

**UNIVERSIDADE FEDERAL DE SANTA CATARINA
DEPARTAMENTO DE INFORMÁTICA E ESTATÍSTICA**

Eduardo Borges Siqueira

**COMPARATIVO DE MÉTODOS PARA O PROBLEMA DE
ROTEAMENTO DE VEÍCULOS HETEROGÊNEOS COM FROTA
FIXA**

Florianópolis

2024

Eduardo Borges Siqueira

**COMPARATIVO DE MÉTODOS PARA O PROBLEMA DE
ROTEAMENTO DE VEÍCULOS HETEROGÊNEOS COM FROTA
FIXA**

Trabalho de conclusão de curso submetido
ao curso de Ciências da Computação para a
obtenção do Grau de Bacharel em Ciências
da Computação.

Orientador: Prof. Dr. Rafael de Santiago

Florianópolis

2024

Eduardo Borges Siqueira

COMPARATIVO DE MÉTODOS PARA O PROBLEMA DE ROTEAMENTO DE VEÍCULOS HETEROGÊNEOS COM FROTA FIXA

Trabalho de conclusão de curso submetido ao curso de Ciências da Computação para a obtenção do Grau de Bacharel em Ciências da Computação.

Comissão Examinadora

Prof. Dr. Rafael de Santiago
Universidade Federal de Santa Catarina
Orientador

Prof. Dr. Andrei de Almeida Sampaio Braga
Universidade Federal da Fronteira Sul

Prof. Dr. Pedro Belin Castellucci
Universidade Federal de Santa Catarina

Florianópolis

2024

RESUMO

Com o crescimento dos centros urbanos e a constante necessidade por produtos, o setor de transportes e logística clama cada vez mais por formas eficientes de fazer o manejo das rotas de seus veículos. Seja transportando passageiros, produtos, materiais ou até mesmo resíduos, planejar o trajeto de suas frotas torna-se uma tarefa complexa devido ao número de variáveis e condições à se atentar. Nesse contexto, os VRPs (*Vehicle Routing Problems* ou Problemas de Roteamento de Veículos) têm se mostrado ferramentas poderosas para, principalmente, reduzir custos logísticos e aumentar a eficiência operacional dos mais diversos sistemas, podendo também gerar impactos positivos em questões ambientais e de mobilidade urbana. Assim, o objetivo deste trabalho é avaliar e comparar soluções para uma variação mais detalhada dos VRPs, os HFFVRPs (*Heterogeneous Fixed Fleet Vehicle Routing Problems* ou Problemas de Roteamento de Veículos Heterogêneos com Frota Fixa). Com esta versão do problema, podemos nos aproximar um pouco mais dos cenários reais, trabalhando com frotas com tamanho estabelecido e com veículos que diferem entre si em aspectos como capacidade, custo operacional e eficiência, por exemplo. Visto isso, seguindo este caminho, foram realizadas revisões na literatura existente sobre do tema, bem como comparativos entre os resultados publicados, visando constatar como soluções computacionais podem atuar na redução dos custos nas etapas de transporte de cadeias logísticas. Como resultado, observamos como algoritmos genéticos demonstraram rapidez na obtenção de soluções e bom desempenho num geral, apresentando limitações somente no refinamento de seus custos finais. Por outro lado, o método baseado em geração de colunas evidenciou maior eficiência na otimização dos resultados, embora tenham exigido tempos de execução significativamente mais elevados.

Palavras-chave: roteamento de veículos, logística, análise de algoritmos

ABSTRACT

With the growth of urban centers and the constant demand for products, the transportation and logistics sector increasingly calls for efficient ways to manage vehicle routes. Whether transporting passengers, products, materials, or even waste, planning fleet routes becomes a complex task due to the number of variables and conditions to consider. In this context, VRPs (Vehicle Routing Problems) have proven to be powerful tools for primarily reducing logistical costs and increasing the operational efficiency of various systems, while also generating positive impacts on environmental issues and urban mobility. Therefore, the objective of this study is to evaluate and compare solutions for a more detailed variation of VRPs, the HFFVRPs (Heterogeneous Fixed Fleet Vehicle Routing Problems). With this version of the problem, we can get closer to real-world scenarios by working with fleets of a fixed size and vehicles that differ in aspects such as capacity, operational cost, and efficiency. To this end, reviews of the existing literature on the topic were conducted, as well as comparisons between published results, aiming to determine how computational solutions can reduce costs in the transportation stages of supply chains. As a result, we observed how genetic algorithms demonstrated speed in obtaining solutions and an overall good performance, presenting limitations only in refining their final costs. On the other hand, the column generation-based method showed greater efficiency in optimizing results, although it required significantly higher execution times.

Keywords: vehicle routing, logistics, algorithm analysis

SUMÁRIO

SUMÁRIO	7	
1	INTRODUÇÃO	9
1.1	Objetivos	10
1.1.1	Objetivo geral	10
1.1.2	Objetivos específicos	10
1.2	Metodologia	10
2	DEFINIÇÃO DO PROBLEMA	11
3	FUNDAMENTAÇÃO TEÓRICA	12
3.1	Grafos, Vizinhança e Caminhos	12
3.2	Heurísticas e Meta-heurísticas	13
3.3	Vehicle Routing Problem (Problema de Roteamento de Veículos)	14
3.4	Heterogeneous Fixed Fleet Vehicle Routing Problems (Problemas de Roteamento de Veículos Heterogêneos com Frota Fixa)	14
3.5	Algoritmos Genéticos	15
3.6	Geração de Colunas	16
4	TRABALHOS RELACIONADOS	17
4.1	Thirty years of heterogeneous vehicle routing - Laporte, Gilbert & Koc, cagri & Bektas, Tolga & Jabali, Ola	17
4.2	PyVRP: A High-Performance VRP Solver Package - Niels A. Wouda & Leon Lan & Wouter Kool	18
4.3	Metaheuristic Lesson: Genetic Algorithm - VRP (Vehicle Routing Problem) - Valdecy Pereira	20
4.4	VRPy: A Python package for solving a range of vehicle routing problems with a column generation approach - Romain Montagné and David Torres Sanchez and Halvard Olsen Storbugt	21
5	RESULTADOS	23
5.1	Métodos Analisados	23
5.2	Especificações dos Experimentos	24
5.3	Análise	27
6	CONCLUSÃO	32
6.1	Trabalhos Futuros	32

	REFERÊNCIAS	34
7	APÊNDICES	37
7.1	Introdução	40
7.2	Metodologia	41
7.3	Trabalhos Relacionados	41
7.4	Thirty years of heterogeneous vehicle routing - Laporte, Gilbert & Koc, cagri & Bektas, Tolga & Jabali, Ola	41
7.4.1	PyVRP: A High-Performance VRP Solver Package - Niels A. Wouda & Leon Lan & Wouter Kool	43
7.4.2	Metaheuristic Lesson: Genetic Algorithm - VRP (Vehicle Routing Problem) - Valdecy Pereira	45
7.4.3	VRPy: A Python package for solving a range of vehicle routing problems with a column generation approach - Romain Montagné and David Torres Sanchez and Halvard Olsen Storbugt	46
7.5	RESULTADOS	47
7.5.1	Métodos analisados	48
7.5.2	Especificações dos experimentos	49
7.5.3	Análise	51
7.6	CONCLUSÕES	55
7.6.1	Trabalhos futuros	56
	REFERÊNCIAS	57
8	ANEXOS	60
8.1	Código Fonte	60
8.2	Parâmetros Utilizados	60

1 INTRODUÇÃO

Problemas de roteamento de veículos são desafios constantes em toda a cadeia logística ao redor do mundo. Seja para transportar matéria-prima para as indústrias, materiais para fábricas, produtos para lojas ou para os clientes, o transporte de cargas se mostra fundamental para a existência de qualquer rede. Com isso, surgindo como alternativa para otimizar as frotas, os custos com transporte, o tempo de deslocamento e até mesmo reduzir os impactos ambientais, os VRPs (*Vehicle Routing Problems* ou Problemas de Roteamento de Veículos) vêm ganhando bastante destaque nos ramos de pesquisa nesta área.

Em grandes economias, como o caso da União Europeia, o setor de transportes chega a representar mais de 10% do produto interno bruto do bloco (HOFF et al., 2010). Mesmo sendo um valor expressivo, Hoff et al. (2010) ainda afirma que devido ao crescimento econômico, crescimento do consumo e da globalização tendem a aumentar esse valor, assim como o aumento da competição entre as empresas do setor logístico e os proprietários dos bens a serem transportados tendem também a criar altas demandas por sistemas mais complexos e eficientes.

Com isso, existem diversas variações para esse problema, dado que cada variação aumenta consideravelmente sua complexidade por se aproximar mais da realidade, adicionando fatores como janelas de tempo nas rotas, possibilidade de carga e descarga nos pontos de parada, entre outros (LAPORTE et al., 2015). Dentre eles, destaca-se uma comum situação da realidade, objeto de pesquisa desta monografia, o problema de roteamento de veículos heterogêneos com frota fixa, ou seja, trabalha-se com um número fixo de veículos, diferentes entre si.

Brandao (2011) mostrou que o problema de HFFVRP é um problema NP-Difícil. Além disso, o mesmo comenta que, quando comparado a problemas de natureza semelhante, como por exemplo FSMVRP (*Fleet Size and Mix Vehicle Routing Problems* - Problemas de Roteamento de Veículos com Frota Mista), onde se tem veículos a vontade, se mostra consideravelmente mais complexo, colocando-o entre os mais difíceis de serem resolvidos.

Nesse contexto, é possível observar um conflito de interesses entre as partes envolvidas na cadeia logística. Por um lado, temos clientes que desejam seus bens entregues o mais rápido possível, resultando num tratamento exclusivo e direto, sem passar por outros pontos. Do outro, temos operadores logísticos que desejam reduzir gastos o máximo possível e diminuir a distância percorrida por seus veículos, buscando uma maior eficiência na entregas.

Com isso, esse trabalho analisa e compara soluções presentes na literatura que se propõem a resolver o problema de roteamento de veículos heterogêneos com frota fixa, buscando como objetivo reduzir tanto os custos fixos do transporte, como o custo por rota realizada por

cada veículo, como os custos variáveis, relacionados principalmente a distância a ser percorrida (para este trabalho, pois na literatura, como aponta Laporte et al. (2015), podemos encontrar diversas variações com outros custos variáveis).

1.1 Objetivos

Esta seção apresenta o objetivo geral e os objetivos específicos deste trabalho.

1.1.1 Objetivo geral

Identificar e comparar métodos para o problema de roteamento de veículos heterogêneos com frota fixa (*Heterogeneous Fixed Fleet Vehicle Routing Problem* - HFFVRP).

1.1.2 Objetivos específicos

Para atingir o objetivo geral, os seguintes objetivos específicos foram definidos:

- selecionar métodos para o HFFVRP;
- comparar métodos para o HFFVRP;
- dar publicidade aos resultados do trabalho.

1.2 Metodologia

Este trabalho caracteriza uma pesquisa qualitativa e exploratória, visando analisar e comparar métodos propostos para o problema de roteamento de veículos heterogêneos com frota fixa, apresentando os resultados analisados.

Para o desenvolvimento da pesquisa, são realizadas as seguintes etapas:

Primeiramente, é feito um levantamento teórico do estado da arte acerca do HFFVRP, por meio de uma pesquisa bibliográfica utilizando a literatura disponível.

Após a primeira etapa, são selecionados na literatura métodos possíveis para resolver o problema.

Em seguida, os métodos são devidamente implementados e analisados, seguindo corretamente o que foi descrito nos trabalhos.

Por fim, os resultados são comparados e discutidos, baseando-se na função objetivo dos mesmos.

2 DEFINIÇÃO DO PROBLEMA

Como este trabalho se limita à análise e comparação de métodos publicados por outros autores, a definição do problema será a fornecida pelos mesmos.

Para isso, ao descrever um sistema de roteamento de veículos, utilizamos uma rede de distribuição, com um único depósito e um ou mais clientes (LAPORTE et al., 2015). Assim, podemos definir que esta se define como um grafo não-direcionado $G = (V, E)$, onde $V = \{0, 1, \dots, n\}$ é o conjunto de vértices, representando os pontos de passagem dos veículos (depósito e clientes), $V_0 = V \setminus \{0\}$ somente os clientes já que o elemento 0 de V representa o depósito, $E = \{(i, j) : i, j \in V\}$ é o conjunto de todas as arestas, representando as rotas entre eles. A cada aresta, por sua vez, é atribuído um peso d_{ij} ($d_{ii} = 0, \forall i \in V$) referente a distância a ser percorrida. Uma frota de T diferentes veículos é informada, onde o número de veículos é n_k ($k = 1, \dots, T$). Uma capacidade Q_k e um custo variável v_k (TAILLARD, 1999) são associados a cada tipo de veículo k ($k = 1, \dots, T$). Segundo Laporte et al. (2015), uma variável binária de controle é adicionada para saber se um veículo já atendeu um cliente, denotada por x_{ij}^k definida como 1 se e somente se um veículo k passou por uma aresta $(i, j) \in E$. Por fim, um custo de viagem para cada aresta $(i, j) \in E$ por um veículo k é dado por $c_{ij} = v_k d_{ij}$ (BRANDAO, 2011).

Precisamos impor restrições para garantir a corretude do problema. São elas:

- o grafo é um grafo conexo
- a frota de veículos tem tamanho fixo
- não serão utilizadas outras variáveis além das definidas
- todas as demandas dos clientes serão satisfeitas
- cada cliente será visitado uma única vez
- as rotas dos veículos começam e terminam no depósito ($V = 0$)
- rotas não podem carecer nem exceder, respectivamente, os números mínimos e máximos de clientes

Para a avaliação e comparação dos resultados obtidos, utiliza-se uma função objetivo cujo valor é o mínimo da soma entre os custos fixos e variáveis dos veículos, semelhante a Baldacci, Battarra and Vigo (2008), conforme apresentada a seguir:

$$\text{Mínimo} \sum_{k \in T} \sum_{j \in V_0} x_{0j}^k + \sum_{k \in T} \sum_{(i,j) \in E} v_k d_{ij} x_{ij}^k$$

3 FUNDAMENTAÇÃO TEÓRICA

Neste capítulo, abordaremos os principais termos e conceitos necessários para que o leitor possa compreender este trabalho como um todo, devido ao seu nível de especificidade. Aqui, será apresentado sobre os principais conceitos de teoria de grafos, vizinhança, caminhos, heurísticas, meta-heurísticas, problema de roteamento de veículos e problema de roteamento de veículos heterogêneos com frota fixa.

3.1 Grafos, Vizinhança e Caminhos

Wilson (1996) define um grafo como um diagrama, composto por um conjunto de pontos, chamados de vértices, e por um conjunto de linhas, chamadas de arestas. Referente ao conjunto vértices, segundo a definição de Wilson (1996), é finito e não vazio. Quanto as arestas, denotamos-a como sendo $\{u, v\}$, onde esta representa a conexão entre os vértices u e v . Se estes mesmos u e v estão conectados por uma aresta, dizemos que são adjacentes. Se um vértice não pode ser alcançado por arestas, dizemos que ele é isolado.

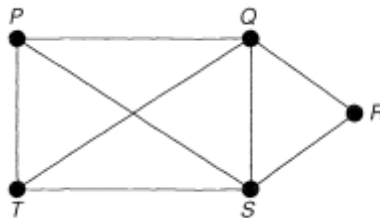


Figura 1 – Exemplo de grafo por Wilson (1996), onde P, Q, R, S, T representam os vértices do grafo e as linhas suas arestas

Quanto as propriedades de um grafo, dizemos que ele é desconexo se não pode ser expressado como a união de dois grafos, e conexo caso possa ser (WILSON, 1996). Sobre a igualdade de grafos, dizemos que dois grafos são iguais se possuem exatamente os mesmos conjuntos de vértices e arestas.

O termo vizinhança se aplica quando, num grafo $G = (V, E)$, um vértice $v \in V$ tem adjacente a ele um conjunto de vértices X , sub-conjunto de V , dizemos que esse conjunto X são os vizinhos de v .

Por fim, segundo Wilson (1996), um caminho é caracterizado por uma sequência finita de arestas adjacentes, ligando um vértice inicial v_0 até um vértice final v_n , caracterizando um caminho de v_0 a v_n .

3.2 Heurísticas e Meta-heurísticas

Para trabalhar esses assuntos, vale introduzir alguns conceitos importantes. O primeiro, solução ótima, representa a melhor solução para um problema, obtida através de métodos exatos, como algoritmos, otimizando a função objetivo (BRANDAO, 2011). Além disso, ótimos locais e globais merecem atenção também. Ótimo local pode ser definido como a melhor solução dentro de uma vizinhança, ou seja, dentro de um subconjunto de soluções adjacentes a ela. Já o ótimo global é a melhor solução dentro de todas as possíveis (ROTHLAUF, 2011).

