



**UNIVERSIDADE ESTADUAL DO RIO GRANDE DO NORTE
UNIVERSIDADE FEDERAL RURAL DO SEMI-ÁRIDO
PROGRAMA DE PÓS-GRADUAÇÃO EM CIÊNCIA DA
COMPUTAÇÃO**



KERGIONALDO PIRES FERNANDES SOARES

**PROBLEMA DE ROTEAMENTO DE VEÍCULOS COM
ENTREGAS FRACIONADAS E DRONES: FORMULAÇÃO
MATEMÁTICA, METAHEURÍSTICAS E ANÁLISE
COMPARATIVA**

MOSSORÓ

2025

KERGIONALDO PIRES FERNANDES SOARES

**PROBLEMA DE ROTEAMENTO DE VEÍCULOS COM
ENTREGAS FRACIONADAS E DRONES: FORMULAÇÃO
MATEMÁTICA, METAHEURÍSTICAS E ANÁLISE
COMPARATIVA**

Dissertação apresentada ao Programa de Pós-Graduação em Ciência da Computação - associação ampla entre a Universidade Federal Rural do Semi-Árido e a Universidade Estadual do Rio Grande do Norte, para a obtenção do título de Mestre em Ciência da Computação.

Orientador: Prof. Dr. Dario José Aloise - UERN

Coorientador: Prof. Dr. Sebastião Emídio Alves Filho - UERN

MOSSORÓ

2025

© Todos os direitos estão reservados a Universidade do Estado do Rio Grande do Norte. O conteúdo desta obra é de inteira responsabilidade do(a) autor(a), sendo o mesmo, passível de sanções administrativas ou penais, caso sejam infringidas as leis que regulamentam a Propriedade Intelectual, respectivamente, Patentes: Lei nº 9.279/1996 e Direitos Autorais: Lei nº 9.610/1998. A mesma poderá servir de base literária para novas pesquisas, desde que a obra e seu(a) respectivo(a) autor(a) sejam devidamente citados e mencionados os seus créditos bibliográficos.

Catálogo da Publicação na Fonte.
Universidade do Estado do Rio Grande do Norte.

S676p Soares, Kergionaldo Pires Fernandes
PROBLEMA DE ROTEAMENTO DE VEÍCULOS COM
ENTREGAS FRACIONADAS E DRONES: FORMULAÇÃO
MATEMÁTICA, METAHEURÍSTICAS E ANÁLISE
COMPARATIVA. / Kergionaldo Pires Fernandes Soares. -
Mossoró, 2025.
64p.

Orientador(a): Prof. Dr. Dario Aloise.
Coorientador(a): Prof. Dr. Sebastião Alves Filho.
Dissertação (Mestrado em Programa de Pós-
Graduação em Ciência da Computação). Universidade do
Estado do Rio Grande do Norte.

1. Roteamento de Veículos. 2. Entregas Fracionadas.
3. Drones. 4. Heurísticas. 5. Metaheurísticas. I. Aloise,
Dario. II. Universidade do Estado do Rio Grande do Norte.
III. Título.

O serviço de Geração Automática de Ficha Catalográfica para Trabalhos de Conclusão de Curso (TCC's) foi desenvolvido pela Diretoria de Informatização (DINF), sob orientação dos bibliotecários do SIB-UERN, para ser adaptado às necessidades da comunidade acadêmica UERN.

KERGIONALDO PIRES FERNANDES SOARES

**PROBLEMA DE ROTEAMENTO DE VEÍCULOS COM
ENTREGAS FRACIONADAS E DRONES: FORMULAÇÃO
MATEMÁTICA, METAHEURÍSTICAS E ANÁLISE
COMPARATIVA**

Dissertação apresentada ao Programa de
Pós-Graduação em Ciência da Computação para
a obtenção do título de Mestre em Ciência da
Computação.

Aprovada em: 29 de Agosto de 2025

BANCA EXAMINADORA

Documento assinado digitalmente



DARIO JOSE ALOISE

Data: 30/08/2025 15:03:49-0300

Verifique em <https://validar.it.gov.br>

Prof. Dr. Dario José Aloise - UERN
Presidente

Documento assinado digitalmente



SEBASTIAO EMIDIO ALVES FILHO

Data: 02/09/2025 21:54:41-0300

Verifique em <https://validar.it.gov.br>

Prof. Dr. Sebastião Emídio Alves Filho - UERN
Primeiro Membro

Documento assinado digitalmente



FABIO FRANCISCO DA COSTA FONTES

Data: 02/09/2025 09:51:25-0300

Verifique em <https://validar.it.gov.br>

Prof. Dr. Fábio Francisco da Costa Fontes - UFERSA
Segundo Membro

Prof. Dra. Andréa Cynthia Santos - Université Le
Havre
Terceiro Membro

À minha família, pois é minha base e meu porto
seguro na vida e na minha formação acadêmica.

AGRADECIMENTOS

À Grande Inteligência Cósmica, pela possibilidade de grandes realizações.

À minha família, minha mãe, meu pai e meus irmãos, aos que recebi total apoio, suporte e incentivo.

À minha companheira, que também passou pela formação de mesma titulação, o mestrado acadêmico, em outra área.

Aos docentes do Programa de Pós-graduação em Ciência da Computação da UERN e UFERSA, especialmente aos membros orientadores desta formação de mestrado, meu orientador e meu coorientador, Dr. Dario José Aloise e Dr. Sebastião Emídio Alves Filho e aos demais membros da banca de defesa da qualificação e defesa.

Aos colegas de mestrado que cursaram as disciplinas em comum e com mesma orientação.

Aos colegas professores e demais profissionais da EEPJA, durante minha atuação profissional como professor de Informática.

Aos amigos, que também tiveram a mesma formação de título em áreas diferentes.

Um agradecimento a nível de crédito de grandes pesquisadores da literatura deste trabalho, os doutores Maurício Resende, Lin e Keld Helsgaun, pelas dúvidas sanadas sobre suas heurísticas e metaheurísticas via email e LinkedIn.

EPÍGRAFE

"O verdadeiro sinal de inteligência não é o conhecimento, mas a imaginação."

(Albert Einstein)

RESUMO

O Problema de Roteamento de Veículos (PRV), uma generalização do clássico Problema do Caixeiro Viajante (PCV), é amplamente estudado na literatura. Com aplicações em diversos setores, esses problemas de roteamento são os mais pesquisados na área de transportes e logísticas, mas abrangem muito outros campos de aplicação. Na última década, com o avanço das tecnologias dos Veículos Aéreos não Tripulados, os drones, os problemas clássicos como o PCV e o PRV ganharam novas variantes e aplicações. Empresas e organizações já têm investimento significativo na última década para garantir grandes vantagens com os drones. Atualmente os drones estão sendo aplicados nos mais diversos setores, possibilitando redução de custo e mobilidade. Com suas vantagens em relação à flexibilidade operacional e capacidade de acesso a locais de difícil alcance, pesquisas em Pesquisa Operacional, Otimização, Robótica e Inteligência Artificial estão em alta e só tendem a aumentar. O presente trabalho aborda o Problema do Roteamento de Veículos com Entregas Fracionadas e Drones para o caso dos serviços de entregas de mercadorias com a cooperação de caminhões e drones e propõe a formulação matemática e a implementação de um algoritmo genético com chaves aleatórias viciadas (BRKGA) com integração da Heurística LKH e da Metaheurística ILS-RVND para buscar soluções otimizadas, que minimizem as rotas e, conseqüentemente, o tempo das entregas. Este trabalho apresenta resultados bons promissores com a utilização das técnicas.

Palavras-chave: Roteamento de Veículos. Entregas Fracionadas. Drones. Heurísticas. Metaheurísticas

ABSTRACT

The Vehicle Routing Problem (VRP), a generalization of the classic Traveling Salesman Problem (TSP), is widely studied in the literature. With applications in various sectors, these routing problems are the most researched in the area of transportation and logistics, but they encompass many other fields of application. In the last decade, with the advancement of Unmanned Aerial Vehicle technologies, drones, classic problems such as the TSP and VRP have gained new variants and applications. Companies and organizations have already made significant investments in the last decade to secure major advantages with drones. Currently, drones are being applied in a wide variety of sectors, enabling cost reduction and mobility. With their advantages in relation to operational flexibility and the ability to access hard-to-reach locations, research in Operations Research, Optimization, Robotics, and Artificial Intelligence is booming and is only expected to increase. This paper addresses the Vehicle Routing Problem with Fractional Deliveries and Drones for the case of goods delivery services with the cooperation of trucks and drones and proposes the mathematical formulation and implementation of a genetic algorithm with biased random keys (BRKGA) with integration of the LKH Heuristic and the ILS-RVND Metaheuristic to seek optimized solutions that minimize routes and, consequently, delivery times. This work shows good outcomes with the used techniques.

Keywords: Vehicle Routing. Split Delivery. Drones. Heuristics. Metaheuristics.

