 128 | 12  | 2   | -   |  |

|       |      | Tempos (h) |      |      |      |      |  |
|-------|------|------------|------|------|------|------|--|
| i \ j | 0    | 1          | 2    | 3    | 4    | 5    |  |
| 0     | -    | 2.64       | 2.91 | 1.17 | 1.41 | 1.44 |  |
| 1     | 2.64 | -          | 0.54 | 2.92 | 3.16 | 3.16 |  |
| 2     | 2.91 | 0.54       | -    | 2.98 | 3.20 | 3.19 |  |
| 3     | 1.17 | 2.92       | 2.98 | -    | 0.28 | 0.30 |  |
| 4     | 1.41 | 3.16       | 3.20 | 0.28 | -    | 0.04 |  |
| 5     | 1.44 | 3.16       | 3.19 | 0.30 | 0.04 | -    |  |

Fonte: adaptado de Dutra et al. (2023).

A solução ótima da instância exemplo foi calculada utilizando o IBM ILOG CPLEX Optimization Studio e pode ser representada pelas rotas 0-4-5-1-0 para o veículo  $v_1$  e 0-3-1-2-0 para o veículo  $v_2$ . Na solução ótima, a demanda é fracionada tal que  $f_1^1 = 0,2$  e  $f_1^2 = 0,8$  e, para evitar concomitância, o veículo  $v_2$  aguarda 0,98 h para atender o cliente 1, sendo que os demais atendimentos são feitos sem tempo ocioso. O custo da solução calculado é de R\$ 2267,81.

## 4.1 Algoritmo de economias de Clarke-Wright

Para resolver o SDHFFVRPTWSD de forma mais rápida do que buscando a solução ótima, detalhada no capítulo anterior, é interessante começar com uma heurística construtiva, e o algoritmo de Clarke e Wright (1964) é um excelente ponto de partida dada sua simplicidade e velocidade computacional. Esse algoritmo inicialmente cria uma rota para cada cliente e, em seguida, calcula economias baseadas na redução de distância ao combinar rotas de dois clientes (CLARKE; WRIGHT, 1964). As rotas são ajustadas de forma iterativa, adicionando clientes com base nas economias, respeitando as restrições de capacidade dos veículos.

### 4.1.1 Etapas do procedimento

Inicia-se o algoritmo calculando, para cada cliente, as economias geradas pelos pares de clientes pela desigualdade triangular, expressa pela fórmula:

$$s_{ij} = d_{0i} + d_{0j} - d_{ij}$$

Em seguida, as economias são ordenadas do maior para o menor valor de economias. Inicia-se então uma lista que conta a parcela da demanda total de cada cliente que já foi atendida e um *loop* que só termina quando a demanda de todos os clientes, iniciada em 0, chega a 1. Essa lista é atualizada após cada inserção com a demanda atendida.

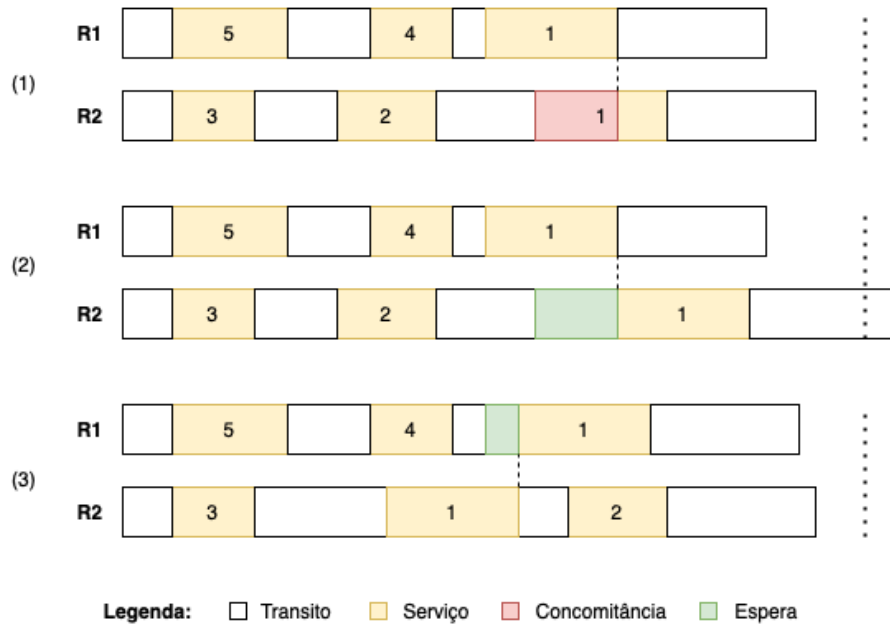
Como apontado no Capítulo 2, o método de Clarke-Wright paralelo é provavelmente a melhor via para utilizá-lo VRPTW e sua aplicação depende de um *seeding*, isto é, da criação inicial de rotas para as quais serão adicionados os clientes, conforme sugerido por Liu e Shen (1999). Como o problema abordado é de frota heterogênea fixa, é interessante inicializar uma rota para cada veículo contendo somente um cliente cada. Visando aumentar a probabilidade de gerar soluções factíveis, o procedimento de criação de rotas

iniciais escolhe os clientes com mais restrições: aqueles clientes com menor número de veículos que podem atendê-lo, como desempate aqueles com menor janela de tempo e como desempate final, aqueles com menor demanda (que portanto não lotarão a capacidade do veículo logo no início do trajeto). Ao final de cada inserção, registra-se a parcela da demanda atendida no vetor  $f$  de entregas fracionadas do veículo. Um contador de carga de cada veículo também é inicializado com as rotas e ele é responsável por acompanhar a lotação de carga (em toneladas) do veículo, representada pela demanda por ele atendida.