Assim, heurísticas podem ser definidas como métodos de otimização sem garantia para encontrar uma solução exata (ROTHLAUF, 2011). Essa forma de resolução é baseada em regras e informações acerca do problema, baseando-se na experiência ou no julgamento sobre o que se deseja resolver (ROTHLAUF, 2011).

Segundo Rothlauf (2011), existem dois grandes tipos de heurísticas que merecem destaque:

- Heurísticas construtivas: Focadas em produzir uma solução passo a passo, iterativamente
- Heurísticas de melhoria: Iniciam já com uma solução completa e iterativamente passam pelo espaço de busca atrás de melhores resultados (muito comuns no escopo deste trabalho).

Já meta-heurísticas surgem como uma forma de otimizar a solução de problemas mais complexos no qual não se consegue obter uma solução ótima, como o caso do HFFVRP, explorando soluções de maneiras mais eficazes que heurísticas dentro do espaço de busca (BLUM; ROLI, 2001).

Seu uso aparece principalmente em problemas de natureza combinatória, como problemas estocásticos, onde normalmente devida a complexidade do problema não é possível encontrar uma solução ótima (BRANDAO, 2011). Normalmente, essa forma de busca se torna a escolhida para resolução de problemas de otimização por conta de sua capacidade de explorar eficientemente grandes espaços de busca.

Algumas estratégias para meta-heurísticas podem ser adotadas afim de se encontrar melhores soluções. A diversificação, por exemplo, incentiva a exploração de regiões inexploradas dentro do espaço de busca, ajudando a busca a não se concentrar apenas em uma área específica, auxiliando a busca de novas soluções. A intensificação, em contra partida, busca focar em áreas que obtiveram resultados promissores, ou seja, resultados boas soluções pra o problema. A implementação destas estratégias pode ser dada com introdução de um sistema de incentivos e penalidades (GLOVER; LAGUNA, 1999). Como demonstrado no trabalho de Brandao (2011), o equilíbrio dessas métricas se mostra crucial para o desempenho das meta-heurísticas.

3.3 Vehicle Routing Problem (Problema de Roteamento de Veículos)

O problema clássico de roteamento de veículos, introduzido por Dantzig and Ramser (1959), surgiu como uma maneira de determinar uma rota otimizada para uma frota de caminhões iguais, com o objetivo de realizar entregas de combustível para clientes baseados em suas demandas. Para este cenário, os caminhões também devem começar e terminar no depósito, os clientes só podem ser visitados uma vez e suas demandas devem ser satisfeitas (DANTZIG; RAMSER, 1959).

Nessa primeira aparição, uma solução para o problema foi apresentada por Dantzig and Ramser (1959). Nela, um procedimento baseado em uma formulação de programação linear foi apresentado, afim de obter algo próximo a uma solução ótima. Devido a época que o trabalho foi publicado, os autores demonstram manualmente os cálculos, mas afirmam que os mesmos podem ser realizados em uma "máquina digital de computação".

As funções objetivo para esse problema podem variar bastante, mas sempre girando em torno de minimizações. Entre as mais comuns, estão:

- Minimizar os custos com os veículos
- Minimizar o tempo de deslocamento
- Minimizar a distância percorrida pelos caminhões
- Minimizar o uso dos veículos

A partir desta publicação, diversos autores se apoiaram em seus sólidos conceitos para elaborar ainda mais o tema, afim de aproxima-lo mais da realidade. Para isso, foi necessário criar variações do problema com novas variáveis a serem analisadas, mas sempre mantendo as raízes no VRP (LAPORTE et al., 2015). Destes, podemos destacar as contribuições de Golden et al. (1984) assumindo frotas de tamanho infinito e, pela primeira vez, veículos diferentes. Além dele, Taillard (1999) traz também uma variação, tema desta monografia, com frotas diferentes e de tamanho definido.

3.4 Heterogeneous Fixed Fleet Vehicle Routing Problems (Problemas de Roteamento de Veículos Heterogêneos com Frota Fixa)

Apresentado pela primeira vez por Taillard (1999), o problema de roteamento de veículos heterogêneos com frota fixa é dado como uma especificação do problema de roteamento de veículos, pois trata-se de um caso específico desse problema. Aqui, como o próprio nome induz, tenta-se aproximar a cenários mais comuns do dia a dia da população, adicionando novos fatores ao sistema.

Um desses fatores é a heterogeneidade do sistema. Assim, como na vida real, o sistema possui veículos com capacidades diferentes entre si. Além da capacidade, fatores como custos fixos (LI; GOLDEN; WASIL, 2007) e custos variáveis (TAILLARD, 1999) podem ser adicionados para representar o preço pago pelo deslocamento do veículo, como custos com combustíveis e desgastes no mesmo.

Além da frota diferente, a frota ser fixa também aparece neste problema. Isso significa, em outras palavras, que o número de veículos é limitado e, como solução válida ao problema, a resposta pode ser adicionar novos veículos. Como menciona Brandao (2011) e Laporte et al. (2015), o fato de se ter um número limitado de veículos dificulta muito o processo de se encontrar uma solução, visto que as opções são limitadas, quando comparado principalmente ao FSMVRP (*Fleet Size and Mix Vehicle Routing Problems* - Problemas de Roteamento de Veículos com Frota Mista), proposto por Golden et al. (1984), que assume um número infinito de veículos, simulando assim um cenário onde os veículos ainda serão adquiridos.

Dessa maneira, o problema de HFFVRP atua também como um auxiliador na tomada de decisão (BRANDAO, 2011), visto que trabalha de maneira a assumir que a frota já está adquirida e que o problema precisa ser resolvido com a mesma.

3.5 Algoritmos Genéticos

Algoritmos genéticos são métodos de otimização inspirados no processo de seleção natural e evolução biológica, comumente utilizados em problemas onde a busca pela solução ótima é complexa. Esses métodos utilizam uma população de soluções possíveis, as quais sofrem operações inspiradas na genética, como seleção, cruzamento e mutação, para evoluir em direção a soluções melhores ao longo de várias gerações (HOLLAND, 1975).

A estrutura básica dos algoritmos genéticos envolve a criação de uma população inicial, que são avaliadas segundo uma função objetivo, ou *fitness function*, que mede a qualidade de cada solução. Essa população inicial é então submetida a uma série de operações genéticas que visam produzir novas gerações com soluções potencialmente melhores. Segundo Goldberg (1989) a eficiência dos algoritmos genéticos se deve à sua capacidade de combinar informações de várias soluções, permitindo uma exploração eficiente do espaço de busca.

Mitchell (1998) enfatiza a importância da diversidade na população de soluções para evitar a convergência prematura para ótimos locais. A diversidade é mantida, em grande parte, pela operação de mutação, que introduz variações nas soluções, garantindo que a busca continue explorando diferentes regiões do espaço de busca.

Assim, os algoritmos genéticos são classificados como meta-heurísticas, visto que não garantem uma solução ótima exata, mas são eficazes para resolver problemas complexos de otimização combinatória.

3.6 Geração de Colunas

A técnica de geração de colunas, amplamente utilizada em problemas de otimização combinatória, tem se destacado como uma abordagem eficiente para resolver problemas de roteamento de veículos (VRPs). Introduzida inicialmente por Dantzig and Wolfe (1954) no contexto do problema de transporte, a geração de colunas é baseada na formulação do problema mestre restrito e na solução iterativa de subproblemas para identificar colunas (neste trabalho, rotas) que possam melhorar a solução atual.

No contexto de VRPs, essa técnica é aplicada utilizando uma formulação baseada em rotas, onde o problema mestre é uma relaxação linear que aloca os clientes a rotas pré-definidas, enquanto o subproblema, frequentemente modelado como um problema de caminho mínimo com restrições, busca gerar novas rotas viáveis que reduzam o custo total da solução (DESAULNIERS; DESROSIERS; SOLOMON, 2005).

A eficiência da geração de colunas em VRPs deve-se à sua capacidade de lidar com a complexidade combinatória. Por exemplo, Desaulniers, Desrosiers and Solomon (2005) detalham como a técnica pode ser adaptada para variantes com restrições específicas, como múltiplos depósitos ou janelas de tempo, utilizando modelos de programação linear inteira para o problema mestre e algoritmos especializados, como programação dinâmica, para resolver o subproblema de precificação.

Outro avanço significativo foi apresentado por Costa, Olivera and Toth (2019), que exploraram o uso de heurísticas híbridas em conjunto com a geração de colunas, como estratégias de mergulho para acelerar a convergência do algoritmo. Eles destacam que a flexibilidade da geração de colunas permite a incorporação de diferentes restrições e objetivos ao modelo, tornando-a uma abordagem robusta e adaptável para uma ampla gama de aplicações logísticas.

Apesar de sua eficácia, a técnica de geração de colunas enfrenta desafios em cenários de grande escala devido ao alto custo computacional associado à resolução iterativa do problema de precificação. Para mitigar esse problema, abordagens modernas têm integrado técnicas heurísticas e metaheurísticas, como algoritmos genéticos e busca tabu, para resolver o subproblema de forma mais eficiente, como discutido por Sadykov et al. (2019b). Essas melhorias têm ampliado a aplicabilidade da geração de colunas, consolidando-a como uma ferramenta essencial na solução de problemas de roteamento de veículos e outras áreas de pesquisa operacional.

Dessa forma, a geração de colunas continua sendo uma técnica promissora para a resolução de VRPs, contribuindo para a obtenção de soluções de alta qualidade em cenários complexos e realistas, combinando rigor matemático e flexibilidade prática.

4 TRABALHOS RELACIONADOS

A busca iniciou-se através do site Periódicos CAPES, buscando trabalhos que possuíssem as palavras chave "HFFVRP" ou "Heterogeneous Fixed Fleet Vehicle Routing Problem", obtendo assim 138 resultados. Deste grupo, foram selecionados três artigos com propostas de solução ao tema. O primeiro, propõe uma análise histórica dos problemas de roteamento de veículos heterogêneos (*Heterogeneous Vehicle Problems - HVRPs*) (LAPORTE et al., 2015), outro propõe uma busca tabu como solução do problema (BRANDAO, 2011) e, por fim, um que implementa uma solução baseada em uma heurística de limite de aceitação (*threshold accepting heuristic*) (TARANTILIS; KIRANOUDIS; VASSILIADIS, 2003).

A partir do trabalho de Laporte et al. (2015) foi possível obter um panorama ampliado do contexto histórico das pesquisas e do desenvolvimento na área de roteamento de veículos heterogêneos. Através dele, conseguiu-se acesso ao trabalho de Tarantilis and Kiranoudis (2001), um dos pioneiros em pesquisa nesse segmento de HFFVRP. Por fim, o trabalho de Brandao (2011) traz uma visão mais abrangente do problema em si, pontuando de maneira mais clara questões teóricas da literatura para esse problema.

4.1 Thirty years of heterogeneous vehicle routing - Laporte, Gilbert & Koc, cagri & Bektas, Tolga & Jabali, Ola

Em Laporte et al. (2015) os autores retratam sobre o contexto histórico dos problemas de roteamento de veículos heterogêneos (HVRPs), montando assim um aglomerado de informações sobre essa classe de problemas e suas mais tradicionais variações.

Os autores então denotam as duas classes mais importantes de problemas de roteamento de veículos com características diferentes, sendo elas FSMVRPs (*Fleet Size and Mix Vehicle Routing Problems - Problemas de Roteamento de Veículos com Frota Mista*) (GOLDEN et al., 1984) e HFFVRPs (*Heterogeneous Fixed Fleet Vehicle Routing Problems* ou Problema de Roteamento de Veículos Heterogêneos com Frota Fixa) (TAILLARD, 1999). Segundo os autores, a diferença entre elas se dá por conta do roteamento com frota mista levar em consideração que a frota é infinita, ou seja, têm-se quantos veículos quiser. Já no problema de frota fixa, assume-se um conjunto limitado de opções de veículos.

Dentro das classe informadas, Laporte et al. (2015) apresenta cinco principais variantes, sendo elas:

- FSMVRPs com custos de veículos fixos e variáveis (FERLAND; MICHELON, 1988);

- FSMVRPs somente com custos fixos de veículos (GOLDEN et al., 1984);
- FSMVRPs somente com custos variáveis de veículos (TAILLARD, 1999);
- HFFVRPs com custos de veículos fixos e variáveis (LI; GOLDEN; WASIL, 2007);
- HFFVRPs somente com custos variáveis de veículos (TAILLARD, 1999);

São tratadas ainda outras variantes, que ajudam o problema a ficar mais próximo da realidade. Dentre elas, temos as janelas de tempo (TARANTILIS et al., 2007), múltiplos depósitos (DONDO; CERDÁ, 2007), coletas e entregas (IRNICH, 2000), sobrecargas (KRITIKOS; IOANNOU, 2013), entre diversas outras catalogadas pelos autores.

Comentado sobre as variações presentes na literatura, os autores descrevem mais detalhadamente sobre as mesmas, explicando com maior profundidade suas peculiaridades. Além disso, mencionam trinta e duas soluções na literatura especificamente para o problema de roteamento de veículos heterogêneos com frota fixa, explicando se as mesmas atacam problemas variantes apresentados, se possuem modelo matemático de solução, a maneira com que solucionaram o problema e se faz parte do estudo de caso realizado por eles.

Após o breve estudo de caso, onde apresentam poucos resultados para debater sobre o assunto, concluem que a área é promissora e que possui apenas soluções que se apresentaram sub-ótimas para as cinco principais variantes apresentadas até o momento, visto a complexidade do problema. Uma crítica mencionada por Laporte et al. (2015) é de que o número de trabalhos com altas abstrações é muito grande, fazendo com que os resultados sejam distantes da realidade.

4.2 PyVRP: A High-Performance VRP Solver Package - Niels A. Wouda & Leon Lan & Wouter Kool

Algoritmos genéticos são amplamente utilizados na resolução de problemas de otimização complexos, como o problema de roteamento de veículos (VRP). Estes algoritmos se destacam por seu método de busca, inspirado na evolução biológica, que envolve operações como seleção, cruzamento e mutação para evoluir uma população de soluções em direção a soluções de melhor qualidade ao longo de várias gerações (HOLLAND, 1975). O modelo proposto em PyVRP utiliza o método de busca genética híbrida *Hybrid Genetic Search*, *HGS*, que combina algoritmos genéticos com busca local, demonstrando resultados de ponta para VRPs capacitados (CVRP) como um todo (VIDAL et al., 2013).

Wouda, Lan and Kool (2024) destacam que, em PyVRP, apenas as partes críticas em desempenho são implementadas em C++, enquanto o restante é programado em Python, o que

facilita a personalização e o uso do seu modelo. Esse design modular permite aos pesquisadores personalizar aspectos como o gerenciamento da população e estratégias de cruzamento, ampliando as possibilidades de aplicação de HGS a diferentes variantes de VRPs.

Os métodos genéticos foram especialmente projetados para equilibrar qualidade e diversidade das soluções, evitando a convergência para ótimos locais. Essa abordagem segue as diretrizes de Vidal et al. (2013) sobre o uso de diversificação e intensificação para melhorar a performance na resolução de VRPs. Por exemplo, este modelo incorpora um gerenciador de penalidades que ajusta dinamicamente os pesos das penalidades para garantir que um percentual específico de buscas locais gere soluções viáveis, uma abordagem que sustenta a eficiência do algoritmo ao explorar e intensificar o espaço de soluções (VIDAL et al., 2013).

Como exemplo de solução utilizando esse método, temos o gráfico exibido na figura 2.