LISTA DE FIGURAS

Figura 1 – Exemplo <i>Vehicle Routing Problem</i> (VRP)	20
Figura 2 – Modelos de asas dos VANTs	22
Figura 3 – Exemplo de Roteamento de Caminhão e Drone	24
Figura 4 – Os problemas e variantes	25
Figura 5 – Exemplo do <i>Split Delivery Vehicle Routing Problem with Drone</i> (SD-VRPD) com <i>split delivery</i> no cliente 4.	27
Figura 6 – Solução SDVRP com Cliente 5 como <i>distante</i>	36
Figura 7 – Solução SDVRPD onde o drone atende o cliente distante.	38
Figura 8 – Exemplo de população de cromossomos	40
Figura 9 – Dados da Instância: Hypercube	41
Figura 10 – Dinâmica entre Populações no BRKGA	42
Figura 11 – Fluxograma do BRKGA	42
Figura 12 – Codificação e Decodificação aplicado ao Problema do Caixeiro Viajante (PCV)	48
Figura 13 – Cálculo do valor fitness	49
Figura 14 – Cruzamento Parametrizado Uniforme	50
Figura 15 – Cálculo <i>fitness</i> das Rotas PRV	51
Figura 16 – Fluxograma deste Projeto	53

LISTA DE TABELAS

Tabela 1 – Comparação entre drone e caminhão	24
Tabela 2 – Trabalhos Problema do Roteamento de Veículos (PRV) com Entregas Fracionadas e/ou Drones	26
Tabela 3 – Comparação das formulações SDVRP e SDVRPD.	34
Tabela 4 – Localização e Demanda dos Nós	36
Tabela 5 – Comparação Quantitativa com Custos Reais	36
Tabela 6 – Resumo Comparativo em Escala para Diferentes Valores de α	38
Tabela 7 – Testes com Metaheurísticas	56
Tabela 8 – Comparação entre Metaheurísticas	56
Tabela 9 – Testes Computacionais <i>Split Delivery Vehicle Routing Problem</i> (SD- VRP) (sem drones)	57
Tabela 10 – Testes Computacionais SDVRPD (com drones)	57

LISTA DE ABREVIATURAS E SIGLAS

BRKGA	<i>Biased Random-Key Genetic Algorithm</i>
CVRP	<i>Capacited Vehicle Routing Problem</i>
FSTSP	<i>Flying Sidekick Traveling Salesman Problem</i>
IA	Inteligência Artificial
ILS	<i>Iterated Local Search</i>
LK	<i>Lin-Kernighan</i>
LKH	<i>Lin-Kernighan-Helsgaun</i>
MTSP	<i>Multiple Traveling Salesman Problem</i>
PCV	Problema do Caixeiro Viajante
PMCV	Problema dos Múltiplos Caixeiros Viajantes
PRV	Problema do Roteamento de Veículos
RKGA	<i>Random-Key Genetic Algorithm</i>
RVND	<i>Randomized Variable Neighborhood Descent</i>
SDVRP	<i>Split Delivery Vehicle Routing Problem</i>
SDVRPD	<i>Split Delivery Vehicle Routing Problem with Drone</i>
TSP	<i>Traveling Salesman Problem</i>
TSP-D	<i>Traveling Salesman Problem with Drone</i>
UAV	<i>Unmanned Aerial Vehicles</i>
VANT	Veículo Aéreo Não Tripulado
VND	<i>Variable Neighborhood Descent</i>
VNS	<i>Variable Neighborhood Search</i>
VRP	<i>Vehicle Routing Problem</i>
VRPD	<i>Vehicle Routing Problem with Drones</i>

SUMÁRIO

1	INTRODUÇÃO	14
1.1	Motivações	16
1.2	Objetivo geral	16
1.3	Objetivos específicos	17
1.4	Organização do Trabalho	17
2	REVISÃO DA LITERATURA	18
2.1	Problema do Caixeiro Viajante	18
2.2	Problema de Roteamento de Veículos	19
2.3	<i>Split Delivery Vehicle Routing Problem</i>	20
2.4	Veículos Aéreos Não Tripulados	21
2.5	Trabalhos relacionados	22
2.5.1	<i>Split Delivery Vehicle Routing Problem with Drones</i>	26
3	FORMULAÇÕES MATEMÁTICAS	28
3.1	Formulação do Split Delivery Vehicle Routing Problem (SDVRP) .	28
3.1.1	<i>Conjuntos e Parâmetros</i>	28
3.1.2	<i>Variáveis de Decisão</i>	28
3.1.3	<i>Formulação Matemática Completa do SDVRP</i>	28
3.1.4	<i>Descrição Detalhada das Restrições do SDVRP</i>	29
3.2	Formulação do Split Delivery Vehicle Routing Problem with Drones (SDVRPD)	30
3.2.1	<i>Conjuntos e Parâmetros Adicionais</i>	30
3.2.2	<i>Variáveis de Decisão Adicionais</i>	30
3.2.3	<i>Formulação Matemática Completa do SDVRPD</i>	30
3.2.4	<i>Descrição Detalhada das Restrições do SDVRPD</i>	32
3.3	Análise Comparativa das Restrições: SDVRP vs. SDVRPD	33
3.4	Exemplo Numérico e Análise Comparativa	35
3.4.1	<i>Configuração do Problem</i>	35
3.4.2	<i>Solução SDVRP (Apenas Caminhões)</i>	35
3.4.3	<i>Solução SDVRPD (Caminhões + Drones)</i>	35
3.4.4	<i>Análise Comparativa</i>	35

3.4.5	<i>Análise de Sensibilidade do Fator de Custo α</i>	37
3.4.6	<i>Conclusão Final da Análise Comparativa</i>	37
4	HEURÍSTICAS E METAHEURÍSTICAS	39
4.1	Algoritmo Genético	39
4.2	<i>Biased Random-Key Genetic Algorithm</i>	39
4.3	<i>Lin-Kernighan-Helsgaun</i>	42
4.4	ILS-RVND	44
5	METODOLOGIA	46
5.1	Instâncias	46
5.2	Representação de solução	47
5.3	BRKGA aplicado ao Problema do Caixeiro Viajante	47
5.4	Aplicação ao Problema de Roteamento de Veículos	50
5.5	<i>Aplicação ao Prblema de Roteamento de Veículos com Entregas Fra-</i> <i>cionadas</i>	51
5.6	Aplicação ao Problema de Roteamento de Veículos com Drones . .	52
6	RESULTADOS	54
6.1	Ambiente de testes	54
6.2	Dados e Parâmetros	54
6.3	Testes computacionais	55
7	CONCLUSÕES	59
	REFERÊNCIAS	60

1 INTRODUÇÃO

O Problema do Roteamento de Veículos (PRV) tem sido muito abordado na literatura nos últimos anos devido ao grande crescimento das inovações tecnológicas na área de transportes nos vários setores. Tal crescimento indica um grande número de possibilidades com a evolução das várias alternativas de tecnologias como a *internet* e conexões de rede, sensoriamento e Inteligência Artificial (IA), mapeamento e os Veículos Aéreos Não Tripulados Veículo Aéreo Não Tripulado (VANT), os famosos drones. Nos últimos anos, a grande busca pelo desenvolvimento tecnológico fez várias organizações científicas e empresas investirem na produção de *softwares* e *hardwares* de alta performance para facilitar e automatizar várias tarefas nos veículos e, inclusive, torná-los autônomos (KONSTANTAKOPOULOS *et al.*, 2022).

Os drones, que podem ser tanto autônomos quanto controlados remotamente, foram desenvolvidos, primeiramente, para aplicações militares. Porém, ultimamente, o desenvolvimento desses veículos aéreos tem sido um dos grandes investimentos das empresas para o setor de logística e entrega de mercadorias. Uma das companhias pioneiras na aplicação de drones para a entrega de mercadorias é a *Amazon*, que inovou ao lançar o serviço *Amazon Prime Air*, no qual os drones começaram a realizar entregas dos produtos vendidos saindo diretamente de armazéns da empresa até o cliente. Posteriormente, a *Google* lançou o serviço *Wing* e a *Walmart* e outras empresas também começaram a aplicar seus próprios serviços (MOHSAN *et al.*, 2022a).

Algumas pesquisas começaram a introduzir uma nova abordagem na literatura dos problema de roteamento. Os serviços de entregas de mercadorias ainda utilizam a forma tradicional no nível urbano: transportes terrestres, como caminhões e outros tipos, realizam as entregas diretamente aos clientes. A nova abordagem das pesquisas adiciona a utilização de drones aos serviços de entregas convencionais com transportes terrestres. Essa abordagem torna esses serviços híbridos e o PRV recebe uma nova variante, PRV com Drones (WANG *et al.*, 2017).

O PRV é baseado no clássico Problema do Caixeiro Viajante (PCV), amplamente estudado nas áreas de Pesquisa Operacional, IA, Otimização Combinatória e Teoria dos Grafos. A definição do PCV consiste em: um conjunto de cidades ou localizações; um conjunto de distâncias entre as cidades. Em um grafo, esses conjuntos são representados por vértices e arestas. Considerando todas as cidades e todas as distâncias entre cada

uma delas, o problema consiste em encontrar a rota de menor distância ou custo na qual, partindo de uma cidade inicial, cada uma das demais cidades seja visitada uma única vez, voltando à inicial, finalizando o percurso. Essa rota, que passa por todas as cidades e volta à inicial é classificada na Teoria dos Grafos como um ciclo hamiltoniano. Isso significa o problema tem o objetivo de encontrar, dentre todos os possíveis ciclos hamiltonianos, o de menor custo (POP *et al.*, 2023).