Somente após a criação das rotas iniciais é que se utiliza de fato os pares ordenados pelas economias. A cada iteração busca-se, percorrendo a lista de economias, o primeiro par de clientes que (I) possui um dos clientes que pertença a uma das rotas e que ambos não estejam com as demandas completamente atendidas e (II) que a rota gerada pela inserção do cliente seja factível por janelas de tempo e dependência de local. Se ambos os critérios (I) e (II) forem atendidos, a busca é interrompida e o cliente do par que não pertence à rota é inserido na penúltima posição dela (antes do depósito). Novamente a parcela da demanda atendida é registrada; se a demanda do cliente for menor que a capacidade restante do veículo (verificada no contador de carga) e o cliente não foi atendido anteriormente, o veículo deve atender a demanda completa (ou a demanda restante, se o cliente já foi atendido mas não por completo). Se e a demanda do cliente for maior que a capacidade restante, o veículo atenderá esse valor de capacidade remanescente e fechará sua rota, isto é, não será mais possível inserir novos clientes ainda que os critérios (I) e (II) sejam atendidos.

Após a demanda atendida de todos os clientes igualar a 1 é obtida uma solução factível para o SDHFFVRPTWSD, exceto pela restrição de concomitâncias. Para cumpri-la, a solução passa por um procedimento adicional que identifica os clientes e rotas onde há concomitância e retorna o valor de espera mínimo que deve-se ter de forma a manter as rotas obtidas até então. Esse valor é igual à diferença entre o tempo de serviço restante para o primeiro veículo a chegar no cliente terminar o atendimento no momento em que o segundo veículo chega nesse cliente. Se essa solução não for factível por janelas de tempo, tenta-se trocar o cliente concomitante pelo cliente anterior na sequência do último veículo a chegar no cliente e recalcula-se o tempo de concomitância, que agora provavelmente será de espera para o outro veículo e menor do que o tempo atual. A Figura 1 representa as fases do procedimento de correção de concomitância ao longo do tempo, aplicado à instância-exemplo. Observa-se que nessa instância, na etapa (1) foi identificada concomitância de dois atendimentos no cliente 1. Para corrigir esse problema, foi adicionado um tempo de espera à rota R2 na etapa (2). Contudo, esse tempo de espera fez com que essa rota se

Figura 1: Procedimento de correção de concomitâncias. Etapas: (1) identificação; (2) correção (3) troca para corrigir janelas de tempo.



Fonte: elaboração própria.

tornasse inefectível em janelas de tempo; assim a etapa (3) se fez necessária, na qual foi feita uma troca de ordem entre o cliente anterior à concomitância e o cliente concomitante na rota R2. Isso fez com que o veículo da rota R2 chegasse ao cliente 1 antes do veículo de R1, que por sua vez teve de esperar o término do serviço de R2 no cliente 1. Esse método não garante solução factível em janelas de tempo, mas se mostrou eficaz em fazê-lo. Por fim, abaixo se encontra uma representação em pseudocódigo do algoritmo de Clarke-Wright paralelo.

---

**Algorithm 1:** Paralell Clarke-Wright Algorithm for the SDHFFVRPTWSD
 

---

**Data:** Savings for all client pairs, vehicle capacities, demands, time windows, and site restrictions.

**Result:** Routes, fraction matrix  $f$ , and wait times.

```

1 Calculate savings for all pairs of clients and sort in descending order;
2 Initialize unserved_demands  $\leftarrow \emptyset$ , fully_served  $\leftarrow \emptyset$ , routes  $\leftarrow \emptyset$ ,
   available_vehicles  $\leftarrow$  vehicles;
3 while there are unserved demands do
4   if a vehicle is unassigned then
5     Choose the most restricted customer  $j$ ;
6     if route  $[0, j, 0]$  is feasible then
7       Assign  $j$  to a new route and update routes;
8   else
9     foreach pair  $(i, j)$  in sorted savings do
10      if  $i$  or  $j$  belongs to an existing route then
11        Check feasibility of merging  $i$  or  $j$  into the route;
12        if feasible then
13          Merge and update route, load, and routes;
14          break;
15      Mark full vehicles as unavailable;
16  Update served demands and fraction matrix  $f$ ;
17 Fix concomitances with waits;
18 return routes,  $f$ , and wait times;

```

---

#### 4.1.2 Exemplo de solução

Ao executar o algoritmo de Clarke-Wright com os dados da instância exemplo, a solução obtida pode ser representada pelas rotas 0-5-4-1-0 para o veículo  $v_1$  e 0-3-1-2-0 para o veículo  $v_2$ . Na solução ótima, a demanda é fracionada tal que  $f_1^1 = 0,2$  e  $f_1^2 = 0,8$  e, para evitar concomitância, o veículo  $v_2$  aguarda 0,95 h para atender o cliente 1, sendo que os demais atendimentos são feitos sem tempo ocioso. O programa levou 0,086 s para encontrar a solução. O custo da solução calculado é de R\$ 2272,35, o que representa um *gap* de 0,2% da solução ótima.

## 4.2 Algoritmo genético

Para resolver o SDHFFVRPTWSD de maneira eficiente e explorar melhor o espaço de soluções, uma abordagem baseada em meta-heurísticas é recomendada. O Algoritmo Genético (AG) se destaca por sua capacidade de encontrar boas soluções em problemas complexos de otimização combinatória, como o VRPTW e suas variações. Inspirado nos princípios da seleção natural e evolução, o AG utiliza uma população de soluções candidatas que são iterativamente melhoradas por meio de operadores genéticos como seleção, cruzamento e mutação (HOLLAND, 1975). Essa abordagem é particularmente eficiente ao combinar estratégias construtivas, como a de Clarke e Wright (1964), para gerar populações iniciais de alta qualidade, e sua flexibilidade permite incorporar restrições específicas, como janelas de tempo, frota com capacidades heterogêneas e entregas fracionadas.

### 4.2.1 Inicialização

Uma população inicial de soluções diversas é necessária para inicializar o algoritmo genético. Começar por sequências totalmente aleatórias pode exigir um longo tempo computacional para encontrar regiões e soluções promissoras, mas concentrar em apenas algumas poucas regiões pode também fazer com que o processo se limite a encontrar um ótimo local (CHIANG; HSU, 2014). Portanto, é comum que se utilizem heurísticas construtivas hibridizadas com componentes randômicos para gerar a população de soluções iniciais. Logo, aproveitando-se da heurística de Clarke-Wright aplicada ao SDHFFVRPTWSD na seção anterior, um algoritmo foi feito realizando-se alterações para gerar uma diversidade inicial de indivíduos (soluções) em uma população de tamanho definido.