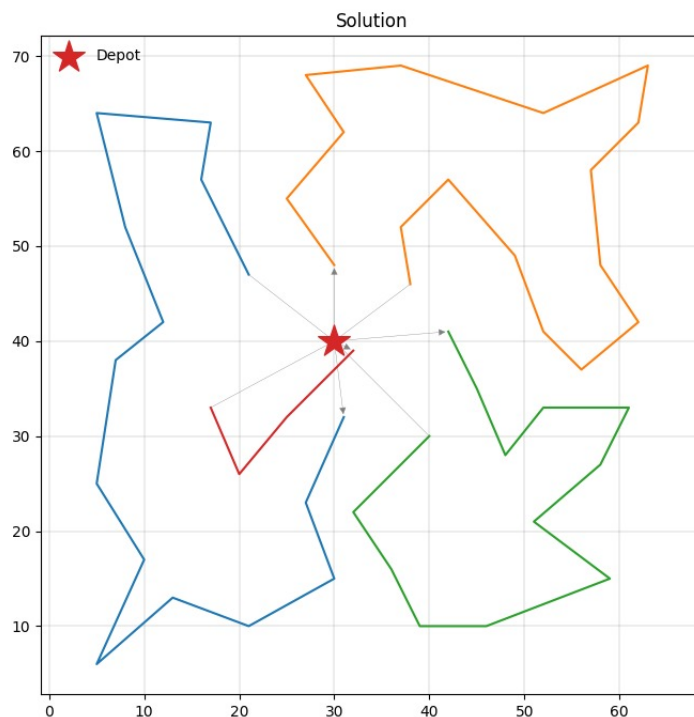


Figura 2 – Exemplo de saída do método proposto por Wouda, Lan and Kool (2024)

4.3 Metaheuristic Lesson: Genetic Algorithm - VRP (Vehicle Routing Problem) - Valdecy Pereira

Pereira (2021) explora extensivamente a aplicação de algoritmos genéticos em VRPs, abordando conceitos como seleção, cruzamento, mutação e elitismo, que são fundamentais para o sucesso dessa meta-heurística na busca de soluções de alta qualidade.

De acordo com Pereira (2021), os algoritmos genéticos funcionam por meio da evolução de uma população inicial de soluções, avaliadas por uma função de *fitness* que determina sua adequação para o problema. Em cada geração, soluções mais “aptas” são selecionadas para reprodução, onde operações de cruzamento combinam informações de diferentes soluções para gerar descendentes com potencial para desempenho superior. A mutação, por outro lado, é aplicada a uma pequena fração das soluções, inserindo variação genética e prevenindo que o algoritmo se acomode em ótimos locais precocemente. Essa diversidade genética é essencial para garantir que o algoritmo explore novas áreas do espaço de busca, mantendo o equilíbrio entre exploração e intensificação.

O autor também destaca a implementação de elitismo, onde as melhores soluções de uma geração são preservadas para a próxima. Essa prática, comum em algoritmos genéticos, garante que a qualidade das soluções não seja comprometida ao longo das gerações, permitindo uma evolução estável e eficaz para encontrar soluções próximas do ótimo global.

Além das operações básicas, o trabalho discute a personalização dos parâmetros do algoritmo, como a taxa de mutação e o tamanho da população, para adaptar o GA a diferentes variantes de VRPs, como o VRP Capacitado (CVRP), o VRP com múltiplos depósitos (MD-VRP) e o VRP com janelas de tempo (VRPTW). Esse nível de flexibilidade é uma das razões pelas quais os algoritmos genéticos são amplamente aplicados em problemas de roteamento de veículos, pois permitem ajustar a busca conforme as necessidades específicas de cada problema.

Este trabalho, assim como o de outros autores renomados na área, como os de Holland (1975) e Goldberg (1989), contribui significativamente para o desenvolvimento de metaheurísticas que combinam eficiência e adaptabilidade, estabelecendo os algoritmos genéticos como uma das principais abordagens para a resolução de VRPs e suas variações.

Por fim, o resultado final de execução do modelo proposto por Pereira (2021) pode ser na figura 3.

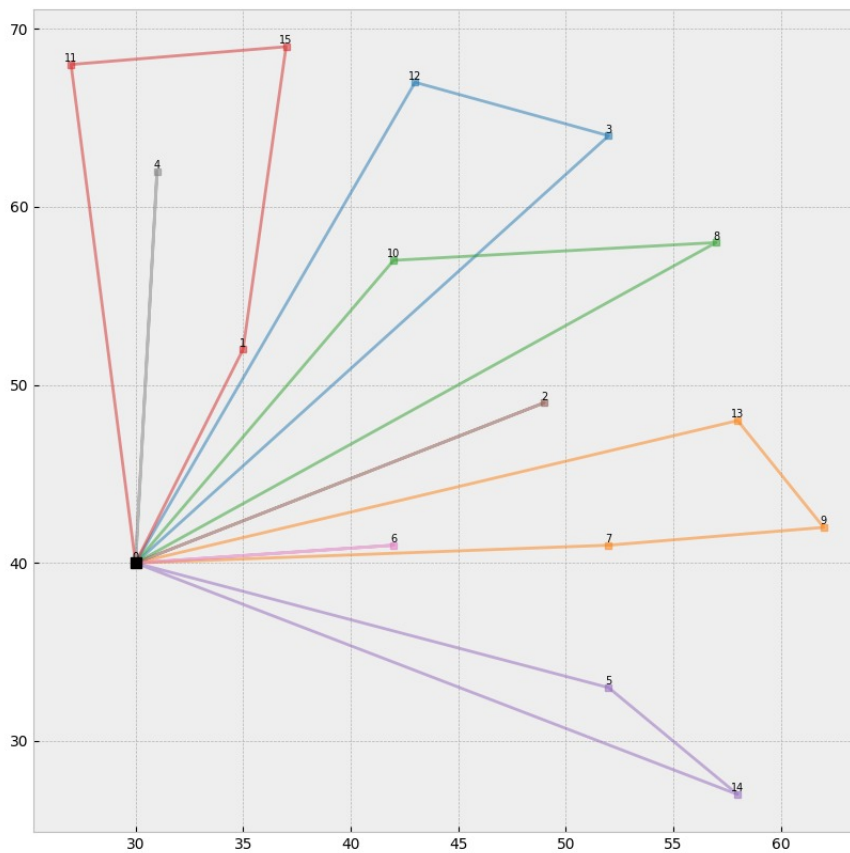


Figura 3 – Exemplo de saída do método proposto por Pereira (2021)

4.4 VRPy: A Python package for solving a range of vehicle routing problems with a column generation approach - Romain Montagné and David Torres Sanchez and Halvard Olsen Storbugt

O método proposto em VRPy oferece uma solução flexível para problemas de VRPs utilizando uma abordagem de geração de colunas. Esse método permite resolver diversas variantes do VRP, como o VRP Capacitado (CVRP), VRP com janelas de tempo (VRPTW) e VRP com coleta e entrega simultânea. O método foi desenvolvido com o objetivo de fornecer uma API intuitiva baseada na biblioteca NetworkX, facilitando a aplicação para pesquisadores e analistas de logística (MONTAGNÉ; SANCHEZ; STORBUGT, 2020).

A abordagem de geração de colunas utilizada envolve a resolução de dois subproblemas principais: o problema mestre e o problema de precificação. O problema mestre é uma formulação linear que particiona os clientes em subconjuntos ou rotas, enquanto o problema de

precificação consiste em encontrar novas rotas viáveis que possam melhorar a solução atual. Essa metodologia permite buscar soluções de alta qualidade, adaptando-se a diferentes restrições e requisitos específicos de cada variante de VRP (COSTA; CONTARDO; DESAULNIERS, 2019).

Montagné, Sanchez and Storbugt (2020) destacam a capacidade de VRPy de competir com outras ferramentas populares, como o OR-Tools do Google, em termos de qualidade de solução e facilidade de uso. A biblioteca foi projetada para resolver problemas de pequena a média escala, o que a torna uma escolha ideal para aplicações industriais e acadêmicas onde é necessária acessível. Além disso, a VRPy permite o uso de estratégias avançadas, como heurísticas de mergulho (SADYKOV et al., 2019a) e hiper-heurísticas para seleção dinâmica de estratégias de precificação, que ajustam automaticamente as configurações de parâmetros para otimizar o desempenho do algoritmo durante a execução.

O método segue evoluindo, com futuras melhorias planejadas para incrementar o desempenho e adicionar novas funcionalidades, consolidando-se como uma solução robusta para a comunidade de pesquisa operacional (MONTAGNÉ; SANCHEZ; STORBUGT, 2020).

5 RESULTADOS

Para comparar a eficiência dos modelos, foram selecionadas três métricas principais: custo na primeira solução, custo na melhor solução e tempo para encontrar a melhor solução. Essas métricas foram escolhidas por sua relevância em problemas de roteamento de veículos, uma vez que ajudam a avaliar tanto a qualidade inicial das soluções quanto a eficácia da otimização ao longo do processo de busca.

A métrica de custo na primeira solução é particularmente importante para avaliar a eficácia inicial dos algoritmos. Em problemas complexos uma boa primeira solução pode ser decisiva, especialmente em aplicações práticas onde o tempo é limitado e soluções rápidas são necessárias. No contexto de VRPy, Montagné, Sanchez and Storbugt (2020) destacam que uma solução inicial próxima ao ótimo pode facilitar o refinamento posterior através da geração de colunas. Comparativamente, os métodos de Wouda, Lan and Kool (2024) e Pereira (2021), que utilizam estratégias baseadas em algoritmos genéticos, buscam gerar uma população inicial de soluções variadas e de qualidade aceitável, facilitando o progresso da otimização.

A segunda métrica, custo na melhor solução, é essencial para avaliar a capacidade dos algoritmos de melhorar a qualidade das soluções ao longo do tempo. A busca pela melhor solução reflete o sucesso da metodologia em explorar o espaço de busca e evitar a convergência para ótimos locais. Esta métrica permite observar até que ponto os métodos conseguem refinar a solução inicial e se aproximar do ótimo global. Essa métrica é especialmente útil para avaliar o VRPy, que aprimora iterativamente a solução.

Por fim, o tempo para encontrar a melhor solução é uma métrica crítica em problemas desta natureza, devido às restrições de tempo em aplicações reais. Esta métrica mede a eficiência de cada método, observando quanto tempo cada algoritmo precisa para alcançar a sua solução de melhor qualidade possível. A avaliação do tempo é fundamental para determinar a aplicabilidade prática de cada modelo, especialmente em contextos onde o tempo de resposta é um fator decisivo.

5.1 Métodos Analisados

A escolha dos três modelos analisados neste trabalho – VRPy, pyVRP e pyVRP(Valdecy) – foi motivada pelas características específicas de cada um, que se mostraram promissoras para resolver o problema escolhido.

O VRPy foi selecionado por sua abordagem de geração de colunas, que é amplamente reconhecida por sua eficácia em resolver problemas de roteamento, especialmente para variantes do VRP com restrições de capacidade e janelas de tempo. A flexibilidade do VRPy e seu suporte

para diferentes restrições de rota tornaram-o uma escolha adequada para análise em HFFVRPs.

O PyVRP, que implementa o método de busca genética híbrida (HGS), foi escolhido por sua capacidade de combinar a exploração global dos algoritmos genéticos com uma busca local intensiva. Essa combinação é particularmente útil para VRPs complexos e de grande escala, onde a flexibilidade para adaptar a busca é essencial. Além disso, o PyVRP permite personalização no gerenciamento da população e nas estratégias de cruzamento, facilitando o ajuste do modelo às necessidades do HFFVRP.

A escolha do Algoritmo Genético para VRP se deu por sua natureza adaptativa e sua capacidade de explorar vastos espaços de soluções, mantendo a diversidade populacional e evitando convergência prematura. Essa característica é fundamental para o HFFVRP, onde as configurações de frotas e as restrições variam amplamente. Além disso, os algoritmos genéticos são conhecidos por sua versatilidade e facilidade de adaptação, o que facilita a análise de resultados sob diferentes configurações e instâncias.

Outros métodos também foram testados. Um método baseado em aprendizado por reforço profundo, apresentado por Li et al. (2022) foi testado, porém o projeto ao qual tive acesso apresentava diversos erros e incompatibilidades, fazendo assim com que ele fosse eliminado do comparativo. Um segundo método baseado em uma rede de ponteiros em matriz 2D, descrito por Liu et al. (2024) também foi testado. Esse apresentou resultados muito incoerentes e fora do esperado para o projeto.

Por fim, a biblioteca para resolução de VRPs denominada VRPSolverEasy, de autoria de Errami et al. (2023) também foi descartada. Esta por sua vez apresentou diversos problemas para compilação e necessitava de um *solver* próprio de autoria dos mesmos, desmotivando assim seu uso neste trabalho.

Além destas considerações, vale destacar que a obtenção de acesso à códigos para resolver HFFVRPs na literatura é um desafio. Muitos métodos desenvolvidos não tem seus códigos fontes disponibilizados publicamente, o que limita a replicabilidade e análise comparativa entre diferentes abordagens. Como destacado por Wouda, Lan and Kool (2024), o PyVRP foi criado com a finalidade de preencher essa lacuna, sendo um dos poucos pacotes de código aberto que oferece suporte abrangente para VRPs no geral. Entretanto, a ausência de outros códigos prontos na literatura dificulta a comparação direta entre diferentes implementações de HFFVRPs, exigindo que pesquisadores implementem suas próprias versões de métodos complexos para realizar análises robustas.

5.2 Especificações dos Experimentos

Como apresentado anteriormente, métodos como Montagné, Sanchez and Storbugt (2020) e Wouda, Lan and Kool (2024), por exemplo, possuem diferenças consideráveis entre si, re-

sultando em abordagens diferentes na elaboração dos programas. Apesar das distinções, os programas para os métodos escolhidos foram desenvolvidos conforme o modelo apresentado a seguir:

- 1 Importe as bibliotecas do método selecionado
- 2 Defina a lista com as coordenadas de cada nodo
- 3 Defina a lista com as demandas para cada nodo
- 4 Defina os veículos e suas capacidades
- 5 Defina as distâncias entre os nodos
- 6 Configure o método
- 7 Execute o método
- 8 Imprima o resultado do método

Em pyVRP (WOUDA; LAN; KOOL, 2024) temos uma lista de tuplas para as coordenadas do depósito e dos clientes, uma lista simples para as demandas, veículos são criados através do modelo da biblioteca, distâncias são calculadas e armazenadas na forma de arcos e, por fim, configurou-se o tempo máximo de execução para 12 horas. Vale mencionar que pyVRP faz uso de paralelismo para acelerar operações de busca.

Na implementação de pyVRP por Pereira (2021) encontramos uma organização um pouco diferente dos demais. As coordenadas e parâmetros (para este trabalho o único parâmetro utilizado será a demanda) são carregadas através de arquivos separados. A matriz de distâncias é montada internamente pelo método, respeitando o uso de distâncias euclidianas entre dois pontos. Quanto a definição dos atributos dos veículos, destaca-se que este método permite a configuração de velocidade para os veículos, que foi mantida com o valor de uma unidade para todos os veículos criados para que não tivesse impacto nos testes. O método apresentou melhores resultados com um tamanho de população igual a 100, uma taxa de mutação de 0.10 e um número de gerações de 2500.

Este método merece um destaque diante os demais pelo alto número de opções de configuração, permitindo assim testes mais refinados. Além disso, a saída deste modelo possui muito mais detalhes sobre as rotas e as demandas, gerando um relatório em formato *CSV* ao final.

Por fim, no método VRPy de Montagné, Sanchez and Storbugt (2020) temos nodos definidos por tuplas de coordenadas e demandas juntas, a criação de todos os nodos na biblioteca *DiGraph* e, em seguida, a criação de todas as arestas sendo que o custo de cada uma é representado por uma lista de custos por veículo. A capacidade dos veículos é informada em uma única lista de tamanho n , onde n é o número de veículos. Por recomendação dos autores, para instâncias grandes, recomenda-se que o parâmetro *greedy* seja ativado.

Em anexo, encontram-se de forma objetiva os parâmetros utilizados nos experimentos.

Quanto as instâncias de testes, foram escolhidas de diferentes *datasets* presentes na literatura. Optou-se, dado os resultados dos experimentos, de não ultrapassar um limite de 101 nodos (sendo 1 depósito e 100 clientes). Os *datasets* foram obtidos em <http://vrp.galgos.inf.puc-rio.br/index.php/en/>.

Para cada instância foi criado um conjunto de veículos com capacidades diferentes, formando assim uma frota fixa e heterogênea, conforme proposto pelo tema do trabalho. Os valores foram selecionados a partir do valor de capacidade fornecido pelos criadores destas mesmas instâncias. A relação entre clientes, número de veículos e capacidade é apresentada na tabela 1.

Clientes	Veículos	Capacidades
15	4	30
	4	40
18	1	120
	1	200
31	2	90
	2	100
	1	120
50	1	150
	1	200
	2	250
78	3	60
	2	80
	3	120
	2	140
100	1	400
	2	600
	2	800
	2	1000

Tabela 1 – Configuração de veículos e capacidades para diferentes quantidades de clientes

Além disso, destaca-se que as instâncias escolhidas apresentam as coordenadas dos nodos em duas dimensões, sendo definido no cabeçalho das instâncias que as distâncias precisam ser calculadas usando distância euclidiana entre dois pontos.