O PCV é amplamente estudado e abordado na literatura por ser um problema simples de definir, porém difícil de solucionar, sendo classificado como um problema *NP*-Difícil. Problemas com essa classificação podem ser solucionados com métodos exatos, porém o tempo necessário para encontrar uma solução ótima dependerá da quantidade de cidades/locais utilizadas, que significa o tamanho de um caso ou instância. No caso do PCV, a partir de uma certa quantidade de cidades, o tempo para encontrar uma solução ótima pode se tornar muito longo, o que demanda alto custo computacional e torna essa proposta inviável quando se necessita de resultados instantâneos para uma solução.

Considerando que buscar por soluções ótimas para problemas que demandam alto custo computacional (processamento/tempo), se torna algo inviável, dependendo das dimensões, algumas técnicas e abordagens foram propostas ao longo das últimas décadas para buscar soluções aproximativas em tempo viável. As técnicas propostas são classificadas na Otimização e na IA como Heurísticas e Metaheurísticas, que são algoritmos que buscam as melhores soluções com o mínimo de tempo possível. Várias Heurísticas e Metaheurísticas foram propostas com diferentes estratégias e configurações para serem aplicadas aos mais variados problemas, principalmente no campo da Otimização Combinatória. Muitas dessas técnicas, principalmente as Metaheurísticas, são propostas generalistas. Isso significa que podem ser aplicadas a uma grande variedade de problemas, não sendo necessárias configurações ou personalizações complexas na adaptação dos algoritmos aos problemas.

No contexto de problemas de Otimização Combinatória, as possíveis soluções para cada problema são representadas por combinações ou sequências do *dataset* selecionado, que é a instância com os dados de um caso específico. Cada solução com uma determinada sequência tem um custo ou valor associado. Esse valor será utilizado pela função objetivo de um algoritmo para decidir se a solução é uma boa candidata a melhor ou não. O algoritmo irá comparar cada solução com as às últimas encontradas, percorrendo todas as

possibilidades dentro de uma condição especificada, até definir qual é a melhor solução.

Quando as instâncias são muito pequenas e são geradas poucas possíveis combinações, algoritmos exatos entregam a solução ótima em tempo viável. Mas quando as instâncias aumentam de dimensão, o tempo para encontrar uma solução ótima pode ser muito longo, a ponto dos algoritmos exatos não serem uma boa opção. Dessa forma, a utilização dos algoritmos heurísticos e metaheurísticos é importante pois, mesmo que não seja possível encontrar as melhores soluções para os problemas, podem técnicas podem ser configuradas e improvisadas para encontrar soluções aproximativas ou sub-ótimas, em um tempo viável.

1.1 Motivações

Com os grandes avanços nos últimos anos das tecnologias de roteamento, a evolução no desenvolvimento dos drones e as inovações nos serviços de entrega de mercadorias, o presente trabalho busca contribuir para o ramo dos serviços de entregas que pretendem realizar ou otimizar o trabalho híbrido com transportes e drones. Tornando-se uma tecnologia cada vez mais pesquisada e aplicada nos mais variados setores, os drones representam um dos maiores investimentos atuais, investimento esse motivado pelas suas possibilidades de alta redução de custo de operações e de tempo. Além desses aspectos, as questões ambientais sobre os impactos desse modelo híbrido com transportes e drones são de extrema relevância nas discussões acadêmicas, pois tal modelo permite a redução das emissões de carbono (CO₂) na atmosfera em comparação com o modelo sem a utilização dos drones. Pesquisas que resultam na possibilidade de melhores operações nos termos de custo e tempo são de alta relevância na Pesquisa Operacional e otimizar os serviços de entregas de mercadorias beneficia tanto profissionais e companhias da área quanto os clientes com a melhoria dessas operações. São objetivos deste trabalho buscar melhores resultados para melhorar as operações desses serviços.

1.2 Objetivo geral

O atual trabalho tem como objetivo o desenvolvimento de uma solução aplicada ao Problema de Roteamento de Veículos com Entregas Fracionárias e Drones para otimizar as rotas em que caminhões e drones realizam as visitas aos clientes no contexto de entregas

fracionadas. A proposta de solução se baseia em um *framework* com abordagem híbrida de Metaheurísticas e Heurísticas, como: o BRKGA, LKH e ILS-RVND.

1.3 Objetivos específicos

- Descrever o Problema de Roteamento de Veículos com Entregas Fracionadas e Drones, junto aos problemas relacionados e a literatura;
- Desenvolver algoritmo BRKGA e aplicar ao PCV;
- Gerar soluções PRV com o BRKGA;
- Otimizar as soluções do BRKGA com LKH;
- Otimizar as soluções PRV do LKH para soluções com Entregas Fracionadas com o ILS-RVND;
- Otimizar as soluções SDVRP para adicionar os drones com LKH;
- Comparar soluções das instâncias com os trabalhos relacionados.

1.4 Organização do Trabalho

Este trabalho está organizado como a estrutura a seguir: No capítulo 2 REVISÃO DA LITERATURA, onde são apresentados aspectos teóricos sobre os problemas abordados e que baseiam o problema foco, bem como os trabalhos relacionados; no capítulo 3 HEURÍSTICAS E METAHEURÍSTICAS, em que serão apresentados os algoritmos metaheurísticos e as Heurísticas de busca local utilizados como técnicas; no capítulo 4 METODOLOGIA, que serão descritas as etapas para as implementações dos algoritmos aos problemas e os passos a serem seguidos para a aplicação do contexto geral; no capítulo 5 RESULTADOS, que serão realizados os testes da implementação com as instâncias e as comparações dos resultados entre trabalhos; e no capítulo 6 CONCLUSÕES, que serão descritos os objetivos alcançados, as contribuições deste trabalho e possíveis trabalhos futuros.

2 REVISÃO DA LITERATURA

Neste capítulo serão feitas abordagens teóricas sobre os problemas em que se baseiam as aplicações reais e os trabalhos relacionados.

2.1 Problema do Caixeiro Viajante

O Problema do Caixeiro Viajante (PCV) é um dos problemas clássicos mais abordados na literatura em várias áreas, como Pesquisa Operacional, Teoria dos Grafos e Inteligência Artificial. É descrito abordado na literatura em inglês como *Traveling Salesman Problem* *Traveling Salesman Problem* (TSP) e há uma variedade de pesquisas que abordam suas aplicações e suas variantes, durante várias décadas até a atualidade, muitas sendo descritas em Cheikhrouhou e Khoufi (2021), Osaba *et al.* (2020) e Pop *et al.* (2024).

O PCV é um dos principais problemas de aplicação da Otimização Combinatória, campo de estudo presente nas áreas citadas, englobando a Matemática Aplicada e Ciência da Computação. Problemas desse campo determinam combinações ou sequências na ordem dos elementos de conjuntos que representam abstrações de situações reais e são estudados com o objetivo de otimizar tais combinações para encontrar as opções de melhor custo (APPLEGATE, 2006). O PCV, então, é definido como um conjunto de cidades e um conjunto de distâncias entre cada par dessas cidades, representadas, respectivamente, por vértices e arestas. São utilizados elementos da Teoria dos Grafos para abstrair o problema para que possam ser aplicadas as técnicas de otimização.

O PCV ganhou bastante atenção com a Pesquisa Operacional, que surgiu no contexto militar da 2ª Guerra Mundial para buscar as melhores alternativas em questões de tempo, custo e recursos para apoiar as decisões de gerenciamento (WINSTON, 2004). Depois do contexto militar, outros setores passaram a receber atenção da Pesquisa Operacional. O PCV é abordado em definições, aplicações, complexidade, além de histórico, por vários autores: (APPLEGATE, 2006), (DANTZIG, 2016), (KARP, 2010). Além disso, foi provado ser um problema de complexidade *NP*-Difícil (GAREY; JOHNSON, 1979).