O algoritmo de Clarke-Wright randomizado inicializa da mesma forma descrita na seção anterior; do mesmo modo, a construção das rotas ocorre de maneira paralela. Contudo, em vez de escolher os pares de melhor economia factíveis de serem inseridos na rota, o procedimento de escolha é aleatório mas enviesado para as melhores soluções. No caso dos pares ordenados de maiores para menores economias, as melhores soluções encontram-se mais próximas do primeiro elemento, então o procedimento de escolha deve ser enviesado à esquerda. Assim, o método escolhida a probabilidade baseada em progressão geométrica (fórmula abaixo), na qual a probabilidade de se escolher um par de economias na posição  $k$  da lista é igual ao dobro da probabilidade de escolher o par da posição  $k + 1$ . Esse método foi utilizado em Qin et al. (2015) para fornecer soluções com médio grau de intensificação e diversidade para um algoritmo de busca iterada.

$$p(k) = \frac{1}{2^k \cdot (1 - \frac{1}{2^n})}$$

Devido à grande quantidade de restrições presente no SDHFFVRPTWSD, é necessário realizar certas relaxações para obter uma população inicial com indivíduos suficientes. Na literatura, encontra-se implementações de algoritmo genético usualmente de 50 a 200 indivíduos, mas o algoritmo de Clarke-Wright randomizado gera, em tempo hábil, no máximo 10 soluções factíveis em todas as restrições. Assim, implementou-se um mecanismo que relaxa as restrições de janela de tempo se o número de iterações do *loop* principal sem inserção (isto é, obtê-ve se o par de clientes de forma randômica e não foi possível inseri-lo em nenhuma rota sem violar restrições) atinge  $2 \cdot n$ . Dessa maneira foi possível gerar as demais soluções, sendo que a infactibilidade será tratada por mecanismos de sobrevivência baseados na função de aptidão.

Da mesma forma que no algoritmo da seção anterior, o Clarke-Wright randomizado só lida com concomitâncias após o término das inserções. Contudo, diferente do algoritmo para uma solução no qual a factibilidade é de extrema importância, para construir as soluções iniciais não é realizada a troca de posições caso a solução não-concomitante não obedeça as janelas de tempo; reservam-se as realocações puramente aos operadores genéticos.

Os cromossomos criados são soluções do problema, que incluem listas de rotas relacionadas cada uma a um veículo e listas de entregas fracionadas e tempos de espera de concomitância para cada veículo/cliente. Esses cromossomos são avaliados por uma função de aptidão (*fitness*) que, além de levar em conta o custo da rota proporcional à distância total percorrida, pune os tempos totais de espera, que são também objetivos de minimização do problema, e também cada infactibilidade, conforme a fórmula abaixo:

$$\text{aptidão} = \text{custo} + \frac{t_{\text{espera}}}{1.000} + 10.000 \cdot \text{infact} + 500 \cdot \text{infact\_concom}$$

O peso da penalização por infactibilidade após a correção de concomitâncias (que multiplica "infact\_concom") é reduzido porque experimentalmente houve convergência precoce nos poucos resultados factíveis gerados na população inicial, e porque é relativamente fácil lidar com esse tipo de infactibilidade com trocas de posição conforme é feito no algoritmo apresentado na seção anterior. O peso das infactibilidades foi escolhido tal que, nas instâncias onde se deseja aplicar o algoritmo, uma solução infactível dificilmente terá melhor aptidão que uma solução factível, visto que a penalidade é uma ordem de grandeza maior que os maiores custos obtidos nas soluções ótimas, somados aos tempos

totais de espera em horas, que são divididos por 1000 para não levar maior importância que o custo na escolha das soluções.

### 4.2.2 Seleção

Da mesma forma que na fase de criação da população inicial, a seleção dos pares de indivíduos pais é estocástica enviesada para as melhores soluções. Utiliza-se um ranqueamento linear das probabilidades de se escolher um indivíduo, que são baseadas no inverso da função de aptidão descrita anteriormente. Os pais são escolhidos dois a dois pelo método da roleta, com probabilidade conforme a fórmula abaixo, proposta em Potvin e Bengio (1996). Como melhores soluções possuem menor valor de aptidão, o parâmetro *fitness* da equação abaixo é na verdade o inverso do valor de aptidão calculado na seção anterior.

$$p(i) = \frac{\text{fitness}_i}{\sum_{j=1, \dots, P} \text{fitness}_j}$$

### 4.2.3 Recombinação

Conforme apontado no Capítulo 2, operadores de troca de rotas são mais adequados para lidar com janelas de tempo. Utilizou-se, portanto, o operador de Cruzamento de Troca de Rotas (CTR) proposto por Chiang e Hsu (2014) e apontado em Stehling e Souza (2017) como uma boa opção para se utilizar em algoritmos genéticos com janelas de tempo. Portanto, esse operador foi implementado para o cruzamento de rotas. A ideia geral do operador CTR é de capturar as melhores características dos indivíduos pais em detrimento das piores (STEHLING; SOUZA, 2017).

Primeiro, cada um dos filhos  $O_1$  e  $O_2$  recebe as rotas dos pais  $P_1$  e  $P_2$ , respectivamente, atribuindo a eles o mesmo veículo dos pais que geraram as rotas. Em seguida, para cada um dos pais é identificada a melhor rota e a pior rota do cromossomo. Para determinar isso, para cada rota é calculada uma função de avaliação (abaixo), a rota que apresentar menor valor é a melhor rota e a de maior valor, a pior, conforme a metodologia proposta em Chiang e Hsu (2014):

$$\text{aptidão da rota} = \frac{\text{distância total percorrida}}{\text{número de clientes na rota}}$$

Consequentemente, do filho  $O_1$  é removida a pior rota do pai que o gerou ( $P_1$ ). Em seguida, a esse filho é adicionada a melhor rota do pai  $P_2$ , preservando a ordem de aten-