Quanto a execução dos experimentos, os programas especificados anteriormente foram executados em um único computador com as seguintes características:

- Processador AMD Ryzen 5 7600
- Memória RAM 32gb 6000mHz
- Placa de vídeo RTX 3070 8gb
- SSD NVMe 2tb 5000mb/s
- Fonte 850w 80 plus

Afim de se manter um controle sobre os testes, foi definido um limite de 12 horas (43200 segundos) para cada execução dos experimentos. Fora esse controle, todas as demais configurações seguiram conforme recomendações dos autores em seus trabalhos, como tamanho de população, número de gerações e modelo do *solver*, por exemplo.

5.3 Análise

A tabela 2 apresenta os resultados obtidos para análise neste trabalho, enquanto a tabela 3 apresenta os mesmos resultados porém adotando o modelo pyVRP de Wouda, Lan and Kool (2024) como valor de referência. Para fins de identificação, a implementação denominada pyVRP de Pereira (2021) teve seu nome substituído para pyVRP (UFF) em alusão a Universidade Federal Fluminense, universidade ontem o autor atua. O outro pyVRP será identificado por pyVRP (WLK), que representam as iniciais dos nomes dos autores.

	Custo na primeira solução			Custo na melhor solução			Tempo para encontrar a melhor solução (s)		
	pyVRP (WLK)	pyVRP (UFF)	VRPy	pyVRP (WLK)	pyVRP (UFF)	VRPy	pyVRP (WLK)	pyVRP (UFF)	VRPy
15 Clientes	430	476	442	430	438	422	1	2	26
18 Clientes	205	384	959	205	212	196	1	12	9002
31 Clientes	743	2027	3740	743	841	883	1	152	43200
50 Clientes	474	1814	2402	459	541	588	2	226	43200
77 Clientes	1424	4357	7199	1342	1407	2009	4	267	43200
100 Clientes	13214	49861	90010	12956	13624	20448	5	2678	43200

Tabela 2 – Comparação entre pyVRP (WLK), pyVRP (UFF) e VRPy para diferentes métricas e tamanhos de clientes.

	Custo na primeira solução			Custo na melhor solução			Tempo para encontrar a melhor solução		
	pyVRP (WLK)	pyVRP (UFF)	VRPy	pyVRP (WLK)	pyVRP (UFF)	VRPy	pyVRP (WLK)	pyVRP (UFF)	VRPy
15 Clientes	430	+10.70%	+2.79%	430	+1.86%	-1.86%	1	+100.00%	+2500.00%
18 Clientes	205	+87.32%	+367.80%	205	+3.41%	-4.39%	1	+1100.00%	+900100.00%
31 Clientes	743	+172.84%	+403.36%	743	+13.17%	+18.85%	1	+15100.00%	+4319900.00%
50 Clientes	474	+282.28%	+406.33%	459	+17.87%	+28.33%	2	+11200.00%	+2159900.00%
78 Clientes	1424	+206.00%	+405.33%	1342	+4.84%	+49.78%	4	+6550.00%	+1079900.00%
100 Clientes	13214	+277.39%	+581.41%	12956	+5.16%	+57.77%	5	+53460.00%	+863900.00%

Tabela 3 – Comparação entre pyVRP (WLK), pyVRP (UFF) e VRPy tomando pyVRP como referência.

A tabela apresenta uma comparação detalhada entre os modelos pyVRP (WLK), pyVRP (UFF) e VRPy, utilizando métricas previamente definidas. Cada uma dessas métricas fornece detalhes importantes sobre a eficácia e eficiência dos modelos analisados. Vale salientar que os valores anotados na tabela para o pyVRP (UFF) correspondem ao menor custo na melhor solução obtido após 12 execuções para cada instância de teste, já que este modelo apresenta por natureza valores diferentes em sua solução a cada execução.

No que diz respeito ao custo na primeira solução, observa-se que o pyVRP (WLK) apresenta valores relativamente baixos em todas as instâncias, especialmente em instâncias menores (15 a 50 clientes). Este resultado é consistente com os achados de Wouda, Lan and Kool (2024), que destacam a eficiência de algoritmos genéticos híbridos para obter boas soluções

iniciais em VRPs. Isso indica que o algoritmo consegue explorar rapidamente soluções de qualidade, o que é vantajoso em contextos onde a velocidade de resposta inicial é um fator crítico. O pyVRP (UFF) exibe custos iniciais significativamente mais elevados em comparação com o pyVRP (WLK), o que sugere que suas modificações podem estar otimizadas para buscar soluções mais refinadas em vez de soluções rápidas iniciais.

O modelo VRPy, que utiliza uma abordagem de geração de colunas, apresenta um custo inicial elevado em todas as instâncias, especialmente em problemas com mais de 30 clientes. Métodos baseados em geração de colunas são projetados para otimizar soluções ao longo do tempo, em vez de obter uma boa solução inicial, o que explica os resultados observados (MONTAGNÉ; SANCHEZ; STORBUGT, 2020). Essa abordagem pode ser útil em casos onde é necessário um refinamento contínuo e detalhado das rotas, embora, para instâncias menores, esse alto custo inicial possa ser uma desvantagem, especialmente em aplicações com restrições de tempo.

Em relação ao custo na melhor solução, o pyVRP (WLK) continua a se destacar nas instâncias menores, especialmente na de 15 clientes, onde alcança um custo final muito próximo do custo inicial. Esse comportamento sugere que o pyVRP (WLK) é capaz de encontrar uma solução de qualidade de maneira rápida e refinar a solução marginalmente, mantendo um alto desempenho em termos de custo total. No entanto, à medida que o número de clientes aumenta, o pyVRP (WLK) tende a perder sua vantagem, porém a limitação de tempo impede que VRPy chegue em resultados provavelmente melhores que os métodos com algoritmos genéticos. A instância de 15 clientes, por ser menor, reflete a capacidade do VRPy de melhorar progressivamente as soluções em problemas mais complexos. Essa vantagem do VRPy confirma o potencial dos algoritmos de geração de colunas em explorar o espaço de soluções de forma eficiente em escalas maiores, conforme observado por Montagné, Sanchez and Storbugt (2020).

O pyVRP (UFF), por sua vez, apresenta uma evolução significativa entre o custo inicial e o custo final. Esse comportamento está alinhado com as observações de Pereira (2021), que ressalta que o uso de adaptações em algoritmos genéticos pode aprimorar a qualidade da solução ao longo do tempo, mesmo que o custo inicial seja alto. Em particular, nas instâncias de 75 e 100 clientes, o pyVRP (UFF) reduz substancialmente o custo final em comparação com o custo inicial, se destacando por sua capacidade de alcançar soluções de alta qualidade mesmo levando um pouco mais de tempo para convergir para o resultado final.

A análise do tempo para encontrar a melhor solução mostra diferenças grandes entre os três métodos. O pyVRP (WLK) é claramente o mais rápido, com tempos de resposta entre 1 e 5 segundos. Esse desempenho é coerente com a literatura, onde Wouda, Lan and Kool (2024) enfatizam a capacidade dos algoritmos genéticos híbridos de explorar rapidamente boas soluções iniciais. O pyVRP (UFF), em contrapartida, apresenta tempos de execução elevados nas instâncias maiores, atingindo até 2678 segundos para 100 clientes. Esses resultados sugerem que as adaptações realizadas podem estar priorizando a qualidade da solução em detrimento

da velocidade, como observado por Pereira (2021), que aponta que melhorias incrementais nos algoritmos genéticos frequentemente exigem um tempo maior para convergir.

Por fim, o VRPy, que mantém um tempo fixo de 43200 segundos para as instâncias de 30 clientes em diante, utiliza esse período para maximizar o refinamento da solução, explorando amplamente o espaço de busca. Embora esse tempo seja alto, ele induz que o VRPy encontre a melhor solução possível dentro de um cenário onde o tempo não é um fator crucial, o que pode ser útil em situações onde a qualidade final da solução é o mais importante.

Nota-se também uma relação entre o a melhora da solução inicial para a melhor solução e o tamanho das instâncias. Com base na tabela 2 é possível observar que conforme o número de clientes aumenta, a diferença entre a primeira solução encontrada e a melhor solução tende a aumentar. Como o modelo pyVRP (WLK) já busca partir de uma solução inicial próxima ao ótimo Wouda, Lan and Kool (2024), este apresenta uma diferença pequena entre estes valores. Esta diferença é maior para pyVRP (UFF), que procura iniciar com soluções mais variadas e ir explorando para baixar esse valor. Para VRPy, como menciona Montagné, Sanchez and Storbugt (2020), um bom valor na solução inicial não é buscado, visto que pretende-se diminuí-lo de maneira iterativa ao longo do tempo.

Quanto as rotas, a figura 4 demonstra os resultados finais da execução para a instância de 31 clientes com os métodos pyVRP (WLK) e pyVRP (UFF), respectivamente. Nelas observamos a alta similaridade entre as rotas encontradas. O método VRPy não exibia rotas de forma gráfica na saída e, como os valores de seus custos se mantiveram muito mais altos devido ao limite de tempo, optou-se por não apresentá-los. Os dois primeiros métodos mencionados apresentam resultados onde se pode procurar por pontos de não similaridade, visto que não muito parecidos. Já nos gráficos do modelo de VRPy procuraria-se por pontos de similaridade no meio de muitas sobreposições, reforçando assim que não agregaria ao trabalho exibi-los.

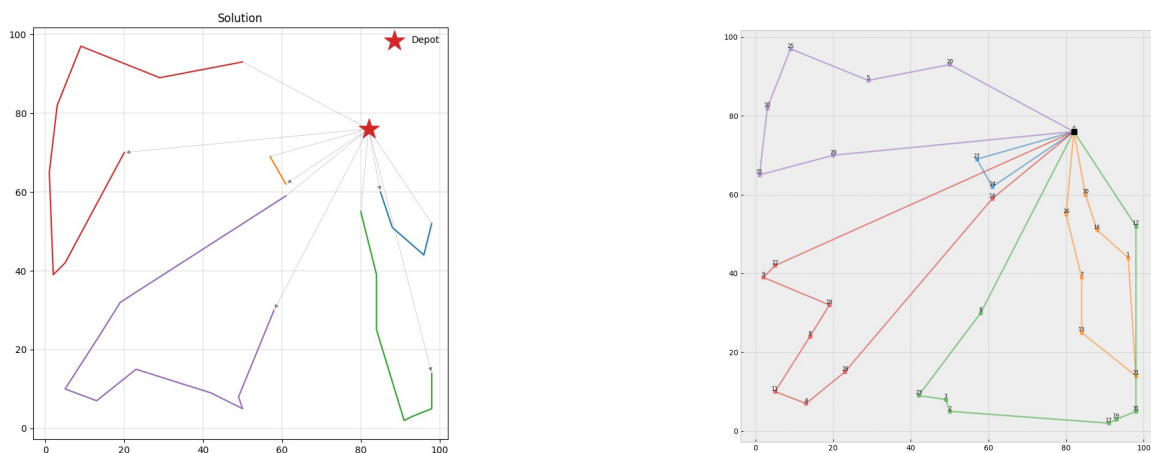


Figura 4 – Comparação de métodos para 31 clientes entre Wouda, Lan and Kool (2024) e Pereira (2021)

Para o exemplo com 31 clientes é possível observar algumas poucas diferenças entre as rotas descobertas, porém para a instância com 100 clientes as diferenças aumentam consideravelmente, principalmente pela diferença de 5,15% no custo entre os modelos, conforme visualizado na figura 5. Para esta instância maior observa-se que principalmente o reflexo desta diferença de custos está no enorme número de sobreposições de rotas usando o método pyVRP (UFF).

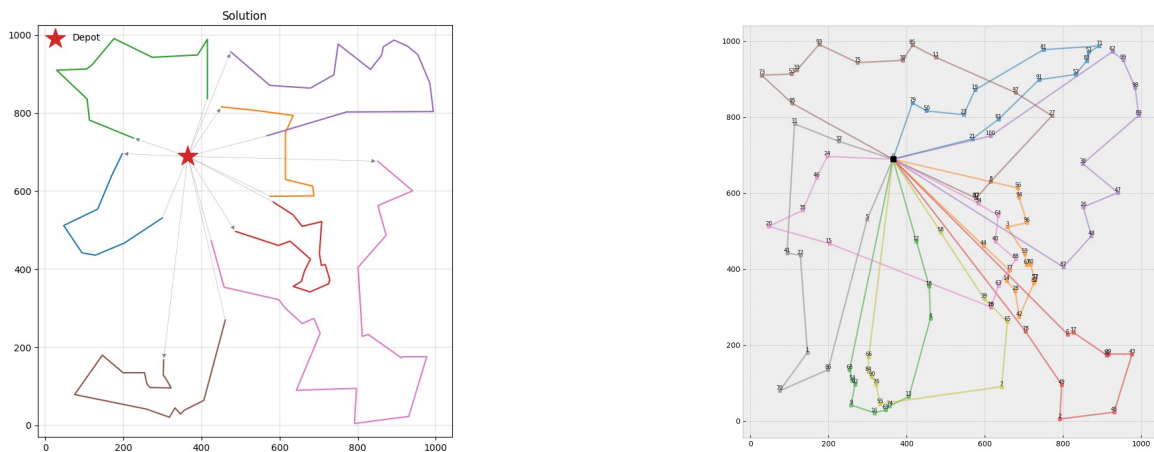


Figura 5 – Comparação de métodos para 100 clientes entre Wouda, Lan and Kool (2024) e Pereira (2021)

O tempo pode ser analisado através do gráfico da figura 6. Podemos observar que o método pyVRP (WLK), para o conjunto de instâncias utilizado, apresentou um comportamento próximo do linear. O método pyVRP (UFF), por sua vez, apresenta um comportamento um pouco diferente do método anterior, tendo um salto visível de tempo para a maior das instâncias. Já o VRPy atingiu rapidamente o limite de tempo e manteve este comportamento até o fim.

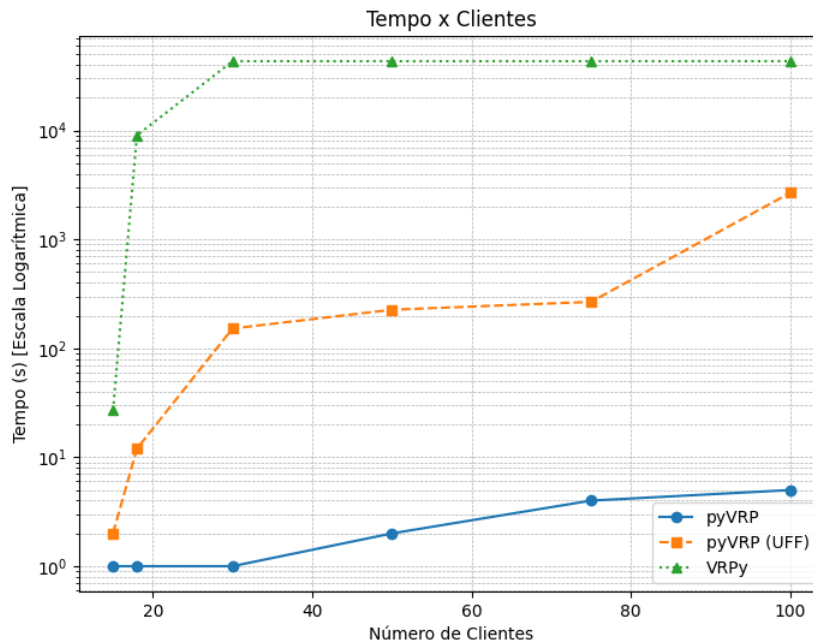


Figura 6 – Gráfico de tempo pelo tamanho das instâncias.

Em resumo, cada método possui características distintas que o tornam adequado para diferentes contextos de aplicação. O pyVRP (WLK) destaca-se pela velocidade e pelo custo competitivo, sendo indicado para problemas onde a rapidez é fundamental. O pyVRP (UFF), apesar do custo inicial maior, oferece soluções de alta qualidade e, como mencionado anteriormente, possui o fator chave de permitir um maior número de customizações do problema, sendo adequado para cenários onde o tempo de execução é secundário em relação à qualidade da solução final ou onde necessita-se modelar o problema com mais detalhes. O VRPy, com seu enfoque na geração de colunas, é mais apropriado para instâncias onde a qualidade da solução é o principal objetivo. Esses resultados confirmam as discussões presentes na literatura, onde a escolha do método depende do equilíbrio entre qualidade e tempo de execução, conforme destacado por Vidal et al. (2013).