Com o passar dos anos, a necessidade de abstrair situações reais mais complexas para outros contextos, como o setor privado, fez com que o PCV e outros problemas ganhassem variantes, como o problema de roteamento descrito por Dantzig *et al.* (1954), resultados das novas pesquisas (ILAVARASI; JOSEPH, 2014). Surgiu, para o PCV, uma

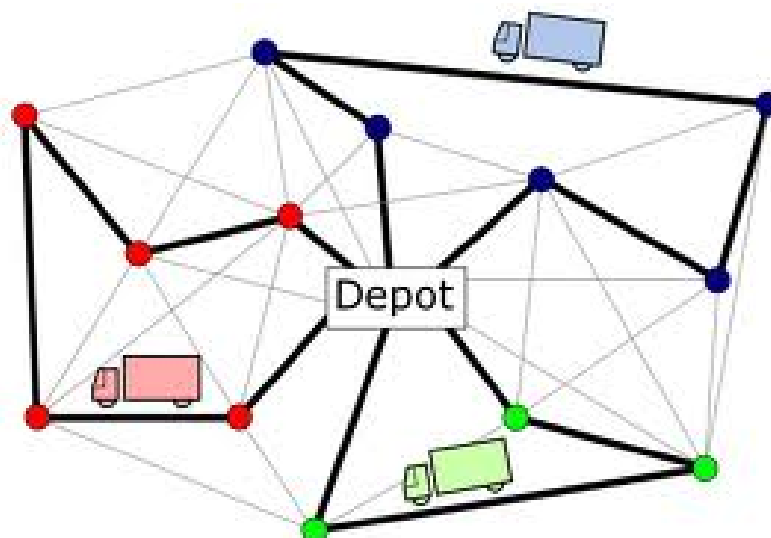
abordagem em que foram considerados mais de um caixeiro viajante para realizar as visitas em um dado conjunto de cidades, sendo essa variante chamada de Problema de Múltiplos Caixeiros Viajantes (BEKTAS, 2006). Nessa variante do problema a as visitas às cidades, dois ou mais caixeiros viajantes dividem o total de cidades entre eles, assumindo uma sub-rota em que um determinado conjunto de cidades do total será atribuído a cada um. Se tornando uma generalização do PCV, o Problema de Múltiplos Caixeiros Viajante (PMCV), do inglês *Multiple Traveling Salesman Problem* (MTSP), têm a mesma definição de conjuntos, adicionando apenas a configuração de m rotas para m viajantes (CHEIKHROUHOU; KHOUI, 2021).

2.2 Problema de Roteamento de Veículos

O Problema do Roteamento de Veículos (PRV), do inglês *Vehicle Routing Problem* (VRP), é um problema muito abordado na literatura, principalmente, aplicado ao setor robótica (MACHARET; CAMPOS, 2018) e de logística (KONSTANTAKOPOULOS *et al.*, 2022). O problema é, comumente, associado ao Problema dos Múltiplos Caixeiros Viajantes (PMCV). Inicialmente formalizado por Dantzig e Ramser (1959), o PRV é uma generalização do PMCV e do PCV. Lenstra e Kan (1981) definiram, através de formulações, que o PRV, como uma generalização do PCV e PMCV, é um problema NP-Difícil. Recentes pesquisas abordam suas variantes e aplicações, como Elshaer e Awad (2020) e Sarbijan e Behnamian (2023). O PRV adiciona ao PMCV as definições de capacidades e demandas. Para sua definição, recebeu, pelos próprios autores, formalizações e soluções propostas com modelos matemáticos e algoritmos.

O PRV é definido como um conjunto de clientes ou locais a serem visitados e um centro de distribuição, ou depósito. A representação em um grafo se configura com um conjunto V de n vértices em que cada vértice representa um dos locais, esses locais representando o depósito e os demais $(n - 1)$, os clientes. O problema consiste em disponibilizar, a partir do depósito, uma quantidade de m veículos com capacidades totais definidas para realizar entregas nos clientes, cujos têm demandas associadas. O objetivo do problema é encontrar, a partir dessa quantidade de veículos e suas capacidades, rotas de custos mínimos utilizando a quantidade mínima necessária de veículos para que visitem e atendam às demandas de todos os clientes. Uma representação pode ser vista na figura 1.

Figura 1 – Exemplo VRP



Fonte: (HELSGAUN, 2017)

2.3 *Split Delivery Vehicle Routing Problem*

Assim como o PRV, o SDVRP (PRV com Entregas Fracionárias, em tradução literal) é definido com os conjuntos de vértices, indicando o depósito e os clientes, bem como as demandas relativas aos clientes e a frota de veículos, indicando suas capacidades. Primeiramente descrito nos trabalhos de Dror e Trudeau (1989), Dror e Trudeau (1990), a grande diferença nesta variação do PRV está no relaxamento da restrição em que um cliente deve ser atendido por apenas um veículo. No SDVRP, cada cliente pode ser atendido por mais de um veículo. Ou seja, dois ou mais veículos podem compartilhar um mesmo cliente em suas rotas. A esse compartilhamento de clientes entre os veículos, dá-se o nome de entregas fracionadas, o que significa que a demanda de um cliente pode ser fracionada (dividida) entre os veículos.

Assim, como os autores mostraram (DROR; TRUDEAU, 1989), em certos conjuntos de rotas (certos casos), a possibilidade de fracionar as demandas de clientes é de extrema importância para a redução de custos. O fracionamento da demanda permite que esses clientes possam ser visitados por mais de um veículo, o que pode levar a uma redução da distância total, quando os custos das distâncias das sub-rotas são somadas. A explicação para essa possibilidade de redução vem da otimização da utilização ao máximo possível das capacidades de cada veículo, seja em frota homogênea ou heterogênea.

Em um conjunto de sub-rotas do *Capacited Vehicle Routing Problem* (CVRP), como a solução ótima de uma sub-rota segue a restrição de que a soma das demandas dos

clientes atendidos nessa sub-rotas não deve ultrapassar a capacidade total do veículo, há muitos casos em que a capacidade total não chega a ser completamente utilizada, o que deixa uma capacidade livre restante. O SDVRP possibilita que clientes que não foram atendidos por uma sub-rotas no CVRP possam ser candidatos a serem incluídos nessa sub-rotas em que o veículo tem capacidade livre, podendo atender a uma fração da demanda do candidato.

Um bom cliente candidato a ser incluído em uma sub-rotas, levando também em consideração sua posição, terá uma fração de sua demanda adicionada dentro do limite da capacidade livre de um veículo. Ao ser realizada, essa operação resulta, consequentemente, na redução de igual valor da fração na demanda atendida pelo veículo na sub-rotas original do cliente. A fração que é adicionada à capacidade de outro veículo é subtraída do veículo original. Esse processo é repetido várias vezes para tentar ocupar o máximo possível de capacidade livre entre os veículos com frações das demandas dos clientes. Como esse processo tende a preencher ao máximo possível a capacidade de cada veículo, as demandas totais de alguns veículos diminuem, o que, em algum momento, esses últimos podem ter capacidade livre para atender um ou mais clientes. Isso possibilita remover clientes de rotas específicas e adicioná-los às rotas desses veículos com espaço livre de demanda, o que implica na redução do custo daquelas rotas.

Ao final de várias tentativas de fracionamento de demanda dentre as sub-rotas de uma instância, o cálculo da distância total é feito. Caso essa distância seja menor que a distância original sem o fracionamento, a nova rota total é aceita como solução atual. O processo completo poderá ser repetido várias vezes, a depender das heurísticas utilizadas e dos parâmetros de iterações ou de tempo.

O SDVRP, com sua flexibilidade, vem sendo estudado ao longo dos últimos. Vários trabalhos de com propostas de heurísticas e de aplicações são bem citados na literatura.

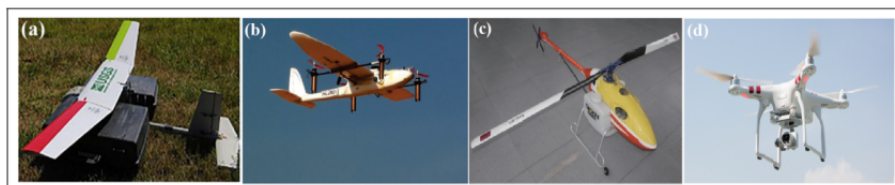
2.4 Veículos Aéreos Não Tripulados

Os drones estão em alta nas aplicações de vários setores devido à atenção de pesquisas científicas recentes na última década (MOHSAN *et al.*, 2022b). Geralmente, o termo “drone” veio para substituir termos como “controlado remotamente” para relacionar a veículos sem tripulante, sejam eles terrestres, aéreos ou aquáticos. Porém, na literatura dos drones, a maioria das pesquisas utiliza a definição de Veículos Aéreos Não Tripulados

(VANT), do inglês *Unmanned Aerial Vehicles* (UAV), termo que começou a ser amplamente utilizado a partir dos anos 1990 (NEWCOME, 2004). Newcome (2004) atribui o nascimento do conceito de VANTs ao famoso inventor Nikola Tesla.

Nas suas classificações, VANTs podem ser caracterizados como modelos de asas fixas ou de asas giratórias, chamadas de hélices. Os VANTs de asas fixas podem chegar a altas velocidades e são utilizados, em geral, no contexto militar (LAGHARI *et al.*, 2023). Já os UAVs com asas giratórias podem realizar os mesmos movimentos que helicópteros, como ter movimento estacionário no ar e movimentação espacial livre, possibilitando pouso e decolagem vertical. Exemplos são mostrados na figura 2: a) asa fixa, b) híbrido, c) 1 asa giratória, d) 4 asas giratórias.