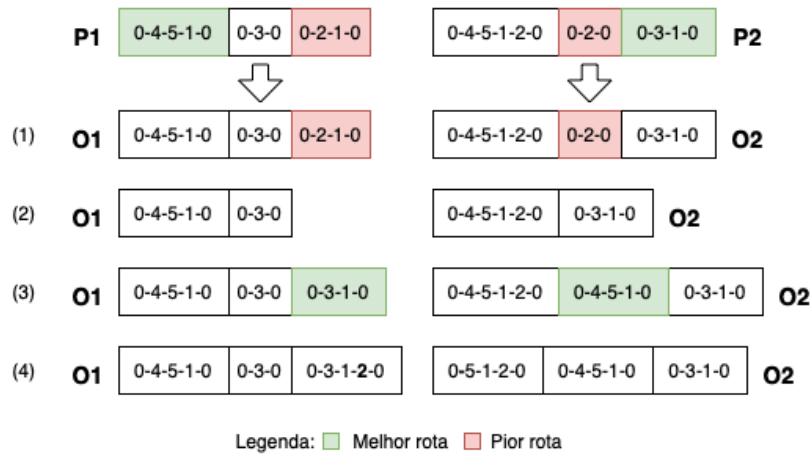
dimento da última. O mesmo é feito para o outro filho, trocando-se os pais. Há vezes que algumas operações consequentes são necessárias para reinserir algum cliente que foi removido da pior rota – essa reinserção é feita pela distância extra mínima, isto é, todas as possíveis posições de inserção na rota são avaliadas e a inserção ocorre na posição que trouxer menor distância extra (MDE), medida pela desigualdade triangular (STEHLLING; SOUZA, 2017). A reinserção se dá na rota cujo veículo possui maior capacidade ociosa, visando evitar infactibilidade por sobrecarga.

Foram necessárias adaptações ao operador para lidar com demandas fracionadas. No vetor de entregas fracionas, as demandas dos clientes retirados da pior rota serão subtraídas dos filhos e as demandas dos clientes inseridos serão também adicionadas aos filhos. Assim, o operador deve garantir que ao final da recombinação, a solução continue atendendo toda a demanda de todos os clientes. Se um cliente for retirado da pior rota e, no momento de sua reinserção, houver outra rota na qual esse cliente é atendido, se possível e factível, em vez de inserir novamente esse cliente na rota, a demanda parcial será atendida nessa outra rota e a reinserção não será necessária. Analogamente, se a melhor rota quando for adicionada a um filho possuir clientes duplicados, o operador modificará o vetor de entregas fracionadas para que não haja super-atendimento da demanda.

É possível que o operador de troca de rotas retire todos os clientes de uma rota, reduzindo o número de veículos que efetivamente realizam entregas. Embora minimizar o número total de veículos não seja objetivo da otimização, isso é desejável pois reduz o custo total da roteirização evitando utilizar veículos com custo de frete mais alto e reduzindo a distância percorrida, o que configura uma clara vantagem da meta-heurística sobre heurísticas construtivas, onde normalmente é considerado vantajoso inserir um cliente em uma rota vazia em vez de adicioná-lo às rotas em construção, ou onde há um procedimento de *seeding* na inicialização que implica em inserir clientes em todas as rotas, como é feito no algoritmo de Clarke-Wright paralelo.

Abaixo se encontra um diagrama representando o procedimento de recombinação proposto. As etapas numeradas são: (1) cópia das rotas dos cromossomos pais para os filhos, (2) remoção das piores rotas, (3) adição das melhores rotas e (4) reparo por reinserção de clientes e demandas faltantes e exclusão de clientes com super-atendimento da demanda.

Figura 2: Cruzamento de Troca de Rotas (CTR).



Fonte: elaboração própria, adaptado de Stehling e Souza (2017)

#### 4.2.4 Mutação

A mutação se dá em uma parcela pequena dos indivíduos filhos ditada pela taxa de mutação, por meio do operador 1M de Potvin e Bengio (1996), que realiza trocas de posição entre dois clientes da mesma rota, "distantes" entre si de dois a três clientes para evitar inafectabilidade por janelas de tempo (STEHLING; SOUZA, 2017). Evitou-se utilizar o operador 2M, que faz trocas de clientes entre rotas, porque o operador de recombinação já cumpre o papel de promover diversidade com um mecanismo de combinação de rotas distintas, e porque esse operador tem grandes chances de causar inafectabilidade por janelas de tempo ou por dependência de local (pois pode trazer para uma rota um cliente que não pode ser atendido para o veículo a ela atribuído).

Uma vez escolhido aleatoriamente o indivíduo que sofrerá a mutação, as trocas ocorrem em rotas e posições também aleatoriamente escolhidas. Após o processo de mutação, o mesmo processo de determinação do tempo de espera para evitar concomitâncias utilizado no procedimento de criação da população inicial é aplicado às soluções filhas e a aptidão é recalculada.

#### 4.2.5 Sobrevivência

Após o processo de mutação, ambas as populações pais e filhas se somam. Contudo, somente metade da população seguirá para a próxima iteração do processo; para determinar quais indivíduos serão descartados, a população total é ranqueada em função da

aptidão. Um operador de elitismo age preservando uma parcela das melhores soluções, evitando que elas passem para o processo de sobrevivência e garantindo que boas soluções não sejam perdidas. Para isso, uma parcela da população com os indivíduos de menor (melhor) aptidão correspondente à taxa de elitismo é separada, enquanto os demais seguem para uma seleção por método da roleta proporcional à probabilidade baseada em aptidão, semelhante à seleção de pais. Uma vez que a soma do número de indivíduos de elite com o número de indivíduos selecionados for igual à metade da população que iniciou a sobrevivência, o processo é encerrado. Assim, é possível que "boas" e "más" soluções coexistam na população mas que haja intensificação suficiente para caminhar em direção a melhores soluções.

#### 4.2.6 Algoritmo geral

O funcionamento geral do algoritmo genético é apresentado no pseudocódigo abaixo. Inicialmente, determina-se os parâmetros necessários para executar o algoritmo genético, nomeadamente o tamanho da população, a taxa de mutação, a taxa de elitismo e o critério de parada. Alguns desses valores são sugeridos pela literatura, outros obtidos por experimentação. O tamanho da população pode ser admitido como um valor entre 50 e 200 indivíduos; a taxa de mutação pode variar de 10% a 20%, conforme visto em algoritmos genéticos aplicados a VRPTW na revisão bibliográfica. O operador de elitismo deve ser usado com moderação para não gerar convergência precoce ou interferir demais no funcionamento do algoritmo; experimentalmente um valor próximo de 10% é adequado. O critério de parada idealmente é a convergência, ou seja, conta-se quantas gerações (iterações) há sem que haja melhora na solução incumbente.