6 CONCLUSÃO

O presente trabalho teve como objetivo comparar diferentes métodos de resolução aplicados ao problema de roteamento de veículos heterogêneos com frota fixa. Através dessa análise, buscou-se avaliar o desempenho e a adequação de cada abordagem em termos de eficiência e qualidade das soluções geradas, visando identificar os métodos mais promissores para esse tipo específico de problema de otimização.

O comparativo realizado evidenciou tanto os pontos fortes quanto as limitações de cada modelo testado. Em particular, foram identificadas abordagens com maior eficácia em certas condições e com desvantagens em outras, permitindo um entendimento mais detalhado das características de cada método e de como eles se comportam em diferentes cenários e restrições.

Observou-se que, em instâncias de menor porte, os métodos apresentaram o comportamento esperado, com resultados eficientes e soluções de boa qualidade. No entanto, à medida que as instâncias aumentaram em complexidade e tamanho, o método baseado em geração de colunas demonstrou limitações significativas, falhando em manter o desempenho e, muitas vezes, não conseguindo alcançar soluções competitivas em comparação com os outros métodos testados.

Desta forma, o método pyVRP (WLK) de Wouda, Lan and Kool (2024) demonstrou rapidez e eficiência para apresentar resultados, porém não conseguiu garantir os menores custos em todas as instâncias. Por sua vez, o método de pyVRP (UFF) apresenta resultados competitivos, porém levando um pouco mais de tempo que pyVRP (WLK). Como ponto positivo, destaca-se o grande número de personalizações possíveis no modelo de Pereira (2021), permitindo melhores modelagens. Por fim, VRPy apresenta o maior potencial para obter custos mais baixos, devido a seu método iterativo, tendo como ponto negativo apenas seu alto tempo de execução.

Os objetivos desse trabalho foram atingidos, tendo selecionado métodos para o problema de roteamento de veículos heterogêneos com frota fixa e comparado o desempenho destes métodos em cenários experimentais. Por fim, pretende-se dar publicidade a esse trabalho para que sirva de base a outros trabalhos na área de problemas de otimização.

6.1 Trabalhos Futuros

Quanto a trabalhos futuros, podemos abordar os seguintes pontos:

- Realizar novos comparativos, utilizando os resultados obtidos neste trabalho como base para futuras comparações.

-
- Realizar testes com tempos de execução mais longos, visando explorar a capacidade dos algoritmos em encontrar soluções ótimas em instâncias complexas.
 - Ampliar a avaliação dos algoritmos com instâncias variadas e dados de benchmark para melhorar a aplicabilidade e robustez em cenários reais.
 - Auxiliar na formalização de uma solução para o problema, já que o comparativo realizado ajuda a entender pontos fortes e fracos de outros modelos, podendo ajudar no desenvolvimento.

Referências

- BALDACCI, R.; BATTARRA, M.; VIGO, D. **Routing a Heterogeneous Fleet of Vehicles**. [S.l.]: Springer, 2008.
- BLUM, C.; ROLI, A. Metaheuristics in combinatorial optimization: Overview and conceptual comparison. **ACM Computing Surveys**, v. 35, p. 268–308, 01 2001.
- BRANDAO, J. A tabu search algorithm for the heterogeneous fixed fleet vehicle routing problem. **Computers Operations Research**, v. 38, p. 140–151, 01 2011.
- COSTA, A.; OLIVERA, J. F.; TOTH, P. Heuristic and exact algorithms for vehicle routing problems with heterogeneous fixed fleets. **Operations Research Perspectives**, Elsevier, v. 6, p. 100116, 2019.
- COSTA, L.; CONTARDO, C.; DESAULNIERS, G. Exact branch-price-and-cut algorithms for vehicle routing. **Transportation Science**, v. 53, n. 4, p. 946–985, 2019.
- DANTZIG, G. B.; RAMSER, J. H. The truck dispatching problem. **Management Science**, v. 6, p. 1–140, 10 1959.
- DANTZIG, G. B.; WOLFE, P. Decomposition principle for linear programs. **Operations Research**, INFORMS, v. 8, n. 1, p. 101–111, 1954.
- DESAULNIERS, G.; DESROSIERS, J.; SOLOMON, M. Column generation. In: **Column Generation**. [S.l.]: Springer, 2005. p. 1–32.
- DONDO, R.; CERDÁ, J. A cluster-based optimization approach for the multi-depot heterogeneous fleet vehicle routing problem with time windows. **European Journal of Operational Research**, v. 176, p. 1478–1507, 02 2007.
- ERRAMI, N. et al. VRPSolverEasy: a python library for the exact solution of a rich vehicle routing problem. **Preprint**, 2023. Disponível em: <<https://inria.hal.science/hal-04057985v2>>.
- FERLAND, J. A.; MICHELON, P. The vehicle scheduling problem with multiple vehicle types. **The Journal of the Operational Research Society**, v. 39, p. 577–583, 06 1988.
- GLOVER, F.; LAGUNA, M. **Tabu search I**. [S.l.]: INFORMS Journal on Computing, 1999.
- GOLDBERG, D. E. **Genetic Algorithms in Search, Optimization, and Machine Learning**. Boston, MA: Addison-Wesley, 1989.
- GOLDEN, B. et al. The fleet size and mix vehicle routing problem. **Computers Operations Research**, v. 11, p. 49–66, 03 1984.
- HOFF, A. et al. Industrial aspects and literature survey: Fleet composition and routing. **Computers Operations Research**, v. 37, p. 2041–2061, 12 2010.
- HOLLAND, J. H. **Adaptation in Natural and Artificial Systems**. Ann Arbor, MI: University of Michigan Press, 1975.

- IRNICH, S. A multi-depot pickup and delivery problem with a single hub and heterogeneous vehicles. **European Journal of Operational Research**, v. 122, p. 310–328, 04 2000.
- KRITIKOS, M. N.; IOANNOU, G. The heterogeneous fleet vehicle routing problem with overloads and time windows. **International Journal of Production Economics**, v. 144, p. 68–75, 07 2013.
- LAPORTE, G. et al. Thirty years of heterogeneous vehicle routing. **European Journal of Operational Research**, v. 249, p. 1–21, 07 2015.
- LI, F.; GOLDEN, B.; WASIL, E. A record-to-record travel algorithm for solving the heterogeneous fleet vehicle routing problem. **Computers & Operations Research**, v. 34, p. 2734–2742, 09 2007.
- LI, J. et al. Deep reinforcement learning for solving the heterogeneous capacitated vehicle routing problem. **IEEE Transactions on Cybernetics**, IEEE, v. 52, n. 12, p. 13572–13584, 2022.
- LIU, Q. et al. 2d-ptr: 2d array pointer network for solving the heterogeneous capacitated vehicle routing problem. In: **Proceedings of the 23rd International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2024)**. Auckland, New Zealand: International Foundation for Autonomous Agents and Multiagent Systems (IFAAMAS), 2024. p. 1238–1247.
- MITCHELL, M. **An Introduction to Genetic Algorithms**. Cambridge, MA: MIT Press, 1998.
- MONTAGNÉ, R.; SANCHEZ, D. T.; STORBUGT, H. O. Vropy: A python package for solving a range of vehicle routing problems with a column generation approach. **Journal of Open Source Software**, The Open Journal, v. 5, n. 55, p. 2408, 2020. Disponível em: <<https://doi.org/10.21105/joss.02408>>.
- PEREIRA, V. **Metaheuristic Lesson: Genetic Algorithm - VRP (Vehicle Routing Problem)**. 2021. Available at <<https://www.researchgate.net/publication/353660717>>.
- ROTHLAUF, F. **Design of Modern Heuristics**. [S.l.]: Springer, 2011.
- SADYKOV, R. et al. Primal heuristics for branch and price: The assets of diving methods. **INFORMS Journal on Computing**, v. 31, n. 2, p. 251–267, 2019.
- SADYKOV, R. et al. Primal heuristics for column generation. **Computers & Operations Research**, Elsevier, v. 106, p. 1–16, 2019.
- TAILLARD, E. D. A heuristic column generation method for the heterogeneous fleet vrp. **RAIRO-Oper. Res**, v. 33, p. 1–14, 01 1999.
- TARANTILIS, C. D.; KIRANOUDIS, C. T. A meta-heuristic algorithm for the efficient distribution of perishable foods. **Journal of Food Engineering**, v. 50, p. 1–9, 10 2001.
- TARANTILIS, C. D.; KIRANOUDIS, C. T.; VASSILIADIS, V. S. A list based threshold accepting metaheuristic for the heterogeneous fixed fleet vehicle routing problem. **Journal of the Operational Research Society**, v. 54, p. 65–71, 03 2003.

TARANTILIS, C. D. et al. A reactive variable neighborhood tabu search for the heterogeneous fleet vehicle routing problem with time windows. **Journal of Heuristics**, v. 14, p. 425–455, 10 2007.

VIDAL, T. et al. A hybrid genetic algorithm with adaptive diversity management for a large class of vehicle routing problems with time-windows. **Computers & Operations Research**, Elsevier, v. 40, p. 475–489, 2013.

WILSON, R. **Graph Theory**. Dover Publications, 1996. ISBN 0582249937. Disponível em: <https://www.maths.ed.ac.uk/~v1ranick/papers/wilsongraph.pdf>.

WOUDA, N. A.; LAN, L.; KOOL, W. Pyvrp: A high-performance vrp solver package. **INFORMS Journal on Computing**, Institute for Operations Research and the Management Sciences (INFORMS), v. 36, n. 4, p. 943–955, 07 2024. ISSN 1526-5528. Disponível em: <http://dx.doi.org/10.1287/ijoc.2023.0055>.

7 Apêndices

COMPARATIVO DE MÉTODOS PARA O PROBLEMA DE ROTEAMENTO DE VEÍCULOS HETEROGÊNEOS COM FROTA FIXA

Eduardo Borges Siqueira, Rafael de Santiago

Universidade Federal de Santa Catarina - Centro Tecnológico - Departamento de Informática e Estatística - Florianópolis, SC

Abstract

With the growth of urban centers and the constant demand for products, the transportation and logistics sector increasingly calls for efficient ways to manage vehicle routes. Whether transporting passengers, products, materials, or even waste, planning fleet routes becomes a complex task due to the number of variables and conditions to consider. In this context, VRPs (Vehicle Routing Problems) have proven to be powerful tools for primarily reducing logistical costs and increasing the operational efficiency of various systems, while also generating positive impacts on environmental issues and urban mobility. Therefore, the objective of this study is to evaluate and compare solutions for a more detailed variation of VRPs, the HFFVRPs (Heterogeneous Fixed Fleet Vehicle Routing Problems). With this version of the problem, we can get closer to real-world scenarios by working with fleets of a fixed size and vehicles that differ in aspects such as capacity, operational cost, and efficiency. To this end, reviews of the existing literature on the topic were conducted, as well as comparisons between published results, aiming to determine how computational solutions can reduce costs in the transportation stages of supply chains. As a result, we observed how genetic algorithms demonstrated speed in obtaining solutions and an overall good performance, presenting limitations only in refining their final costs. On the other hand, the column generation-based method showed greater efficiency in optimizing results, although it required significantly higher execution times.

Resumo

Com o crescimento dos centros urbanos e a constante necessidade por produtos, o setor de transportes e logística clama cada vez mais por formas eficientes de fazer o manejo das rotas de seus veículos. Seja transportando passageiros, produtos, materiais ou até mesmo resíduos, planejar o trajeto de suas frotas torna-se uma tarefa complexa devido ao número de variáveis e condições à se atentar. Nesse contexto, os VRPs (*Vehicle Routing Problems* ou Problemas de Roteamento de Veículos) têm se mostrado ferramentas poderosas para, principalmente, reduzir custos logísticos e aumentar a eficiência operacional dos mais diversos sistemas, podendo também gerar impactos positivos em questões ambientais e de mobilidade urbana. Assim, o

objetivo deste trabalho é avaliar e comparar soluções para uma variação mais detalhada dos VRPs, os HFFVRPs (*Heterogeneous Fixed Fleet Vehicle Routing Problems* ou Problemas de Roteamento de Veículos Heterogêneos com Frota Fixa). Com esta versão do problema, podemos nos aproximar um pouco mais dos cenários reais, trabalhando com frotas com tamanho estabelecido e com veículos que diferem entre si em aspectos como capacidade, custo operacional e eficiência, por exemplo. Visto isso, seguindo este caminho, foram realizadas revisões na literatura existente sobre do tema, bem como comparativos entre os resultados publicados, visando constatar como soluções computacionais podem atuar na redução dos custos nas etapas de transporte de cadeias logísticas. Como resultado, observamos como algoritmos genéticos demonstraram rapidez na obtenção de soluções e bom desempenho num geral, apresentando limitações somente no refinamento de seus custos finais. Por outro lado, o método baseado em geração de colunas evidenciou maior eficiência na otimização dos resultados, embora tenham exigido tempos de execução significativamente mais elevados.

7.1 Introdução

Problemas de roteamento de veículos são desafios constantes em toda a cadeia logística ao redor do mundo. Seja para transportar matéria-prima para as indústrias, materiais para fábricas, produtos para lojas ou para os clientes, o transporte de cargas se mostra fundamental para a existência de qualquer rede. Com isso, surgindo como alternativa para otimizar as frotas, os custos com transporte, o tempo de deslocamento e até mesmo reduzir os impactos ambientais, os VRPs (*Vehicle Routing Problems* ou Problemas de Roteamento de Veículos) vêm ganhando bastante destaque nos ramos de pesquisa nesta área.

Em grandes economias, como o caso da União Europeia, o setor de transportes chega a representar mais de 10% do produto interno bruto do bloco (HOFF et al., 2010). Mesmo sendo um valor expressivo, Hoff et al. (2010) ainda afirma que devido ao crescimento econômico, crescimento do consumo e da globalização tendem a aumentar esse valor, assim como o aumento da competição entre as empresas do setor logístico e os proprietários dos bens a serem transportados tendem também a criar altas demandas por sistemas mais complexos e eficientes.

Com isso, existem diversas variações para esse problema, dado que cada variação aumenta consideravelmente sua complexidade por se aproximar mais da realidade, adicionando fatores como janelas de tempo nas rotas, possibilidade de carga e descarga nos pontos de parada, entre outros (LAPORTE et al., 2015). Dentre eles, destaca-se uma comum situação da realidade, objeto de pesquisa desta monografia, o problema de roteamento de veículos heterogêneos com frota fixa, ou seja, trabalha-se com um número fixo de veículos, diferentes entre si.

Brandao (2011) mostrou que o problema de HFFVRP é um problema NP-Difícil. Além disso, o mesmo comenta que, quando comparado a problemas de natureza semelhante, como por exemplo FSMVRP (*Fleet Size and Mix Vehicle Routing Problems* - Problemas de Roteamento de Veículos com Frota Mista), onde se tem veículos a vontade, se mostra consideravelmente mais complexo, colocando-o entre os mais difíceis de serem resolvidos.

Nesse contexto, é possível observar um conflito de interesses entre as partes envolvidas na cadeia logística. Por um lado, temos clientes que desejam seus bens entregues o mais rápido possível, resultando num tratamento exclusivo e direto, sem passar por outros pontos. Do outro, temos operadores logísticos que desejam reduzir gastos o máximo possível e diminuir a distância percorrida por seus veículos, buscando uma maior eficiência na entregas.

Com isso, esse trabalho analisa e compara soluções presentes na literatura que se propõem a resolver o problema de roteamento de veículos heterogêneos com frota fixa, buscando como objetivo reduzir tanto os custos fixos do transporte, como o custo por rota realizada por cada veículo, como os custos variáveis, relacionados principalmente a distância a ser percorrida (para este trabalho, pois na literatura, como aponta Laporte et al. (2015), podemos encontrar diversas variações com outros custos variáveis).

7.2 Metodologia

Este trabalho caracteriza uma pesquisa qualitativa e exploratória, visando analisar e comparar métodos propostos para o problema de roteamento de veículos heterogêneos com frota fixa, apresentando os resultados analisados.

Para o desenvolvimento da pesquisa, são realizadas as seguintes etapas:

Primeiramente, é feito um levantamento teórico do estado da arte acerca do HFFVRP, por meio de uma pesquisa bibliográfica utilizando a literatura disponível.

Após a primeira etapa, são selecionados na literatura métodos possíveis para resolver o problema.

Em seguida, os métodos são devidamente implementados e analisados, seguindo corretamente o que foi descrito nos trabalhos.

Por fim, os resultados são comparados e discutidos, baseando-se na função objetivo dos mesmos.