Figura 2 – Modelos de asas dos VANTs



Fonte: (MOHSAN *et al.*, 2022a)

Nos últimos anos os drones têm sido aplicados a uma gama de novas atividades para vários setores: científico, industrial, comercial, etc. (SARBIJAN; BEHNAMIAN, 2023). Dois dos setores que mais aplicaram drones foi o de logística de entregas de mercadorias (*delivery*) e de comidas, fortemente influenciado pelas empresas de *e-commerce* e de aplicativos de entregas (BENARBIA; KYAMAKYA, 2021). Tais aplicações se beneficiam das vantagens dos drones: podem ser pequenos; têm mobilidade aérea, não limitados às formas tradicionais de locomoção; têm menos necessidade de desviar de obstáculos; e rapidez (DORLING *et al.*, 2017). Mesmo não tendo alta autonomia de bateria e tendo capacidade limitada a transportar objetos leves, são uma ótima possibilidade de redução de custos e demais vantagens. Singhal *et al.* (2018) demonstra as comparações entre os drones e outros veículos aéreos e mostra várias possíveis aplicações.

2.5 Trabalhos relacionados

A seguir são apresentados os trabalhos relevantes sobre VRP que abordam Entregas Fracionadas e/ou a cooperação com drones.

Há uma variedade de estudos que aplicam métodos exatos e soluções com heu-

rísticas e metaheurísticas para a resolução do SDVRP. Os trabalhos pioneiros de Dror e Trudeau (1989), Dror e Trudeau (1990) implementaram uma heurística de busca local ao problema. Uma nova aplicação com implementação de heurística só veio, depois de muito tempo, com o trabalho de Archetti *et al.* (2006), que utilizou o algoritmo *Tabu Search* para a resolução do problema, além da formulação de um modelo matemático com programação linear.

O trabalho de Silva *et al.* (2015), sobre o SDVRP, implementou um algoritmo híbrido de busca local baseado na metaheurística *Iterated Local Search* (ILS), proposta por Lourenço *et al.* (2003) e na *Variable Neighborhood Descent* (VND) (MLADENOVIC; HANSEN, 1997) com ordem aleatória de vizinhança. Essa abordagem de busca local híbrida com *Randomized Variable Neighborhood Descent* (RVND), chamada de ILS-RVND, foi proposta no trabalho de Subramanian *et al.* (2010), sendo uma extensão do trabalho anterior (SUBRAMANIAN *et al.*, 2008), que abordou apenas ILS com VND. Sua eficiência tem sido mostrada, além do original, em trabalhos relevantes, como Silva *et al.* (2012) e Penna *et al.* (2013).

Alguns trabalhos mais recentes apresentam bastante relevância. Gschwind *et al.* (2019) propõem um algoritmo *Branch-Price-and-Cut* para otimizar o SDVRP com restrições de *commodity*. O trabalho de Gouveia *et al.* (2023) apresenta novos modelos e um algoritmo *Branch-and-Cut* para o problema SDVRP com múltiplos depósitos, *Multi Depot SPVRP*. Ji *et al.* (2021) aplicaram um algoritmo de busca em vizinhança para o SDVRP com restrições de carregamento bi-dimensional. O trabalho de Wu e Wu (2022) desenvolve uma abordagem complexa do SDVRP com múltiplas janelas de tempo para o setor de produção agrícola. Ren *et al.* (2023) aplicam a metaheurística *Ant Colony Optimization* melhorada para o SDVRP com *delivery and pick-up*. Um recente trabalho (LIN *et al.*, 2025) implementa uma extensão da ILS-RVND e novas heurísticas são utilizadas, demonstrando um aumento na eficiência.

O problema que adiciona drones aos serviços tradicionais de entregas com veículos foi introduzido por Murray e Chu (2015). O problema se define como seguinte: uma frota de caminhões com drones para realizar entregas a um conjunto de clientes. O objetivo é minimizar o tempo máximo em que todas as entregas são realizadas para todos os clientes, com rotas de custo mínimo (WANG; SHEU, 2019). Esse problema tem sido classificado no contexto da entregas de última milha, bastante abordada na literatura como *last-mile*

delivery (DANG *et al.*, 2024). O trabalho de Murray e Chu (2015) propôs a formulação de um modelo matemático de programação linear e uma heurística para aplicação ao problema de roteamento de um caminhão de entregas com um drone, o *Flying Sidekick Traveling Salesman Problem* (FSTSP).

Problema similar ao de Murray e Chu (2015), uma variante do PCV chamado de *Traveling Salesman Problem with Drone* (TSP-D), foi proposto por Agatz *et al.* (2018), que utilizou formulação de programação inteira e a aplicação várias heurísticas baseadas em busca local e programação dinâmica na abordagem. Agatz *et al.* (2018) fazem uma comparação na tabela 1 entre caminhão e drone, que pode explicar por que a cooperação pode ser vantajosa:

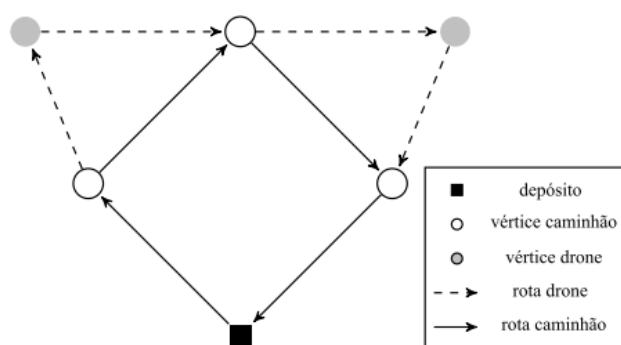
Tabela 1 – Comparação entre drone e caminhão

	velocidade	peso	capacidade	mobilidade
drone	alta	leve	um	longa
caminhão	baixa	pesado	vários	curta

Fonte: (AGATZ *et al.*, 2018)

Uma visualização simples de como é feita distribuição de clientes (vértices) em rotas para do problema para um caminhão e um drone é vista na figura 3:

Figura 3 – Exemplo de Roteamento de Caminhão e Drone



Fonte: (AGATZ *et al.*, 2018)

Wang *et al.* (2017) adicionou a generalização que associa a uma frota, um número de drones a cada veículo. O trabalho de Poikonen *et al.* (2017) revisa e estende o trabalho anterior (WANG *et al.*, 2017), adicionando novas abordagens dos casos estudados e uma nova variante do *Vehicle Routing Problem with Drones* (VRPD). Costa (2019) propôs

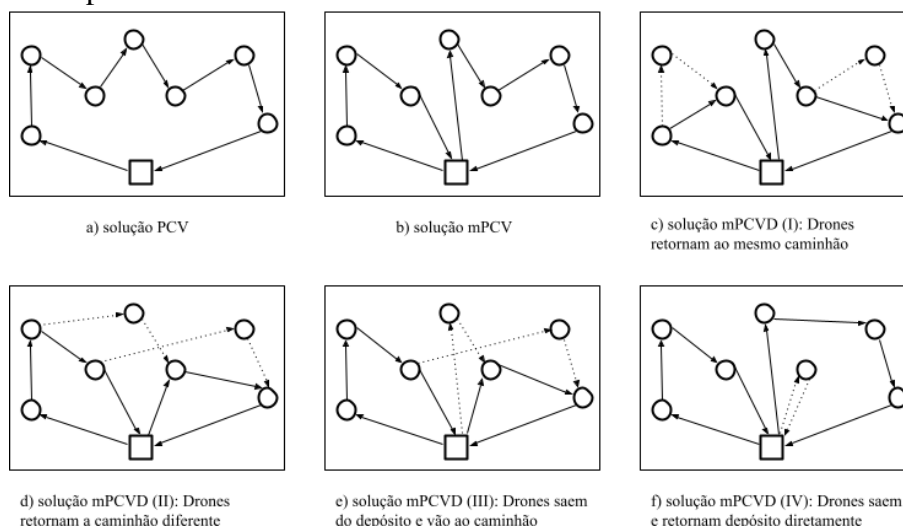
um modelo matemático de programação inteira mista e um algoritmo genético híbrido para solucionar o VRPD. Sacramento *et al.* (2019) utilizam a proposta da metaheurística *Adaptive Large Neighborhood Search*. Tamke e Buscher (2021) desenvolvem uma formulação de programação linear, além de solucionar o problema com o algoritmo *Branch-and-Cut*. Kuo *et al.* (2022) fazem uma abordagem com a definição de uma formulação matemática e apresentam uma solução com *Variable Neighborhood Search* (VNS) para o problema, considerando janelas de tempo.

O recente trabalho de Costa (2023) aplicou um Algoritmo Memético e o *Lin-Kernighan-Helsgaun* (LKH) para otimizar as rotas do PRV com caminhões e drones, considerando capacidades e demandas para o contexto de entregas fracionárias de mercadorias, o SDVRPD.

Para uma maior abordagem entre as aplicações e as variantes, as recentes revisões de literatura reúnem os trabalhos relevantes sobre o problema do VRPD (VILORIA *et al.*, 2021; DANG *et al.*, 2024; SONG *et al.*, 2025).

A figura 4 mostra os exemplos de algumas das generalizações e variantes do PCV e PRV sem e com os drones:

Figura 4 – Os problemas e variantes



Fonte: Adaptação de Kitjacharoenchai *et al.* (2019)

A tabela 2, a seguir, traz um quadro comparativos entre trabalhos relevantes na literatura recente que abordam o PRV nos contextos de Entregas Fracionadas e cooperação com Drones.