---

**Algorithm 2:** Genetic Algorithm with Clarke-Wright Initialization
 

---

**Data:** Initial population size, mutation rate, elite percentage  $p\_elite$ , termination criterium.

**Result:** Best individual with optimized routes and associated costs.

```

1 Generate initial population using clarke_wright_randomized(size);
2 population ← initial_population;
3 while criterium is unmet do
4   while size(population) < 2 · size do
5     parents ← selection(population);
6     offspring ← crossover(parents);
7     offspring ← mutation(offspring, mutation_rate);
8     population ← population + offspring;
9   population_elite ← elite(p_elite, population);
10  population ← population - population_elite;
11  population ← survival(population);
12  population ← population + population_elite;
13  Evaluate criterium;
14 return best individual from population;

```

---

#### 4.2.7 Exemplo de solução

Foi utilizada na instância-exemplo o valor de 50 iterações sem melhora como critério de parada, taxa de mutação de 20%, taxa de elitismo de 10% e uma população inicial de 50 indivíduos. O tempo de execução do algoritmo foi de 0,322 s e a solução ótima foi encontrada.

## 5 RESULTADOS

Neste capítulo, apresentam-se os resultados obtidos a partir da execução das 32 instâncias de teste nos algoritmos Clarke-Wright e Algoritmo Genético. O objetivo inicial foi avaliar o desempenho de cada método em termos de custo total das rotas geradas e qualidade das soluções encontradas, considerando as particularidades de cada abordagem.

Em um segundo momento, realizou-se uma análise estatística dos dados obtidos. Para verificar a adequação dos resultados às condições de normalidade, aplicou-se o teste de Shapiro-Wilk. Em seguida, utilizou-se a análise de variância (ANOVA) para identificar diferenças significativas entre os desempenhos dos métodos. Esta análise buscou confirmar, com base em fundamentos estatísticos, a hipótese de superioridade ou equivalência entre os algoritmos.

### 5.1 Comparativo de resultados

Para avaliar a efetividade dos algoritmos propostos, utilizaram-se as 32 instâncias de teste criadas por Dutra et al. (2023), baseadas em dados reais de veículos, clientes, distâncias e custos de uma empresa da indústria de cimentos. Dessa forma, foi possível estabelecer uma comparação não apenas entre os métodos explorados neste trabalho, mas também com a heurística de inserção apresentada no estudo original.

Na tabela abaixo se encontram os resultados dos três métodos comparados ao lado de informações sobre as instâncias (identificação  $I$ , número de veículos  $V$  e número de clientes  $n$ ). Os índices representam as heurísticas da comparação ( $HI$  = heurística de inserção de Dutra et al. (2023);  $CW$  = Clarke-Wright paralelo;  $AG$  = Algoritmo Genético) e o caso ótimo ( $OP$ ). Os resultados são dispostos tanto na forma de custo da rota ( $C$ , em R\$), calculado pela primeira parcela da função objetivo do problema que realiza a soma das distâncias ponderada pelo custo de frete do veículo que as percorre, quanto na forma de *gap* percentual, expresso por  $Gap_{Heur} \% = \frac{C_{Heur} - C_{OP}}{C_{OP}}$ .

Tabela 5: Resultado dos algoritmos nas instâncias de teste.

| <b>Inst</b>  | <b>n</b> | <b>V</b> | $C_{OP}$ | $C_{HI}$ | $C_{CW}$ | $C_{AG}$ | $Gap_{HI}$ | $Gap_{CW}$ | $Gap_{AG}$ |
|--------------|----------|----------|----------|----------|----------|----------|------------|------------|------------|
| I.1          | 5        | 2        | 994      | 994      | 994      | 994      | 0%         | 0%         | 0%         |
| I.2          | 5        | 3        | 1296     | 1722     | 2027     | 1473     | 33%        | 56%        | 14%        |
| I.3          | 5        | 1        | 178      | 178      | 182      | 178      | 0%         | 2%         | 0%         |
| I.4          | 5        | 2        | 935      | 1217     | 1022     | 1180     | 30%        | 9%         | 26%        |
| I.5          | 5        | 4        | 1916     | 1989     | 2261     | 2143     | 4%         | 18%        | 12%        |
| I.6          | 5        | 3        | 1253     | 1294     | 1989     | 1253     | 3%         | 59%        | 0%         |
| I.7          | 5        | 2        | 665      | 1170     | 1192     | 1170     | 76%        | 79%        | 76%        |
| I.8          | 6        | 2        | 1106     | 1387     | 1217     | 1217     | 25%        | 10%        | 10%        |
| I.9          | 6        | 4        | 1304     | 1464     | 1903     | 1903     | 12%        | 46%        | 46%        |
| I.10         | 6        | 5        | 1869     | 2164     | 2690     | 1869     | 16%        | 44%        | 0%         |
| I.11         | 6        | 3        | 1042     | 1042     | 1915     | 1119     | 0%         | 84%        | 7%         |
| I.12         | 6        | 4        | 1993     | 1993     | 2551     | 2284     | 0%         | 28%        | 15%        |
| I.13         | 6        | 6        | 3452     | 3714     | 3451     | 3451     | 8%         | 0%         | 0%         |
| I.14         | 7        | 6        | 2898     | 3730     | 3415     | 3415     | 29%        | 18%        | 18%        |
| I.15         | 7        | 3        | 938      | 1139     | 1722     | 967      | 21%        | 83%        | 3%         |
| I.16         | 7        | 4        | 1893     | 2306     | 2656     | 1902     | 22%        | 40%        | 0%         |
| I.17         | 7        | 6        | 1948     | 2057     | 1970     | 1970     | 6%         | 1%         | 1%         |
| I.18         | 7        | 5        | 1418     | 2025     | 2948     | 2075     | 43%        | 108%       | 46%        |
| I.19         | 7        | 6        | 1756     | 2015     | 2665     | 2069     | 15%        | 52%        | 18%        |
| I.20         | 7        | 3        | 1312     | 1312     | 1553     | 1548     | 0%         | 18%        | 18%        |
| I.21         | 8        | 2        | 581      | 586      | 581      | 581      | 1%         | 0%         | 0%         |
| I.22         | 8        | 4        | 2765     | 3500     | 4013     | 3795     | 27%        | 45%        | 37%        |
| I.23         | 8        | 4        | 939      | 1767     | 1481     | 939      | 88%        | 58%        | 0%         |
| I.24         | 8        | 2        | 707      | 832      | 761      | 761      | 18%        | 8%         | 8%         |
| I.25         | 8        | 5        | 1662     | 1857     | 2902     | 2175     | 12%        | 75%        | 31%        |
| I.26         | 9        | 4        | 2081     | 2548     | 2420     | 2456     | 22%        | 16%        | 18%        |
| I.27         | 9        | 4        | 1079     | 1840     | 1876     | 1622     | 70%        | 74%        | 50%        |
| I.28         | 10       | 4        | 1300     | 1362     | 1793     | 1570     | 5%         | 38%        | 21%        |
| I.29         | 10       | 4        | 2370     | 2502     | 3310     | 3464     | 6%         | 40%        | 46%        |
| I.30         | 10       | 5        | 1707     | 1743     | 2057     | 2029     | 2%         | 20%        | 19%        |
| I.31         | 15       | 3        | 1782     | 1997     | 2390     | 1838     | 12%        | 34%        | 3%         |
| I.32         | 20       | 4        | 2330     | 3541     | 2588     | 2601     | 52%        | 11%        | 12%        |
| <b>Média</b> |          |          |          |          |          |          | <b>21%</b> | <b>37%</b> | <b>17%</b> |