7.3 Trabalhos Relacionados

A busca iniciou-se através do site Periódicos CAPES, buscando trabalhos que possuíssem as palavras chave "HFFVRP" ou "Heterogeneous Fixed Fleet Vehicle Routing Problem", obtendo assim 138 resultados. Deste grupo, foram selecionados três artigos com propostas de solução ao tema. O primeiro, propõe uma análise histórica dos problemas de roteamento de veículos heterogêneos (*Heterogeneous Vehicle Problems - HVRPs*) (LAPORTE et al., 2015), outro propõe uma busca tabu como solução do problema (BRANDAO, 2011) e, por fim, um que implementa uma solução baseada em uma heurística de limite de aceitação (*threshold accepting heuristic*) (TARANTILIS; KIRANOUDIS; VASSILIADIS, 2003).

A partir do trabalho de Laporte et al. (2015) foi possível obter um panorama ampliado do contexto histórico das pesquisas e do desenvolvimento na área de roteamento de veículos heterogêneos. Através dele, conseguiu-se acesso ao trabalho de Tarantilis and Kiranoudis (2001), um dos pioneiros em pesquisa nesse segmento de HFFVRP. Por fim, o trabalho de Brandao (2011) traz uma visão mais abrangente do problema em si, pontuando de maneira mais clara questões teóricas da literatura para esse problema.

7.4 Thirty years of heterogeneous vehicle routing - Laporte, Gilbert & Koc, cagri & Bektas, Tolga & Jabali, Ola

Em Laporte et al. (2015) os autores retratam sobre o contexto histórico dos problemas de roteamento de veículos heterogêneos (HVRPs), montando assim um aglomerado de informações

sobre essa classe de problemas e suas mais tradicionais variações.

Os autores então denotam as duas classes mais importantes de problemas de roteamento de veículos com características diferentes, sendo elas FSMVRPs (*Fleet Size and Mix Vehicle Routing Problems* - Problemas de Roteamento de Veículos com Frota Mista) (GOLDEN et al., 1984) e HFFVRPs (*Heterogeneous Fixed Fleet Vehicle Routing Problems* ou Problema de Roteamento de Veículos Heterogêneos com Frota Fixa) (TAILLARD, 1999). Segundo os autores, a diferença entre elas se dá por conta do roteamento com frota mista levar em consideração que a frota é infinita, ou seja, têm-se quantos veículos quiser. Já no problema de frota fixa, assume-se um conjunto limitado de opções de veículos.

Dentro das classe informadas, Laporte et al. (2015) apresenta cinco principais variantes, sendo elas:

- FSMVRPs com custos de veículos fixos e variáveis (FERLAND; MICHELON, 1988);
- FSMVRPs somente com custos fixos de veículos (GOLDEN et al., 1984);
- FSMVRPs somente com custos variáveis de veículos (TAILLARD, 1999);
- HFFVRPs com custos de veículos fixos e variáveis (LI; GOLDEN; WASIL, 2007);
- HFFVRPs somente com custos variáveis de veículos (TAILLARD, 1999);

São tratadas ainda outras variantes, que ajudam o problema a ficar mais próximo da realidade. Dentre elas, temos as janelas de tempo (TARANTILIS et al., 2007), múltiplos depósitos (DONDO; CERDÁ, 2007), coletas e entregas (IRNICH, 2000), sobrecargas (KRITIKOS; IOANNOU, 2013), entre diversas outras catalogadas pelos autores.

Comentado sobre as variações presentes na literatura, os autores percorrem mais detalhadamente sobre as mesmas, explicando com maior profundidade suas peculiaridades. Além disso, mencionam trinta e duas soluções na literatura especificamente para o problema de roteamento de veículos heterogêneos com frota fixa, explicando se as mesmas atacam problemas variantes apresentados, se possuem modelo matemático de solução, a maneira com que solucionaram o problema e se faz parte do estudo de caso realizado por eles.

Após o breve estudo de caso, onde apresentam poucos resultados para debater sobre o assunto, concluem que a área é promissora e que possui apenas soluções que se apresentaram sub-ótimas para as cinco principais variantes apresentadas até o momento, visto a complexidade do problema. Uma crítica mencionada por Laporte et al. (2015) é de que o número de trabalhos com altas abstrações é muito grande, fazendo com que os resultados sejam distantes da realidade.

7.4.1 PyVRP: A High-Performance VRP Solver Package - Niels A. Wouda & Leon Lan & Wouter Kool

Algoritmos genéticos são amplamente utilizados na resolução de problemas de otimização complexos, como o problema de roteamento de veículos (VRP). Estes algoritmos se destacam por seu método de busca, inspirado na evolução biológica, que envolve operações como seleção, cruzamento e mutação para evoluir uma população de soluções em direção a soluções de melhor qualidade ao longo de várias gerações (HOLLAND, 1975). O modelo proposto em PyVRP utiliza o método de busca genética híbrida *Hybrid Genetic Search*, HGS, que combina algoritmos genéticos com busca local, demonstrando resultados de ponta para VRPs capacitados (CVRP) como um todo (VIDAL et al., 2013).

Wouda, Lan and Kool (2024) destacam que, em PyVRP, apenas as partes críticas em desempenho são implementadas em C++, enquanto o restante é programado em Python, o que facilita a personalização e o uso do seu modelo. Esse design modular permite aos pesquisadores personalizar aspectos como o gerenciamento da população e estratégias de cruzamento, ampliando as possibilidades de aplicação de HGS a diferentes variantes de VRPs.

Os métodos genéticos foram especialmente projetados para equilibrar qualidade e diversidade das soluções, evitando a convergência para ótimos locais. Essa abordagem segue as diretrizes de Vidal et al. (2013) sobre o uso de diversificação e intensificação para melhorar a performance na resolução de VRPs. Por exemplo, este modelo incorpora um gerenciador de penalidades que ajusta dinamicamente os pesos das penalidades para garantir que um percentual específico de buscas locais gere soluções viáveis, uma abordagem que sustenta a eficiência do algoritmo ao explorar e intensificar o espaço de soluções (VIDAL et al., 2013).

Como exemplo de solução utilizando esse método, temos o gráfico exibido na figura 2.

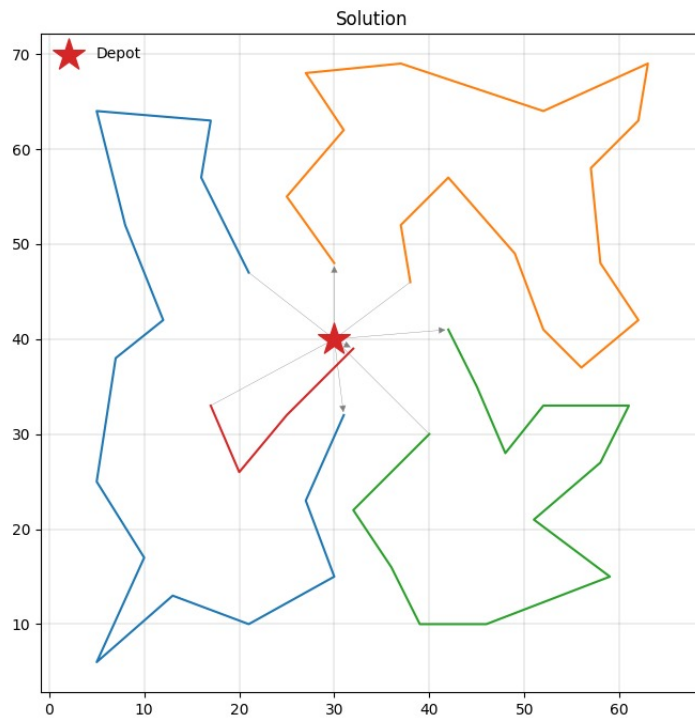


Figura 7 – Exemplo de saída do método proposto por Wouda, Lan and Kool (2024)

7.4.2 Metaheuristic Lesson: Genetic Algorithm - VRP (Vehicle Routing Problem) - Valdecy Pereira

Pereira (2021) explora extensivamente a aplicação de algoritmos genéticos em VRPs, abordando conceitos como seleção, cruzamento, mutação e elitismo, que são fundamentais para o sucesso dessa meta-heurística na busca de soluções de alta qualidade.

De acordo com Pereira (2021), os algoritmos genéticos funcionam por meio da evolução de uma população inicial de soluções, avaliadas por uma função de *fitness* que determina sua adequação para o problema. Em cada geração, soluções mais “aptas” são selecionadas para reprodução, onde operações de cruzamento combinam informações de diferentes soluções para gerar descendentes com potencial para desempenho superior. A mutação, por outro lado, é aplicada a uma pequena fração das soluções, inserindo variação genética e prevenindo que o algoritmo se acomode em ótimos locais precocemente. Essa diversidade genética é essencial para garantir que o algoritmo explore novas áreas do espaço de busca, mantendo o equilíbrio entre exploração e intensificação.

O autor também destaca a implementação de elitismo, onde as melhores soluções de uma geração são preservadas para a próxima. Essa prática, comum em algoritmos genéticos, garante que a qualidade das soluções não seja comprometida ao longo das gerações, permitindo uma evolução estável e eficaz para encontrar soluções próximas do ótimo global.

Além das operações básicas, o trabalho discute a personalização dos parâmetros do algoritmo, como a taxa de mutação e o tamanho da população, para adaptar o GA a diferentes variantes de VRPs, como o VRP Capacitado (CVRP), o VRP com múltiplos depósitos (MD-VRP) e o VRP com janelas de tempo (VRPTW). Esse nível de flexibilidade é uma das razões pelas quais os algoritmos genéticos são amplamente aplicados em problemas de roteamento de veículos, pois permitem ajustar a busca conforme as necessidades específicas de cada problema.

Este trabalho, assim como o de outros autores renomados na área, como os de Holland (1975) e Goldberg (1989), contribui significativamente para o desenvolvimento de metaheurísticas que combinam eficiência e adaptabilidade, estabelecendo os algoritmos genéticos como uma das principais abordagens para a resolução de VRPs e suas variações.

Por fim, o resultado final de execução do modelo proposto por Pereira (2021) pode ser na figura 3.

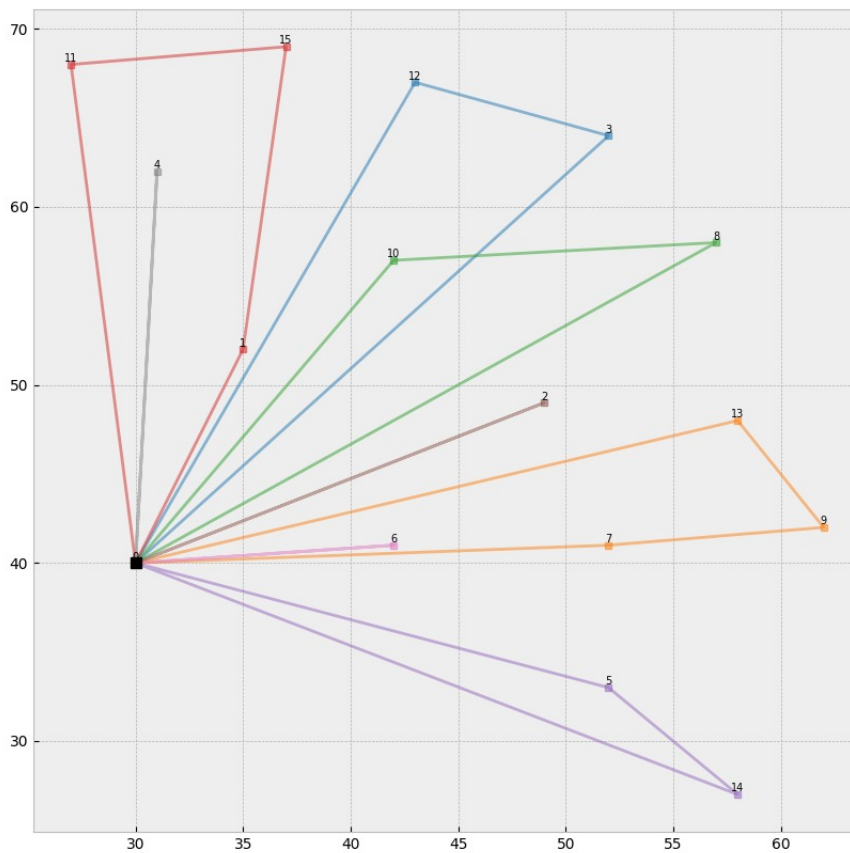


Figura 8 – Exemplo de saída do método proposto por Pereira (2021)

7.4.3 VRPy: A Python package for solving a range of vehicle routing problems with a column generation approach - Romain Montagné and David Torres Sanchez and Halvard Olsen Storbugt

O método proposto em VRPy oferece uma solução flexível para problemas de VRPs utilizando uma abordagem de geração de colunas. Esse método permite resolver diversas variantes do VRP, como o VRP Capacitado (CVRP), VRP com janelas de tempo (VRPTW) e VRP com coleta e entrega simultânea. O método foi desenvolvido com o objetivo de fornecer uma API intuitiva baseada na biblioteca NetworkX, facilitando a aplicação para pesquisadores e analistas de logística (MONTAGNÉ; SANCHEZ; STORBUGT, 2020).

A abordagem de geração de colunas utilizada envolve a resolução de dois subproblemas principais: o problema mestre e o problema de precificação. O problema mestre é uma formulação linear que particiona os clientes em subconjuntos ou rotas, enquanto o problema de precificação consiste em encontrar novas rotas viáveis que possam melhorar a solução atual.

Essa metodologia permite buscar soluções de alta qualidade, adaptando-se a diferentes restrições e requisitos específicos de cada variante de VRP (COSTA; CONTARDO; DESAULNIERS, 2019).

Montagné, Sanchez and Storbugt (2020) destacam a capacidade de VRPy de competir com outras ferramentas populares, como o OR-Tools do Google, em termos de qualidade de solução e facilidade de uso. A biblioteca foi projetada para resolver problemas de pequena a média escala, o que a torna uma escolha ideal para aplicações industriais e acadêmicas onde é necessária acessível. Além disso, a VRPy permite o uso de estratégias avançadas, como heurísticas de mergulho (SADYKOV et al., 2019a) e hiper-heurísticas para seleção dinâmica de estratégias de precificação, que ajustam automaticamente as configurações de parâmetros para otimizar o desempenho do algoritmo durante a execução.

O método segue evoluindo, com futuras melhorias planejadas para incrementar o desempenho e adicionar novas funcionalidades, consolidando-se como uma solução robusta para a comunidade de pesquisa operacional (MONTAGNÉ; SANCHEZ; STORBUGT, 2020).

7.5 RESULTADOS

Para comparar a eficiência dos modelos, foram selecionadas três métricas principais: custo na primeira solução, custo na melhor solução e tempo para encontrar a melhor solução. Essas métricas foram escolhidas por sua relevância em problemas de roteamento de veículos, uma vez que ajudam a avaliar tanto a qualidade inicial das soluções quanto a eficácia da otimização ao longo do processo de busca.

A métrica de custo na primeira solução é particularmente importante para avaliar a eficácia inicial dos algoritmos. Em problemas complexos uma boa primeira solução pode ser decisiva, especialmente em aplicações práticas onde o tempo é limitado e soluções rápidas são necessárias. No contexto de VRPy, Montagné, Sanchez and Storbugt (2020) destacam que uma solução inicial próxima ao ótimo pode facilitar o refinamento posterior através da geração de colunas. Comparativamente, os métodos de Wouda, Lan and Kool (2024) e Pereira (2021), que utilizam estratégias baseadas em algoritmos genéticos, buscam gerar uma população inicial de soluções variadas e de qualidade aceitável, facilitando o progresso da otimização.

A segunda métrica, custo na melhor solução, é essencial para avaliar a capacidade dos algoritmos de melhorar a qualidade das soluções ao longo do tempo. A busca pela melhor solução reflete o sucesso da metodologia em explorar o espaço de busca e evitar a convergência para ótimos locais. Esta métrica permite observar até que ponto os métodos conseguem refinar a solução inicial e se aproximar do ótimo global. Essa métrica é especialmente útil para avaliar o VRPy, que aprimora iterativamente a solução.

Por fim, o tempo para encontrar a melhor solução é uma métrica crítica em problemas

desta natureza, devido às restrições de tempo em aplicações reais. Esta métrica mede a eficiência de cada método, observando quanto tempo cada algoritmo precisa para alcançar a sua solução de melhor qualidade possível. A avaliação do tempo é fundamental para determinar a aplicabilidade prática de cada modelo, especialmente em contextos onde o tempo de resposta é um fator decisivo.

7.5.1 Métodos analisados

A escolha dos três modelos analisados neste trabalho – VRPy, pyVRP e pyVRP(Valdecy) – foi motivada pelas características específicas de cada um, que se mostraram promissoras para resolver o problema escolhido.