Tabela 2 – Trabalhos PRV com Entregas Fracionadas e/ou Drones

Trabalho	Abordagem	Entregas Fracionadas	Drones
Silva <i>et al.</i> (2015)	Híbrida	Sim	Não
Gschwind <i>et al.</i> (2019)	Exata	Sim	Não
Ji <i>et al.</i> (2021)	Metaheurística	Sim	Não
Wu e Wu (2022)	Híbrida	Sim	Não
Gouveia <i>et al.</i> (2023)	Exata	Sim	Não
Ren <i>et al.</i> (2023)	Metaheurística	Sim	Não
Lin <i>et al.</i> (2025)	Híbrida	Sim	Não
Murray e Chu (2015)	Heurística	Não	Sim
Agatz <i>et al.</i> (2018)	Híbrida	Não	Sim
Wang <i>et al.</i> (2017)	Híbrida	Não	Sim
Costa (2019)	Híbrida	Não	Sim
Wang e Sheu (2019)	Híbrida	Não	Sim
Nguyen <i>et al.</i> (2022)	Heurística	Não	Sim
Costa (2023)	Híbrida	Sim	Sim
Mahmoudi e Eshghi (2025)	Metaheurística	Sim	Sim
Este trabalho	Híbrida	Sim	Sim

Fonte: Autoria própria

2.5.1 *Split Delivery Vehicle Routing Problem with Drones*

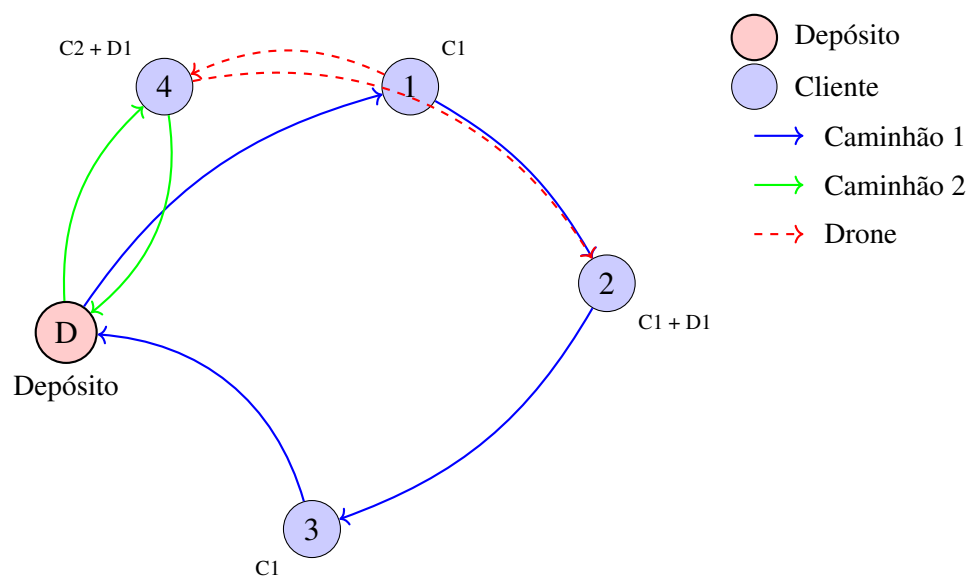
O *Split Delivery Vehicle Routing Problem with Drone* (SDVRPD) representa uma síntese inovadora entre o SDVRP e o VRPD. Esse problema surgiu da necessidade de combinar a flexibilidade da abordagem das entregas fracionárias (*split delivery*), onde múltiplos veículos podem atender um mesmo cliente, com as vantagens operacionais dos drones em sistemas de entrega modernos.

O surgimento do SDVRPD pode ser atribuído à convergência das possibilidades de benefícios dos dois problemas. Primeiro, o SDVRP foi desenvolvido como relaxação do VRP clássico, removendo a restrição de que cada cliente deve ser visitado exatamente uma vez, possibilitando uma redução na soma das rotas (BELENGUER *et al.*, 2000). Segundo, o interesse em drones para entrega comercial intensificou-se após o anúncio da *Amazon* em 2013, levando Murray e Chu (MURRAY; CHU, 2015) a formalizarem o *FSTSP* em 2015.

A importância do SDVRPD manifesta-se em múltiplas aplicações práticas. Em logística comercial, empresas como *Amazon*, DHL e UPS implementam sistemas híbridos caminhão-drone. Em operações de emergência, a combinação permite acesso a áreas remotas mantendo capacidade de carga. Durante a pandemia de COVID-19, a empresa *Soarizon* utilizou drones nas Terras Altas da Escócia, reduzindo tempos de entrega de seis horas para quinze minutos (MAHMOUDI; ESHGHI, 2025).

A Figura 5 ilustra um exemplo prático do SDVRPD, demonstrando como caminhões e drones colaboram para atender clientes através do *split delivery*.

Figura 5 – Exemplo do SDVRPD com *split delivery* no cliente 4.



Fonte: Autoria própria

3 FORMULAÇÕES MATEMÁTICAS

3.1 Formulação do Split Delivery Vehicle Routing Problem (SDVRP)

O Split Delivery Vehicle Routing Problem (SDVRP) pode ser formulado como um problema de programação linear inteira mista que estende o modelo clássico do CVRP para permitir entregas fracionadas.

3.1.1 Conjuntos e Parâmetros

Seja $G = (V, A)$ um grafo completo onde $V = \{0, 1, 2, \dots, n\}$ representa o conjunto de nós, com 0 denotando o depósito e $V' = \{1, 2, \dots, n\}$ representando o conjunto de clientes.

Os parâmetros do problema são definidos como:

- d_i : demanda do cliente $i \in V'$ (em unidades de carga)
- Q : capacidade uniforme de cada veículo (em unidades de carga)
- c_{ij} : custo de viagem do arco $(i, j) \in A$
- K : conjunto de veículos (caminhões) disponíveis

3.1.2 Variáveis de Decisão

As variáveis de decisão do modelo são:

- $x_{ijk} \in \{0, 1\}$: variável binária que assume valor 1 se o veículo k percorre o arco (i, j) , 0 caso contrário
- $y_{ik} \geq 0$: quantidade de demanda do cliente i atendida pelo veículo k
- $u_{ik} \geq 0$: variável auxiliar representando a carga no veículo k ao sair do cliente i

3.1.3 Formulação Matemática Completa do SDVRP

A formulação matemática do SDVRP é apresentada como segue:

$$\text{Minimizar } \sum_{k \in K} \sum_{(i,j) \in A} c_{ij} x_{ijk} \quad (3.1)$$

$$\text{Sujeito a: } \sum_{k \in K} y_{ik} = d_i, \quad \forall i \in V' \quad (3.2)$$

$$\sum_{j \in V} x_{ijk} = \sum_{j \in V} x_{jik}, \quad \forall i \in V, \forall k \in K \quad (3.3)$$

$$\sum_{j \in V'} x_{0jk} \leq 1, \quad \forall k \in K \quad (3.4)$$

$$\sum_{i \in V'} x_{i0k} = \sum_{j \in V'} x_{0jk}, \quad \forall k \in K \quad (3.5)$$

$$y_{ik} \leq d_i \sum_{j \in V} x_{ijk}, \quad \forall i \in V', \forall k \in K \quad (3.6)$$

$$\sum_{i \in V'} y_{ik} \leq Q \sum_{j \in V'} x_{0jk}, \quad \forall k \in K \quad (3.7)$$

$$u_{ik} - u_{jk} \geq y_{ik} - Q(1 - x_{ijk}), \quad \forall i, j \in V', i \neq j, \forall k \in K \quad (3.8)$$

$$y_{ik} \leq u_{ik} \leq Q \cdot \sum_{j \in V} x_{ijk}, \quad \forall i \in V', \forall k \in K \quad (3.9)$$

$$x_{ijk} \in \{0, 1\}, \quad \forall (i, j) \in A, \forall k \in K \quad (3.10)$$

$$y_{ik} \geq 0, \quad \forall i \in V', \forall k \in K \quad (3.11)$$

$$u_{ik} \geq 0, \quad \forall i \in V', \forall k \in K \quad (3.12)$$

3.1.4 Descrição Detalhada das Restrições do SDVRP

A função objetivo (3.1) minimiza o custo total de transporte. As restrições (3.2) garantem que a demanda de cada cliente seja completamente atendida. As restrições (3.3) asseguram a conservação de fluxo em cada nó. As restrições (3.4) e (3.5) controlam o uso dos veículos e garantem que retornem ao depósito. A restrição (3.6) vincula as variáveis de entrega às rotas dos veículos, enquanto (3.7) respeita a capacidade dos veículos. As restrições (3.8) e (3.9) são as desigualdades de Miller-Tucker-Zemlin (MTZ) para eliminação de sub-rotas. Finalmente, as restrições de (3.10) a (3.12) definem os domínios das variáveis.