Fonte: elaboração própria.

Para obter as soluções, os programas que contêm a heurística Clarke-Wright paralela e o Algoritmo Genético foram executados em sequência para cada uma das instâncias. Os parâmetros utilizados no Algoritmo Genético são 50 iterações sem melhora como critério de parada, taxa de mutação de 20%, taxa de elitismo de 10% e uma população inicial de 100 indivíduos.

A execução do código em Python se deu por meio de um computador com processador *1,8 GHz Dual-Core Intel Core i5*. O tempo de execução das as instâncias da heurística construtiva variou de 0,1 a 0,7 milissegundos. Já o tempo de execução do algoritmo genético variou de 0,4318 a 3,5967 segundos, sendo que para ambos as instâncias I.1 e I.32 foram aquelas que menos e mais tomaram tempo, respectivamente; com as demais instâncias apresentando valores intermediários de tempo, observa-se correlação entre o número de clientes e veículos e o tempo necessário para executá-las. Portanto, o tempo despendido não é problemático para uma aplicação prática como é em certas instâncias quando executada pelo método exato.

Todas as soluções encontradas são factíveis de acordo com as restrições do problema enunciadas no Capítulo 3. Apesar de, nas heurísticas, não haver restrições sobre número máximo de entregas fracionadas, em nenhuma das soluções houve mais de um atendimento no mesmo cliente.

Enquanto a heurística de Clarke-Wright forneceu soluções ótimas para três instâncias (I.1, I.13 e I.21), o Algoritmo Genético obteve oito soluções ótimas (I.1, I.3, I.6, I.10, I.13, I.16, I.21 e I.23), em comparação com a heurística de inserção de Dutra et al. (2023) que forneceu cinco soluções ótimas (I.1, I.3, I.11, I.12 e I.20). As soluções ótimas encontradas pela heurística de Clarke-Wright foram todas encontradas pelo Algoritmo Genético; ora, o último usa o primeiro para gerar suas rotas originais e por isso tem grandes chances de gerá-lo em uma de suas soluções. Em cinco instâncias, o Algoritmo Genético partiu de soluções de Clarke-Wright não-ótimas e obteve a optimalidade.

Houve duas instâncias em que o algoritmo de inserção sequencial obteve a solução ótima e os demais não. Da mesma forma, houve instâncias em que a heurística de inserção obteve resultados melhores que ambas as outras. Nesses casos, observa-se que os valores da heurística de Clarke-Wright e do Algoritmo Genético estão muito próximos (por exemplo, nas instâncias I.9 e I.20), e que por isso se pode inferir que a diversidade de soluções exploradas não foi suficiente para escapar de um ótimo local.

## 5.2 Análise estatística

Nesta seção, foi realizada uma análise estatística para avaliar o desempenho das duas heurísticas propostas: o método de Clarke-Wright paralelo e o Algoritmo Genético. Além disso, os resultados dessas heurísticas foram comparados a uma heurística de inserção de Dutra et al. (2023), utilizada como referência.

Inicialmente, aplicou-se o teste de normalidade de Kolmogorov-Smirnov aos resultados das três heurísticas, com o objetivo de verificar se os dados seguiam uma distribuição normal. Essa verificação foi essencial, pois a análise de variância (ANOVA), utilizada em seguida, assume a normalidade dos dados como uma de suas premissas.

A ANOVA foi então empregada para comparar as médias de desempenho das heurísticas e determinar se existiam diferenças estatisticamente significativas entre elas. Essa análise permitiu uma avaliação objetiva e fundamentada sobre a eficácia das abordagens propostas em relação à heurística de inserção de referência, contribuindo para a validação dos resultados e a identificação da melhor solução para o problema em questão.

### 5.2.1 Teste de normalidade Kolmogorov-Smirnov

Para avaliar a aderência das amostras de custo das soluções das três heurísticas comparadas, aplicou-se o teste de Kolmogorov-Smirnov (KS), que mede a distância entre os valores das amostras de uma função teórica, no caso, uma distribuição normal de mesma média.

Utilizou-se a biblioteca *Scipy* para aplicar o teste KS. Conforme a documentação da biblioteca, temos as seguintes hipóteses para um teste bicaudal:

$H_0$ : Os dados são aderentes à distribuição especificada.

$H_1$ : Os dados **não** são aderentes à distribuição especificada.

Admitindo-se um valor de significância de 5%, se o p-valor encontrado é menor que 0,05, rejeita-se a hipótese nula ( $H_0$ ) em favor da alternativa ( $H_1$ ) padrão: os dados não estão distribuídos de acordo com a distribuição normal (SCIPYDEVELOPERS, 2024). Na tabela abaixo estão representados os resultados do teste KS para os três conjuntos analisados. Como nenhum dos três p-valores é inferior à significância admitida de 5%, não se rejeita a hipótese nula de que cada um é aderente à distribuição normal de mesma média. Além disso, a tabela mostra para cada caso a estatística  $D$ , que representa a distância



máxima, em módulo, entre o valor da distribuição acumulada da amostra e da distribuição teórica.