O VRPy foi selecionado por sua abordagem de geração de colunas, que é amplamente reconhecida por sua eficácia em resolver problemas de roteamento, especialmente para variantes do VRP com restrições de capacidade e janelas de tempo. A flexibilidade do VRPy e seu suporte para diferentes restrições de rota tornaram-o uma escolha adequada para análise em HFFVRPs.

O PyVRP, que implementa o método de busca genética híbrida (HGS), foi escolhido por sua capacidade de combinar a exploração global dos algoritmos genéticos com uma busca local intensiva. Essa combinação é particularmente útil para VRPs complexos e de grande escala, onde a flexibilidade para adaptar a busca é essencial. Além disso, o PyVRP permite personalização no gerenciamento da população e nas estratégias de cruzamento, facilitando o ajuste do modelo às necessidades do HFFVRP.

A escolha do Algoritmo Genético para VRP se deu por sua natureza adaptativa e sua capacidade de explorar vastos espaços de soluções, mantendo a diversidade populacional e evitando convergência prematura. Essa característica é fundamental para o HFFVRP, onde as configurações de frotas e as restrições variam amplamente. Além disso, os algoritmos genéticos são conhecidos por sua versatilidade e facilidade de adaptação, o que facilita a análise de resultados sob diferentes configurações e instâncias.

Outros métodos também foram testados. Um método baseado em aprendizado por reforço profundo, apresentado por Li et al. (2022) foi testado, porém o projeto ao qual tive acesso apresentava diversos erros e incompatibilidades, fazendo assim com que ele fosse eliminado do comparativo. Um segundo método baseado em uma rede de ponteiros em matriz 2D, descrito por Liu et al. (2024) também foi testado. Esse apresentou resultados muito incoerentes e fora do esperado para o projeto.

Por fim, a biblioteca para resolução de VRPs denominada VRPSolverEasy, de autoria de Errami et al. (2023) também foi descartada. Esta por sua vez apresentou diversos problemas para compilação e necessitava de um *solver* próprio de autoria dos mesmos, desmotivando assim seu uso neste trabalho.

Além destas considerações, vale destacar que a obtenção de acesso à códigos para resolver HFFVRPs na literatura é um desafiadora. Muitos métodos desenvolvidos não tem seus códigos fontes disponibilizados publicamente, o que limita a replicabilidade e análise comparativa entre diferentes abordagens. Como destacado por Wouda, Lan and Kool (2024), o PyVRP foi criado com a finalidade de preencher essa lacuna, sendo um dos poucos pacotes de código aberto que oferece suporte abrangente para VRPs no geral. Entretanto, a ausência de outros códigos prontos na literatura dificulta a comparação direta entre diferentes implementações de HFFVRPs, exigindo que pesquisadores implementem suas próprias versões de métodos complexos para realizar análises robustas.

7.5.2 Especificações dos experimentos

Como apresentado anteriormente, métodos como Montagné, Sanchez and Storbugt (2020) e Wouda, Lan and Kool (2024), por exemplo, possuem diferenças consideráveis entre si, resultando em abordagens diferentes na elaboração dos programas. Apesar das distinções, os programas para os métodos escolhidos foram desenvolvidos conforme o modelo apresentado a seguir:

- 1 Importe as bibliotecas do método selecionado
- 2 Defina a lista com as coordenadas de cada nodo
- 3 Defina a lista com as demandas para cada nodo
- 4 Defina os veículos e suas capacidades
- 5 Defina as distâncias entre os nodos
- 6 Configure o método
- 7 Execute o método
- 8 Imprima o resultado do método

Em pyVRP (WOUDA; LAN; KOOL, 2024) temos uma lista de tuplas para as coordenadas do depósito e dos clientes, uma lista simples para as demandas, veículos são criados através do modelo da biblioteca, distâncias são calculadas e armazenadas na forma de arcos e, por fim, configurou-se o tempo máximo de execução para 12 horas. Vale mencionar que pyVRP faz uso de paralelismo para acelerar operações de busca.

Na implementação de pyVRP por Pereira (2021) encontramos uma organização um pouco diferente dos demais. As coordenadas e parâmetros (para este trabalho o único parâmetro utilizado será a demanda) são carregadas através de arquivos separados. A matriz de distâncias é montada internamente pelo método, respeitando o uso de distâncias euclidianas entre dois pontos. Quanto a definição dos atributos dos veículos, destaca-se que este método permite a configuração de velocidade para os veículos, que foi mantida com o valor de uma unidade

para todos os veículos criados para que não tivesse impacto nos testes. O método apresentou melhores resultados com um tamanho de população igual a 100, uma taxa de mutação de 0.10 e um número de gerações de 2500.

Este método merece um destaque diante os demais pelo alto número de opções de configuração, permitindo assim testes mais refinados. Além disso, a saída deste modelo possui muito mais detalhes sobre as rotas e as demandas, gerando um relatório em formato *CSV* ao final.

Por fim, no método VRPy de Montagné, Sanchez and Storbugt (2020) temos nodos definidos por tuplas de coordenadas e demandas juntas, a criação de todos os nodos na biblioteca *DiGraph* e, em seguida, a criação de todas as arestas sendo que o custo de cada uma é representado por uma lista de custos por veículo. A capacidade dos veículos é informada em uma única lista de tamanho n , onde n é o número de veículos. Por recomendação dos autores, para instâncias grandes, recomenda-se que o parâmetro *greedy* seja ativado.

Em anexo, encontram-se de forma objetiva os parâmetros utilizados nos experimentos.

Quanto as instâncias de testes, foram escolhidas de diferentes *datasets* presentes na literatura. Optou-se, dado os resultados dos experimentos, de não ultrapassar um limite de 101 nodos (sendo 1 depósito e 100 clientes). Os *datasets* foram obitidos em <http://vrp.galgos.inf.puc-rio.br/index.php/en/>.

Para cada instância foi criado um conjunto de veículos com capacidades diferentes, formando assim uma frota fixa e heterogênea, conforme proposto pelo tema do trabalho. Os valores foram selecionados a partir do valor de capacidade fornecido pelos criadores destas mesmas instâncias. A relação entre clientes, número de veículos e capacidade é apresentada na tabela 1.

Clientes	Veículos	Capacidades
15	4	30
	4	40
18	1	120
	1	200
31	2	90
	2	100
	1	120
50	1	150
	1	200
	2	250
78	3	60
	2	80
	3	120
	2	140
100	1	400
	2	600
	2	800
	2	1000

Tabela 4 – Configuração de veículos e capacidades para diferentes quantidades de clientes

Além disso, destaca-se que as instâncias escolhidas apresentam as coordenadas dos nodos em duas dimensões, sendo definido no cabeçalho das instâncias que as distâncias precisam ser calculadas usando distância euclidiana entre dois pontos.

Quanto a execução dos experimentos, os programas especificados anteriormente foram executados em um único computador com as seguintes características:

- Processador AMD Ryzen 5 7600
- Memória RAM 32gb 6000mHz
- Placa de vídeo RTX 3070 8gb
- SSD NVMe 2tb 5000mb/s
- Fonte 850w 80 plus

Afim de se manter um controle sobre os testes, foi definido um limite de 12 horas (43200 segundos) para cada execução dos experimentos. Fora esse controle, todas as demais configurações seguiram conforme recomendações dos autores em seus trabalhos, como tamanho de população, número de gerações e modelo do *solver*, por exemplo.

7.5.3 Análise

A tabela 2 apresenta os resultados obtidos para análise neste trabalho, enquanto a tabela 3 apresenta os mesmos resultados porém adotando o modelo pyVRP de Wouda, Lan and Kool (2024) como valor de referência. Para fins de identificação, a implementação denominada pyVRP de Pereira (2021) teve seu nome substituído para pyVRP (UFF) em alusão a Universidade Federal Fluminense, universidade onde o autor atua. O outro pyVRP será identificado por pyVRP (WLK), que representam as iniciais dos nomes dos autores.

	Custo na primeira solução			Custo na melhor solução			Tempo para encontrar a melhor solução (s)		
	pyVRP (WLK)	pyVRP (UFF)	VRPy	pyVRP (WLK)	pyVRP (UFF)	VRPy	pyVRP (WLK)	pyVRP (UFF)	VRPy
15 Clientes	430	476	442	430	438	422	1	2	26
18 Clientes	205	384	959	205	212	196	1	12	9002
31 Clientes	743	2027	3740	743	841	883	1	152	43200
50 Clientes	474	1814	2402	459	541	588	2	226	43200
77 Clientes	1424	4357	7199	1342	1407	2009	4	267	43200
100 Clientes	13214	49861	90010	12956	13624	20448	5	2678	43200

Tabela 5 – Comparação entre pyVRP (WLK), pyVRP (UFF) e VRPy para diferentes métricas e tamanhos de clientes.

A tabela apresenta uma comparação detalhada entre os modelos pyVRP (WLK), pyVRP (UFF) e VRPy, utilizando métricas previamente definidas. Cada uma dessas métricas fornece detalhes importantes sobre a eficácia e eficiência dos modelos analisados. Vale salientar que os valores anotados na tabela para o pyVRP (UFF) correspondem ao menor custo na melhor

	Custo na primeira solução			Custo na melhor solução			Tempo para encontrar a melhor solução		
	pyVRP (WLK)	pyVRP (UFF)	VRPy	pyVRP (WLK)	pyVRP (UFF)	VRPy	pyVRP (WLK)	pyVRP (UFF)	VRPy
15 Clientes	430	+10.70%	+2.79%	430	+1.86%	-1.86%	1	+100.00%	+2500.00%
18 Clientes	205	+87.32%	+367.80%	205	+3.41%	-4.39%	1	+1100.00%	+900100.00%
31 Clientes	743	+172.84%	+403.36%	743	+13.17%	+18.85%	1	+15100.00%	+4319900.00%
50 Clientes	474	+282.28%	+406.33%	459	+17.87%	+28.33%	2	+11200.00%	+2159900.00%
78 Clientes	1424	+206.00%	+405.33%	1342	+4.84%	+49.78%	4	+6550.00%	+1079990.00%
100 Clientes	13214	+277.39%	+581.41%	12956	+5.16%	+57.77%	5	+53460.00%	+863900.00%

Tabela 6 – Comparação entre pyVRP (WLK), pyVRP (UFF) e VRPy tomando pyVRP como referência. solução obtido após 12 execuções para cada instância de teste, já que este modelo apresenta por natureza valores diferentes em sua solução a cada execução.

No que diz respeito ao custo na primeira solução, observa-se que o pyVRP (WLK) apresenta valores relativamente baixos em todas as instâncias, especialmente em instâncias menores (15 a 50 clientes). Este resultado é consistente com os achados de Wouda, Lan and Kool (2024), que destacam a eficiência de algoritmos genéticos híbridos para obter boas soluções iniciais em VRPs. Isso indica que o algoritmo consegue explorar rapidamente soluções de qualidade, o que é vantajoso em contextos onde a velocidade de resposta inicial é um fator crítico. O pyVRP (UFF) exibe custos iniciais significativamente mais elevados em comparação com o pyVRP (WLK), o que sugere que suas modificações podem estar otimizadas para buscar soluções mais refinadas em vez de soluções rápidas iniciais.

O modelo VRPy, que utiliza uma abordagem de geração de colunas, apresenta um custo inicial elevado em todas as instâncias, especialmente em problemas com mais de 30 clientes. Métodos baseados em geração de colunas são projetados para otimizar soluções ao longo do tempo, em vez de obter uma boa solução inicial, o que explica os resultados observados (MONTAGNÉ; SANCHEZ; STORBUGT, 2020). Essa abordagem pode ser útil em casos onde é necessário um refinamento contínuo e detalhado das rotas, embora, para instâncias menores, esse alto custo inicial possa ser uma desvantagem, especialmente em aplicações com restrições de tempo.

Em relação ao custo na melhor solução, o pyVRP (WLK) continua a se destacar nas instâncias menores, especialmente na de 15 clientes, onde alcança um custo final muito próximo do custo inicial. Esse comportamento sugere que o pyVRP (WLK) é capaz de encontrar uma solução de qualidade de maneira rápida e refinar a solução marginalmente, mantendo um alto desempenho em termos de custo total. No entanto, à medida que o número de clientes aumenta, o pyVRP (WLK) tende a perder sua vantagem, porém a limitação de tempo impede que VRPy chegue em resultados provavelmente melhores que os métodos com algoritmos genéticos. A instância de 15 clientes, por ser menor, reflete a capacidade do VRPy de melhorar progressivamente as soluções em problemas mais complexos. Essa vantagem do VRPy confirma o potencial dos algoritmos de geração de colunas em explorar o espaço de soluções de forma eficiente em escalas maiores, conforme observado por Montagné, Sanchez and Storbugt (2020).

O pyVRP (UFF), por sua vez, apresenta uma evolução significativa entre o custo inicial e o custo final. Esse comportamento está alinhado com as observações de Pereira (2021), que

ressalta que o uso de adaptações em algoritmos genéticos pode aprimorar a qualidade da solução ao longo do tempo, mesmo que o custo inicial seja alto. Em particular, nas instâncias de 75 e 100 clientes, o pyVRP (UFF) reduz substancialmente o custo final em comparação com o custo inicial, se destacando por sua capacidade de alcançar soluções de alta qualidade mesmo levando um pouco mais de tempo para convergir para o resultado final.

A análise do tempo para encontrar a melhor solução mostra diferenças grandes entre os três métodos. O pyVRP (WLK) é claramente o mais rápido, com tempos de resposta entre 1 e 5 segundos. Esse desempenho é coerente com a literatura, onde Wouda, Lan and Kool (2024) enfatizam a capacidade dos algoritmos genéticos híbridos de explorar rapidamente boas soluções iniciais. O pyVRP (UFF), em contrapartida, apresenta tempos de execução elevados nas instâncias maiores, atingindo até 2678 segundos para 100 clientes. Esses resultados sugerem que as adaptações realizadas podem estar priorizando a qualidade da solução em detrimento da velocidade, como observado por Pereira (2021), que aponta que melhorias incrementais nos algoritmos genéticos frequentemente exigem um tempo maior para convergir.

Por fim, o VRPy, que mantém um tempo fixo de 43200 segundos para as instâncias de 30 clientes em diante, utiliza esse período para maximizar o refinamento da solução, explorando amplamente o espaço de busca. Embora esse tempo seja alto, ele induz que o VRPy encontre a melhor solução possível dentro de um cenário onde o tempo não é um fator crucial, o que pode ser útil em situações onde a qualidade final da solução é o mais importante.

Nota-se também uma relação entre o a melhora da solução inicial para a melhor solução e o tamanho das instâncias. Com base na tabela 2 é possível observar que conforme o número de clientes aumenta, a diferença entre a primeira solução encontrada e a melhor solução tende a aumentar. Como o modelo pyVRP (WLK) já busca partir de uma solução inicial próxima ao ótimo Wouda, Lan and Kool (2024), este apresenta uma diferença pequena entre estes valores. Esta diferença é maior para pyVRP (UFF), que procura iniciar com soluções mais variadas e ir explorando para baixar esse valor. Para VRPy, como menciona Montagné, Sanchez and Storbugt (2020), um bom valor na solução inicial não é buscado, visto que pretende-se diminuí-lo de maneira iterativa ao longo do tempo.

Quanto as rotas, a figura 4 demonstra os resultados finais da execução para a instância de 31 clientes com os métodos pyVRP (WLK) e pyVRP (UFF), respectivamente. Nelas observamos a alta similaridade entre as rotas encontradas. O método VRPy não exibiu rotas de forma gráfica na saída e, como os valores de seus custos se mantiveram muito mais altos devido ao limite de tempo, optou-se por não apresentá-los. Os dois primeiros métodos mencionados apresentam resultados onde se pode procurar por pontos de não similaridade, visto que não muito parecidos. Já nos gráficos do modelo de VRPy procuraria-se por pontos de similaridade no meio de muitas sobreposições, reforçando assim que não agregaria ao trabalho exibi-los.