3.2 Formulação do Split Delivery Vehicle Routing Problem with Drones (SDVRPD)

A formulação do SDVRPD estende o modelo SDVRP para incorporar drones como veículos auxiliares.

Simplificações e Justificativas

Para tornar o SDVRPD mais manejável, adotamos as seguintes simplificações: velocidades constantes para todos os veículos; autonomia ilimitada e recarga instantânea para drones; operações de lançamento e recuperação restritas aos clientes; uso do modelo *flying sidekick*, onde drones realizam entregas ponto a ponto; e ausência de custos fixos. O custo do drone é um fator proporcional ($\alpha < 1$) ao custo do caminhão, e suas capacidades são diferenciadas.

3.2.1 Conjuntos e Parâmetros Adicionais

O SDVRPD introduz:

- $D = \{1, 2, \dots, p\}$: conjunto de drones disponíveis
- Q_d : capacidade uniforme de cada drone (em unidades de carga)
- $\alpha \in (0, 1)$: fator de custo dos drones em relação aos caminhões

3.2.2 Variáveis de Decisão Adicionais

As variáveis adicionais para o SDVRPD são:

- $p_{ik} \in \{0, 1\}$: variável binária 1 se o caminhão k para e serve o cliente i , 0 caso contrário
- $z_{ijd} \in \{0, 1\}$: variável binária 1 se o drone d voa do nó i para o nó j , 0 caso contrário
- $w_{ikd} \geq 0$: demanda do cliente i atendida pelo drone d lançado do caminhão k
- $l_{ikd} \in \{0, 1\}$: variável binária 1 se o drone d é lançado do caminhão k no cliente i , 0 caso contrário
- $r_{jkd} \in \{0, 1\}$: variável binária 1 se o drone d é recuperado pelo caminhão k no cliente j , 0 caso contrário

3.2.3 Formulação Matemática Completa do SDVRPD

A formulação completa do SDVRPD é apresentada abaixo. Esta versão é reforçada com uma restrição de exclusão mútua para alinhar o modelo estritamente à lógica FSTSP.

$$\text{Minimizar } \sum_{k \in K} \sum_{(i,j) \in A} c_{ij} x_{ijk} + \sum_{d \in D} \sum_{(i,j) \in A} \alpha c_{ij} z_{ijd} \quad (3.13)$$

$$\text{Sujeito a: } \sum_{k \in K} y_{ik} + \sum_{d \in D} \sum_{k \in K} w_{ikd} = d_i, \quad \forall i \in V' \quad (3.14)$$

$$y_{ik} \leq Q \cdot p_{ik}, \quad \forall i \in V', \forall k \in K \quad (3.15)$$

$$\sum_{d \in D} w_{ikd} \leq d_i \cdot (1 - p_{ik}), \quad \forall i \in V', \forall k \in K \quad (3.16)$$

$$p_{ik} \leq \sum_{j \in V} x_{jik}, \quad \forall i \in V', \forall k \in K \quad (3.17)$$

$$\sum_{i \in V'} w_{ikd} \leq Q d, \quad \forall k \in K, \forall d \in D \quad (3.18)$$

$$\sum_{j \in V'} z_{ijd} = \sum_{k \in K} l_{ikd}, \quad \forall i \in V', \forall d \in D \quad (3.19)$$

$$\sum_{i \in V'} z_{ijd} = \sum_{k \in K} r_{jkd}, \quad \forall j \in V', \forall d \in D \quad (3.20)$$

$$\sum_{i \in V'} l_{ikd} = \sum_{j \in V'} r_{jkd}, \quad \forall k \in K, \forall d \in D \quad (3.21)$$

$$l_{ikd} \leq \sum_{j \in V} x_{jik}, \quad \forall i \in V', \forall k \in K, \forall d \in D \quad (3.22)$$

$$r_{jkd} \leq \sum_{i \in V} x_{ijk}, \quad \forall j \in V', \forall k \in K, \forall d \in D \quad (3.23)$$

$$\sum_{j \in V} x_{ijk} = \sum_{j \in V} x_{jik}, \quad \forall i \in V, \forall k \in K \quad (3.24)$$

$$\sum_{j \in V'} x_{0jk} \leq 1, \quad \forall k \in K \quad (3.25)$$

$$\sum_{i \in V'} x_{i0k} = \sum_{j \in V'} x_{0jk}, \quad \forall k \in K \quad (3.26)$$

$$\sum_{i \in V'} y_{ik} \leq Q \sum_{j \in V'} x_{0jk}, \quad \forall k \in K \quad (3.27)$$

$$u_{ik} - u_{jk} \geq y_{ik} - Q(1 - x_{ijk}), \quad \forall i, j \in V', i \neq j, \forall k \in K \quad (3.28)$$

$$y_{ik} \leq u_{ik} \leq Q \cdot \sum_{j \in V} x_{ijk}, \quad \forall i \in V', \forall k \in K \quad (3.29)$$

$$p_{ik} \in \{0, 1\}, \quad \forall i \in V', \forall k \in K \quad (3.30)$$

$$x_{ijk}, z_{ijd}, l_{ikd}, r_{jkd} \in \{0, 1\}, \quad \forall (i, j) \in A, \forall k \in K, \forall d \in D \quad (3.31)$$

$$y_{ik}, w_{ikd}, u_{ik} \geq 0 \quad (3.32)$$

3.2.4 Descrição Detalhada das Restrições do SDVRPD

A seguir, cada restrição do modelo SDVRPD é detalhada.

Função Objetivo (3.13) Minimiza a soma dos custos totais de viagem, combinando o custo das rotas dos caminhões com o custo das rotas dos drones, este último ponderado pelo fator α .

Demanda (3.14) Garante que a demanda total d_i de cada cliente i seja exatamente satisfeita pela soma das entregas feitas por todos os caminhões (y_{ik}) e por todos os drones (w_{ikd}).

Exclusão Mútua (Caminhão) (3.15) Permite que a entrega do caminhão k ao cliente i (y_{ik}) seja maior que zero somente se o caminhão k for designado para parar e servir aquele cliente ($p_{ik} = 1$).

Exclusão Mútua (Drone) (3.16) Permite que a entrega de um drone (lançado pelo caminhão k) ao cliente i (w_{ikd}) seja maior que zero somente se o caminhão k não for designado para parar naquele cliente ($p_{ik} = 0$). Juntamente com a restrição anterior, proíbe a redundância.

Vínculo de Visita (3.17) Conecta a nova variável de decisão p_{ik} à rota do caminhão. Um caminhão k só pode ser designado para servir o cliente i se ele de fato viajar para aquele cliente.

Capacidade do Drone (3.18) Assegura que a carga total transportada por um drone em uma missão, lançado pelo caminhão k , não exceda a capacidade máxima do drone, Q_d .

Lançamento do Drone (3.19) Garante que, se um drone d realiza uma missão partindo do cliente i , ele deve ter sido lançado por exatamente um caminhão k naquele cliente i .

Recuperação do Drone (3.20) Garante que, se um drone d finaliza uma missão chegando ao cliente j , ele deve ser recuperado por exatamente um caminhão k naquele cliente j .

Retorno ao Mesmo Caminhão (3.21) Força que cada drone lançado por um caminhão k seja recuperado pelo mesmo caminhão k , garantindo que o número de lançamentos seja igual ao número de recuperações para cada par caminhão-drone.

Sincronização de Lançamento (3.22) Um drone só pode ser lançado pelo caminhão k no cliente i se o caminhão k efetivamente visitar o cliente i .

Sincronização de Recuperação (3.23) Um drone só pode ser recuperado pelo caminhão k no cliente j se o caminhão k efetivamente visitar o cliente j .

Conservação de Fluxo (Caminhão) (3.24) Restrição padrão que garante que, para cada nó, o número de arcos de caminhão que entram é igual ao número de arcos que saem, mantendo a continuidade da rota.

Saída do Depósito (Caminhão) (3.25) Garante que cada caminhão possa sair do depósito no máximo uma vez.

Retorno ao Depósito (Caminhão) (3.26) Assegura que todo caminhão que sai do depósito deve retornar a ele.

Capacidade do Caminhão (3.27) Garante que a soma de todas as entregas y_{ik} feitas por um caminhão k em sua rota não exceda sua capacidade máxima Q .

Eliminação de Sub-rotas (MTZ - Fluxo) (3.28) Modela a diminuição da carga no caminhão ao longo da rota. Impede a formação de rotas desconectadas do depósito.

Eliminação de Sub-rotas (MTZ - Limites) (3.29) Define os limites superior e inferior para a variável de fluxo de carga u_{ik} , garantindo que ela seja consistente com a capacidade do veículo e a entrega realizada.

Domínio das Variáveis (3.30)-(3.32) Definem os tipos de cada variável (binária ou não negativa).

3.3 Análise Comparativa das Restrições: SDVRP vs. SDVRPD

A tabela a seguir apresenta uma comparação direta entre as formulações matemáticas do *Split Delivery Vehicle Routing Problem* (SDVRP) e sua extensão com drones, o *Split Delivery Vehicle Routing Problem with Drones* (SDVRPD).