Tabela 6: Resultados dos testes Kolmogorov-Smirnov fornecidos pelo *Scipy*.

| Dados                  | Estatística D | p-valor |
|------------------------|---------------|---------|
| Heurística de Inserção | 0,153         | 0,403   |
| Clarke-Wright          | 0,072         | 0,992   |
| Algoritmo Genético     | 0,119         | 0,707   |

Fonte: elaboração própria.

## 5.2.2 ANOVA

Uma vez que o teste de aderência para os três conjuntos indicou normalidade, foi possível realizar uma análise de variância. Novamente, foi utilizada a biblioteca *Scipy* para realizar a ANOVA. A análise produziu uma estatística F de 0,879 e um p-valor 0,419. Assim, não foi possível rejeitar a hipótese nula de que, com significância de 5%, não há diferenças significativas entre as médias dos custos das heurísticas analisadas.

Assim, identifica-se a necessidade de se realizar testes em um conjunto maior de instâncias para verificar se um método é melhor que o outro com significância estatística, uma vez que a variação dos dados entre os grupos não é significativamente maior do que a variação interna dos custos dentro de cada heurística.

## 6 CONCLUSÃO

Neste trabalho, o problema de roteirização de veículos com uma frota heterogênea fixa, considerando restrições de janelas de tempo, entregas fracionadas e limitações de acesso (SDHFFVRPTWSD), proposto inicialmente por Dutra et al. (2023) para fornecer melhorias logísticas nas entregas de uma empresa da indústria de cimentos, foi explorado mais a fundo com o intuito de gerar soluções por métodos alternativos à heurística de inserção proposta que podem gerar redução de custo da operação logística.

Assim, uma heurística construtiva e uma meta-heurística foram implementadas para atacar o problema. A heurística de Clarke-Wright, que possui uma lógica de construção das soluções mais simples e diferente da heurística de inserção, mais comum na literatura por sua efetividade em gerar soluções para problemas com restrições de janela de tempo e implementada por Dutra et al. (2023). Foram necessárias diversas adaptações ao método proposto em Clarke e Wright (1964) para acomodar as variadas restrições do problema, mas a lógica de ordenação de clientes para inserção pelo cálculo de economias foi preservada, de modo que foi possível obter, ao final, um algoritmo fiel à ideia original mas capaz de construir rotas factíveis sob as restrições do SDHFFVRPTWSD.

A heurística de Clarke-Wright desenvolvida mostrou-se flexível o suficiente para ser adaptada de maneira eficiente, permitindo a geração de uma variedade de soluções distintas. Essas soluções foram fundamentais para povoar a população inicial da meta-heurística escolhida, o Algoritmo Genético. O AG, por sua vez, incorporou várias das soluções desenvolvidas na heurística construtiva para lidar com as restrições do SDHFFVRPTWSD. Além disso, foi possível combinar operadores de diferentes trabalhos da literatura, selecionando aqueles que mais se adequam ao problema proposto.

Ao executar as instâncias propostas em Dutra et al. (2023) feitas a partir de dados reais da operação, é possível concluir que a abordagem proposta é capaz de gerar soluções viáveis e com boa qualidade, considerando as diversas restrições envolvidas. A heurística de Clarke-Wright não foi capaz de gerar soluções superiores à heurística de inserção, mas sua adaptação teve grande utilidade para fornecer uma diversidade de soluções iniciais

suficiente para que o Algoritmo Genético encontre boas soluções, em muitos casos inclusive melhores que aquelas da heurística de inserção, como evidenciado pela melhoria na média do *gap* percentual quando comparados os custos das heurísticas com os das soluções ótimas encontradas pela resolução do modelo de programação linear por métodos exatos (de 21% na heurística de inserção para 17% no Algoritmo Genético). Entretanto, não foi possível afirmar com significância estatística e efetividade de uma heurística comparada às demais; é possível que trabalhos futuros, contando com uma gama maior de instâncias reais de grande porte, possam fazê-lo.

Além disso, a comparação entre os diversos cenários e a heurística de referência revelou que a proposta apresenta um desempenho computacional competitivo, com tempos de processamento reduzidos. Isso destaca a eficácia do modelo desenvolvido para resolver problemas do mundo real que exigem flexibilidade e adaptabilidade, onde o uso de métodos exatos se torna inviável devido aos elevados custos e tempo computacional. Aqui está a frase completada:

Pode-se aprimorar as soluções do Algoritmo Genético em trabalhos futuros. Seria interessante aprimorar a escolha dos diversos parâmetros utilizados no algoritmo por meio de métodos automatizados de ajuste, como o IRACE (*Iterated Racing*), que permite a seleção eficiente de configurações de parâmetros, otimizando o desempenho do algoritmo sem a necessidade de um ajuste manual intensivo, e garantindo a adaptação do modelo a diferentes instâncias do problema (LÓPEZ-IBÁÑEZ et al., 2016). Além disso, a diversidade de soluções observada indica que o AG poderia se beneficiar de uma gama maior de soluções iniciais para escapar de ótimos locais. Um modo de fazê-lo seria combinar soluções da heurística de Clarke-Wright com soluções da heurística de inserção, assim propiciando que o Algoritmo Genético explore espaços de solução que contenham pelo menos os ótimos atingíveis pelas duas heurísticas. Outras medidas que poderiam trazer potenciais melhorias em trabalhos futuros incluem a exploração de outros operadores de recombinação e mutação, ou mesmo a hibridização com outras meta-heurísticas ou com a busca local.