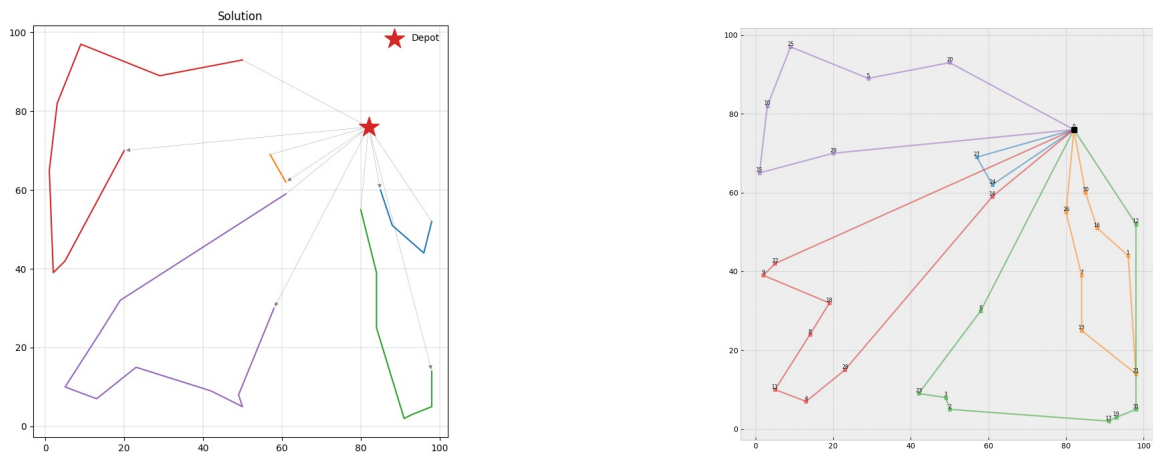


Figura 9 – Comparação de métodos para 31 clientes entre Wouda, Lan and Kool (2024) e Pereira (2021)

Para o exemplo com 31 clientes é possível observar algumas poucas diferenças entre as rotas descobertas, porém para a instância com 100 clientes as diferenças aumentam consideravelmente, principalmente pela diferença de 5,15% no custo entre os modelos, conforme visualizado na figura 5. Para esta instância maior observa-se que principalmente o reflexo desta diferença de custos está no enorme número de sobreposições de rotas usando o método pyVRP (UFF).

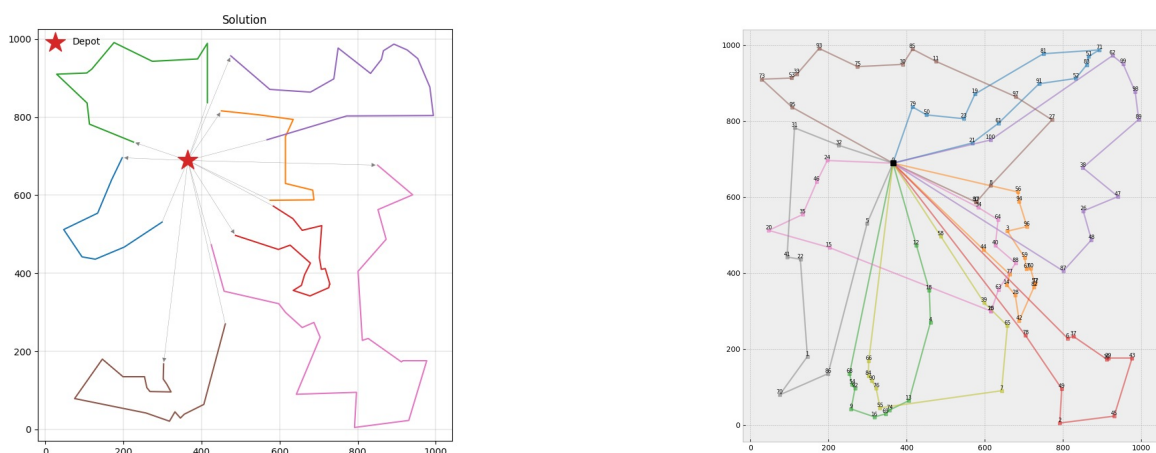


Figura 10 – Comparação de métodos para 100 clientes entre Wouda, Lan and Kool (2024) e Pereira (2021)

O tempo pode ser analisado através do gráfico da figura 6. Podemos observar que o método pyVRP (WLK), para o conjunto de instâncias utilizado, apresentou um comportamento próximo do linear. O método pyVRP (UFF), por sua vez, apresenta um comportamento um pouco diferente do método anterior, tendo um salto visível de tempo para a maior das instâncias.

Já o VRPy atingiu rapidamente o limite de tempo e manteve este comportamento até o fim.

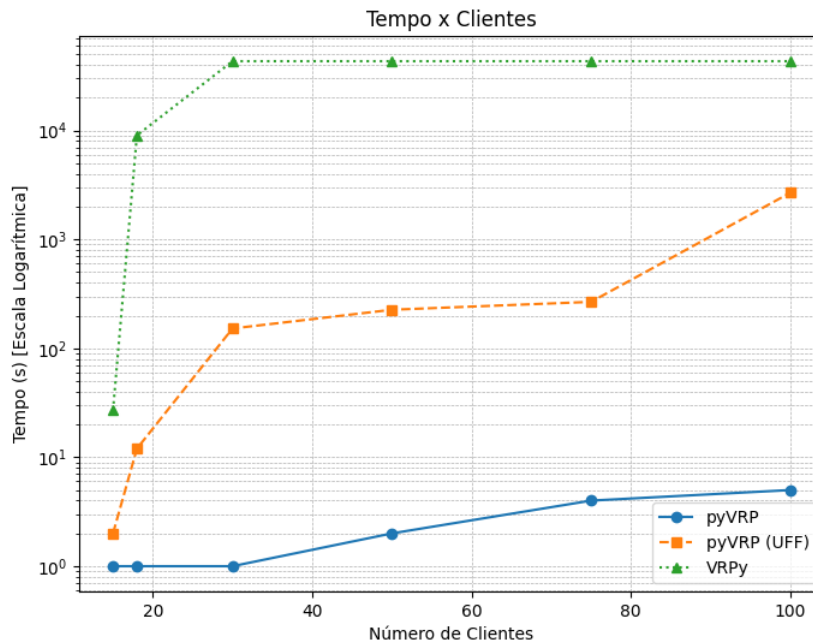


Figura 11 – Gráfico de tempo pelo tamanho das instâncias.

Em resumo, cada método possui características distintas que o tornam adequado para diferentes contextos de aplicação. O pyVRP (WLK) destaca-se pela velocidade e pelo custo competitivo, sendo indicado para problemas onde a rapidez é fundamental. O pyVRP (UFF), apesar do custo inicial maior, oferece soluções de alta qualidade e, como mencionado anteriormente, possui o fator chave de permitir um maior número de customizações do problema, sendo adequado para cenários onde o tempo de execução é secundário em relação à qualidade da solução final ou onde necessita-se modelar o problema com mais detalhes. O VRPy, com seu enfoque na geração de colunas, é mais apropriado para instâncias onde a qualidade da solução é o principal objetivo. Esses resultados confirmam as discussões presentes na literatura, onde a escolha do método depende do equilíbrio entre qualidade e tempo de execução, conforme destacado por Vidal et al. (2013).

7.6 CONCLUSÕES

O presente trabalho teve como objetivo comparar diferentes métodos de resolução aplicados ao problema de roteamento de veículos heterogêneos com frota fixa. Através dessa análise, buscou-se avaliar o desempenho e a adequação de cada abordagem em termos de eficiência e qualidade das soluções geradas, visando identificar os métodos mais promissores para esse tipo específico de problema de otimização.

O comparativo realizado evidenciou tanto os pontos fortes quanto as limitações de cada modelo testado. Em particular, foram identificadas abordagens com maior eficácia em certas condições e com desvantagens em outras, permitindo um entendimento mais detalhado das características de cada método e de como eles se comportam em diferentes cenários e restrições.

Observou-se que, em instâncias de menor porte, os métodos apresentaram o comportamento esperado, com resultados eficientes e soluções de boa qualidade. No entanto, à medida que as instâncias aumentaram em complexidade e tamanho, o método baseado em geração de colunas demonstrou limitações significativas, falhando em manter o desempenho e, muitas vezes, não conseguindo alcançar soluções competitivas em comparação com os outros métodos testados.

Desta forma, o método pyVRP (WLK) de Wouda, Lan and Kool (2024) demonstrou rapidez e eficiência para apresentar resultados, porém não conseguiu garantir os menores custos em todas as instâncias. Por sua vez, o método de pyVRP (UFF) apresenta resultados competitivos, porém levando um pouco mais de tempo que pyVRP (WLK). Como ponto positivo, destaca-se o grande número de personalizações possíveis no modelo de Pereira (2021), permitindo melhores modelagens. Por fim, VRPy apresenta o maior potencial para obter custos mais baixos, devido a seu método iterativo, tendo como ponto negativo apenas seu alto tempo de execução.

Os objetivos desse trabalho foram atingidos, tendo selecionado métodos para o problema de roteamento de veículos heterogêneos com frota fixa e comparado o desempenho destes métodos em cenários experimentais. Por fim, pretende-se dar publicidade a esse trabalho para que sirva de base a outros trabalhos na área de problemas de otimização.

7.6.1 Trabalhos futuros

Quanto a trabalhos futuros, podemos abordar os seguintes pontos:

- Realizar novos comparativos, utilizando os resultados obtidos neste trabalho como base para futuras comparações.
- Realizar testes com tempos de execução mais longos, visando explorar a capacidade dos algoritmos em encontrar soluções ótimas em instâncias complexas.
- Ampliar a avaliação dos algoritmos com instâncias variadas e dados de benchmark para melhorar a aplicabilidade e robustez em cenários reais.
- Auxiliar na formalização de uma solução para o problema, já que o comparativo realizado ajuda a entender pontos fortes e fracos de outros modelos, podendo ajudar no desenvolvimento.

Referências

- BALDACCI, R.; BATTARRA, M.; VIGO, D. **Routing a Heterogeneous Fleet of Vehicles**. [S.l.]: Springer, 2008.
- BLUM, C.; ROLI, A. Metaheuristics in combinatorial optimization: Overview and conceptual comparison. **ACM Computing Surveys**, v. 35, p. 268–308, 01 2001.
- BRANDAO, J. A tabu search algorithm for the heterogeneous fixed fleet vehicle routing problem. **Computers Operations Research**, v. 38, p. 140–151, 01 2011.
- COSTA, A.; OLIVERA, J. F.; TOTH, P. Heuristic and exact algorithms for vehicle routing problems with heterogeneous fixed fleets. **Operations Research Perspectives**, Elsevier, v. 6, p. 100116, 2019.
- COSTA, L.; CONTARDO, C.; DESAULNIERS, G. Exact branch-price-and-cut algorithms for vehicle routing. **Transportation Science**, v. 53, n. 4, p. 946–985, 2019.
- DANTZIG, G. B.; RAMSER, J. H. The truck dispatching problem. **Management Science**, v. 6, p. 1–140, 10 1959.
- DANTZIG, G. B.; WOLFE, P. Decomposition principle for linear programs. **Operations Research**, INFORMS, v. 8, n. 1, p. 101–111, 1954.
- DESAULNIERS, G.; DESROSIERS, J.; SOLOMON, M. Column generation. In: **Column Generation**. [S.l.]: Springer, 2005. p. 1–32.
- DONDO, R.; CERDÁ, J. A cluster-based optimization approach for the multi-depot heterogeneous fleet vehicle routing problem with time windows. **European Journal of Operational Research**, v. 176, p. 1478–1507, 02 2007.
- ERRAMI, N. et al. VRPSolverEasy: a python library for the exact solution of a rich vehicle routing problem. **Preprint**, 2023. Disponível em: <<https://inria.hal.science/hal-04057985v2>>.
- FERLAND, J. A.; MICHELON, P. The vehicle scheduling problem with multiple vehicle types. **The Journal of the Operational Research Society**, v. 39, p. 577–583, 06 1988.
- GLOVER, F.; LAGUNA, M. **Tabu search I**. [S.l.]: INFORMS Journal on Computing, 1999.
- GOLDBERG, D. E. **Genetic Algorithms in Search, Optimization, and Machine Learning**. Boston, MA: Addison-Wesley, 1989.
- GOLDEN, B. et al. The fleet size and mix vehicle routing problem. **Computers Operations Research**, v. 11, p. 49–66, 03 1984.
- HOFF, A. et al. Industrial aspects and literature survey: Fleet composition and routing. **Computers Operations Research**, v. 37, p. 2041–2061, 12 2010.
- HOLLAND, J. H. **Adaptation in Natural and Artificial Systems**. Ann Arbor, MI: University of Michigan Press, 1975.

- IRNICH, S. A multi-depot pickup and delivery problem with a single hub and heterogeneous vehicles. **European Journal of Operational Research**, v. 122, p. 310–328, 04 2000.
- KRITIKOS, M. N.; IOANNOU, G. The heterogeneous fleet vehicle routing problem with overloads and time windows. **International Journal of Production Economics**, v. 144, p. 68–75, 07 2013.
- LAPORTE, G. et al. Thirty years of heterogeneous vehicle routing. **European Journal of Operational Research**, v. 249, p. 1–21, 07 2015.
- LI, F.; GOLDEN, B.; WASIL, E. A record-to-record travel algorithm for solving the heterogeneous fleet vehicle routing problem. **Computers & Operations Research**, v. 34, p. 2734–2742, 09 2007.
- LI, J. et al. Deep reinforcement learning for solving the heterogeneous capacitated vehicle routing problem. **IEEE Transactions on Cybernetics**, IEEE, v. 52, n. 12, p. 13572–13584, 2022.
- LIU, Q. et al. 2d-ptr: 2d array pointer network for solving the heterogeneous capacitated vehicle routing problem. In: **Proceedings of the 23rd International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2024)**. Auckland, New Zealand: International Foundation for Autonomous Agents and Multiagent Systems (IFAAMAS), 2024. p. 1238–1247.
- MITCHELL, M. **An Introduction to Genetic Algorithms**. Cambridge, MA: MIT Press, 1998.
- MONTAGNÉ, R.; SANCHEZ, D. T.; STORBUGT, H. O. Vropy: A python package for solving a range of vehicle routing problems with a column generation approach. **Journal of Open Source Software**, The Open Journal, v. 5, n. 55, p. 2408, 2020. Disponível em: <<https://doi.org/10.21105/joss.02408>>.
- PEREIRA, V. **Metaheuristic Lesson: Genetic Algorithm - VRP (Vehicle Routing Problem)**. 2021. Available at <<https://www.researchgate.net/publication/353660717>>.
- ROTHLAUF, F. **Design of Modern Heuristics**. [S.l.]: Springer, 2011.
- SADYKOV, R. et al. Primal heuristics for branch and price: The assets of diving methods. **INFORMS Journal on Computing**, v. 31, n. 2, p. 251–267, 2019.
- SADYKOV, R. et al. Primal heuristics for column generation. **Computers & Operations Research**, Elsevier, v. 106, p. 1–16, 2019.
- TAILLARD, E. D. A heuristic column generation method for the heterogeneous fleet vrp. **RAIRO-Oper. Res**, v. 33, p. 1–14, 01 1999.
- TARANTILIS, C. D.; KIRANOUDIS, C. T. A meta-heuristic algorithm for the efficient distribution of perishable foods. **Journal of Food Engineering**, v. 50, p. 1–9, 10 2001.
- TARANTILIS, C. D.; KIRANOUDIS, C. T.; VASSILIADIS, V. S. A list based threshold accepting metaheuristic for the heterogeneous fixed fleet vehicle routing problem. **Journal of the Operational Research Society**, v. 54, p. 65–71, 03 2003.

TARANTILIS, C. D. et al. A reactive variable neighborhood tabu search for the heterogeneous fleet vehicle routing problem with time windows. **Journal of Heuristics**, v. 14, p. 425–455, 10 2007.

VIDAL, T. et al. A hybrid genetic algorithm with adaptive diversity management for a large class of vehicle routing problems with time-windows. **Computers & Operations Research**, Elsevier, v. 40, p. 475–489, 2013.

WILSON, R. **Graph Theory**. Dover Publications, 1996. ISBN 0582249937. Disponível em: <https://www.maths.ed.ac.uk/~v1ranick/papers/wilsongraph.pdf>.

WOUDA, N. A.; LAN, L.; KOOL, W. Pyvrp: A high-performance vrp solver package. **INFORMS Journal on Computing**, Institute for Operations Research and the Management Sciences (INFORMS), v. 36, n. 4, p. 943–955, 07 2024. ISSN 1526-5528. Disponível em: <http://dx.doi.org/10.1287/ijoc.2023.0055>.

8 ANEXOS

8.1 Código Fonte

O código fonte desenvolvido neste trabalho encontra-se no seguinte repositório público:
<https://github.com/ebsiqueira/TCC>

8.2 Parâmetros Utilizados

Método	Parâmetros
pyVRP	Nenhum
pyVRP (UFF)	penalty_value = 10000 population_size = 100 mutation_rate = 0.10 elite = 10 generations = 2500
VRPy	greedy = true

Tabela 7 – Parâmetros Utilizados por Método