## REFERÊNCIAS

- AARTS, E.; LENSTRA, J. K. **Local search in combinatorial optimization**. [S.l.: s.n.], 2018. ISSN 01605682.
- ARENALES, M.; ARMENTANO, V.; MORABITO, R.; YANASSE, H. **Pesquisa Operacional para Cursos de Engenharia**. [S.l.: s.n.], 2007. ISSN 16130073.
- BELFIORE, P.; YOSHIZAKI, H. T. Heuristic methods for the fleet size and mix vehicle routing problem with time windows and split deliveries. **Computers Industrial Engineering**, v. 64, n. 2, p. 589–601, 2013. ISSN 0360-8352. Disponível em: <<https://www.sciencedirect.com/science/article/pii/S0360835212002859>>.
- CAMPOS, G. G. D.; YOSHIZAKI, H. T. Y.; BELFIORE, P. P. Genetic algorithms and parallel computing for a vehicle routing problem with time windows and split deliveries; [algoritmos genéticos e computação paralela para problemas de roteirização de veículos com janelas de tempo e entregas fracionadas]. **Gestao e Producao**, v. 13, n. 2, p. 271 – 281, 2006. Cited by: 5; All Open Access, Gold Open Access, Green Open Access. Disponível em: <<https://www.scopus.com/inward/record.uri?eid=2-s2.0-33749315526&doi=10.1590%2fs0104-530x2006000200009&partnerID=40&md5=d7b5fd655f8c940babec0d6120b12cff>>.
- CHIANG, T. C.; HSU, W. H. A knowledge-based evolutionary algorithm for the multiobjective vehicle routing problem with time windows. **Computers and Operations Research**, v. 45, 2014. ISSN 03050548.
- CLARKE, G.; WRIGHT, J. W. Scheduling of vehicles from a central depot to a number of delivery points. **Operations Research**, v. 12, 1964. ISSN 0030-364X.
- DANTZIG, G. B.; RAMSER, J. H. The truck dispatching problem. **Management Science**, v. 6, 1959. ISSN 0025-1909.
- DROR, M.; LAPORTE, G.; TRUDEAU, P. Vehicle routing with split deliveries. **Discrete Applied Mathematics**, v. 50, n. 3, p. 239–254, 1994. ISSN 0166-218X. Disponível em: <<https://www.sciencedirect.com/science/article/pii/0166218X9200172I>>.
- DUTRA, Q. F. d. S.; RONCONI, D. P.; JUNQUEIRA, L. **Problema de roteirização de veículos com frota heterogênea fixa, janelas de tempo, entregas fracionadas e limitações de acesso**. Dissertação (Mestrado) — Universidade de São Paulo, 2023.
- GOLDEN, B.; ASSAD, A.; LEVY, L.; GHEYSENS, F. The fleet size and mix vehicle routing problem. **Computers & Operations Research**, v. 11, n. 1, p. 49–66, 1984. ISSN 0305-0548. Disponível em: <<https://www.sciencedirect.com/science/article/pii/0305054884900078>>.
- HOLLAND, J. H. **Adaptation in Natural and Artificial Systems**. Ann Arbor, MI: University of Michigan Press, 1975. ISBN 978-0472084609.

KOOL, W.; JUNINCK, J. O.; ROOS, E.; CORNELISSEN, K.; AGTERBERG, P.; HOORN, J. van; VISSER, T. Hybrid genetic search for the vehicle routing problem with time windows: a high-performance implementation. In: **ORTEC. 12th DIMACS Implementation Challenge, VRPTW Track, Phase 1**. [S.l.], 2022. Code source to be released.

KRITIKOS, M. N.; IOANNOU, G. The heterogeneous fleet vehicle routing problem with overloads and time windows. **International Journal of Production Economics**, v. 144, 2013. ISSN 09255273.

LAPORTE, G.; NOBERT, Y. Exact algorithms for the vehicle routing problem\*\*the authors are grateful to the canadian natural sciences and engineering research council (grants a4747 and a5486) and to the quebec government (fcac grant 80eq04228) for their financial support. In: MARTELLO, S.; LAPORTE, G.; MINOUX, M.; RIBEIRO, C. (Ed.). **Surveys in Combinatorial Optimization**. North-Holland, 1987, (North-Holland Mathematics Studies, v. 132). p. 147–184. Disponível em: <<https://www.sciencedirect.com/science/article/pii/S0304020808732353>>.

LIU, F. H.; SHEN, S. Y. The fleet size and mix vehicle routing problem with time windows. **Journal of the Operational Research Society**, v. 50, 1999. ISSN 14769360.

LÓPEZ-IBÁÑEZ, M.; DUBOIS-LACOSTE, J.; Pérez Cáceres, L.; BIRATTARI, M.; STÜTZLE, T. The irace package: Iterated racing for automatic algorithm configuration. **Operations Research Perspectives**, v. 3, p. 43–58, 2016. ISSN 2214-7160. Disponível em: <<https://www.sciencedirect.com/science/article/pii/S2214716015300270>>.

POTVIN, J. Y.; BENGIO, S. The vehicle routing problem with time windows part ii: Genetic search. **INFORMS Journal on Computing**, v. 8, 1996. ISSN 10919856.

QIN, T.; PENG, B.; BENLIC, U.; CHENG, T.; WANG, Y.; Lü, Z. Iterated local search based on multi-type perturbation for single-machine earliness/tardiness scheduling. **Computers Operations Research**, v. 61, p. 81–88, 2015. ISSN 0305-0548. Disponível em: <<https://www.sciencedirect.com/science/article/pii/S0305054815000635>>.

SCIPYDEVELOPERS. **Kolmogorov-Smirnov Test (kstest) Documentation**. 2024. Accessed: 2024-11-24. Disponível em: <<https://docs.scipy.org/doc/scipy/reference/generated/scipy.stats.kstest.html>>.

SOLOMON, M. M. Algorithms for the vehicle routing and scheduling problems with time windows constrains. **Operations Research**, v. 35, p. 254–265, 1987.

STEHLLING, T. M.; SOUZA, S. R. D. A comparison of crossover operators applied to the vehicle routing problem with time window. In: **Proceedings - 2017 Brazilian Conference on Intelligent Systems, BRACIS 2017**. [S.l.: s.n.], 2017. v. 2018-January.

VIDAL, T.; CRAINIC, T. G.; GENDREAU, M.; PRINS, C. Heuristics for multi-attribute vehicle routing problems: A survey and synthesis. **European Journal of Operational Research**, v. 231, 2013. ISSN 03772217.

WINSTON, W. L. Introduction to mathematical programming (fourth edition). **Mathematics and Computer Education**, v. 1, 2003. ISSN 978-0-534-35964-5.