Tabela 3 – Comparação das formulações SDVRP e SDVRPD.

Categoria	SDVRP	SDVRPD	Ajuste/Adição
Função Objetivo	Minimiza custo das rotas dos caminhões.	Minimiza custo de caminhões e drones.	Alterada: Inclui o custo ponderado (α) das operações dos drones.
Demanda	Atendida apenas por caminhões.	Atendida por caminhões ou drones.	Alterada: Expande a entrega para incluir drones (w_{ikd}), aumentando a flexibilidade.
Exclusão Mútua	Não aplicável.	Cliente servido por caminhão ou drone, não ambos.	Nova: Evita redundância na entrega. Se $p_{ik} = 1$, a entrega por drone é proibida.
Visita/Parada	Entrega vinculada à visita do caminhão.	Parada do caminhão (p_{ik}) vinculada à rota.	Nova: Formaliza a decisão de parada do caminhão como pré-requisito para sua entrega.
Capacidade (Caminhão)	Carga da rota não excede a capacidade Q .	Carga da rota não excede a capacidade Q .	Inalterada: A lógica de capacidade do caminhão permanece a mesma.
Capacidade (Drone)	Não aplicável.	Carga da missão não excede a capacidade Q_d .	Nova: Adiciona um limite de carga específico para as missões dos drones.
Voo/Recuperação	Não aplicável.	Controla o início e o fim das missões dos drones.	Novas: Garantem pontos de lançamento e recuperação válidos para cada voo.
Sincronização	Não aplicável.	Drone lançado e recuperado pelo mesmo caminhão.	Novas: Sincroniza o voo do drone com a rota do caminhão-mãe.

3.4 Exemplo Numérico e Análise Comparativa

Para demonstrar a aplicação prática das formulações matemáticas desenvolvidas e evidenciar as vantagens potenciais do SDVRPD em relação ao SDVRP tradicional, apresenta-se um exemplo numérico detalhado. Assume-se uma capacidade de **50 kg** para cada caminhão e $\alpha = 1$. Os custos das rotas são calculados usando a distância Euclidiana.

3.4.1 Configuração do Problem

3.4.2 Solução SDVRP (Apenas Caminhões)

Nesta configuração, atender o Cliente 5 força um dos caminhões a um desvio longo e ineficiente, aumentando o custo total da solução.

Rotas e Custos

- **Caminhão 1 (45 kg):** Rota D → C1 → C5 → D. Custo = 163,04.
- **Caminhão 2 (45 kg):** Rota D → C2 → C4 → D. Custo = 141,42.
- **Caminhão 3 (30 kg):** Rota D → C3 → D. Custo = 87,75.

O custo total real da solução SDVRP é de **392,21** unidades.

3.4.3 Solução SDVRPD (Caminhões + Drones)

A solução híbrida delega a entrega inconveniente do Cliente 5 a um drone, permitindo que os caminhões executem rotas mais curtas e densas.

Rotas e Custos

- **Caminhões:** Rotas otimizadas sem o desvio para C5. Custo total = **289,71**.
- **Drones:** Missão C1 → C5 → C2 e entrega fracionada. Custo total = **95,97**.

O custo total real da solução SDVRPD, assumindo $\alpha = 1$, é de **385,68** unidades.

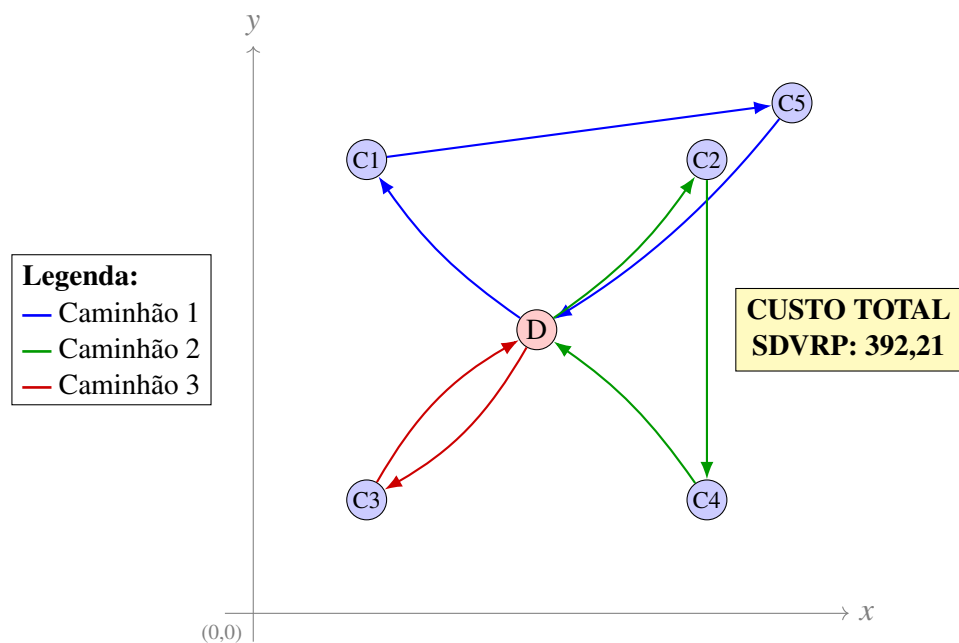
3.4.4 Análise Comparativa

A comparação dos custos reais para esta configuração estratégica demonstra a vantagem da abordagem híbrida.

Neste cenário projetado, a solução SDVRPD alcançou uma **redução de custo de 1,66%** em relação à solução tradicional, mesmo no cenário de custo menos favorável para o drone ($\alpha = 1$). Este resultado valida o potencial do modelo: ao delegar entregas em

Tabela 4 – Localização e Demanda dos Nós

Nó	Coordenadas	Demanda (kg)
Depósito (0)	(50, 50)	–
Cliente 1	(20, 80)	30
Cliente 2	(80, 80)	25
Cliente 3	(20, 20)	30
Cliente 4	(80, 20)	20
Cliente 5	(95, 90)	15
Demanda Total dos Clientes		120 kg

Figura 6 – Solução SDVRP com Cliente 5 como *distante*.

Fonte: Autoria própria

Tabela 5 – Comparação Quantitativa com Custos Reais

Métrica	SDVRP	SDVRPD
Custo Total	392,21	385,68
Número de Viagens de Caminhão	3	2
Utilização Média da Capacidade	80%	100%
Economia Absoluta	–	6,53

locais geograficamente desfavoráveis para drones, o sistema consegue otimizar as rotas dos caminhões, reduzir o número de viagens e a distância total percorrida, resultando em uma operação logística mais eficiente.

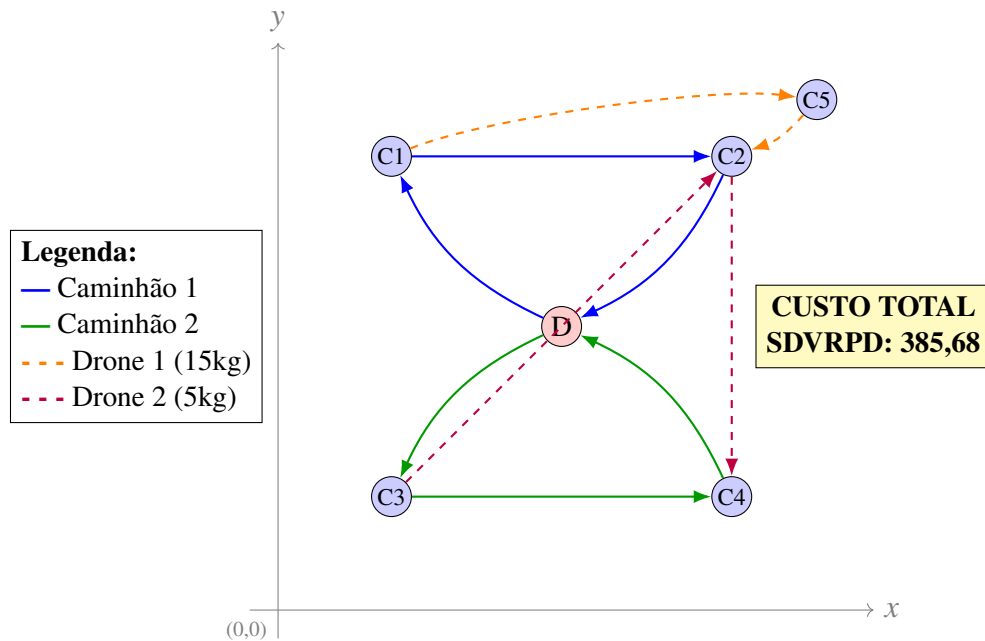
3.4.5 Análise de Sensibilidade do Fator de Custo α

Para entender como a eficiência operacional do drone impacta a viabilidade da solução híbrida, realizou-se uma análise de sensibilidade variando o fator de custo α . A tabela abaixo apresenta os resultados para uma escala de valores de α , comparando sempre com o custo da solução SDVRP (392,21).

3.4.6 Conclusão Final da Análise Comparativa

A análise de sensibilidade do fator de custo α mostra que, quanto mais eficiente for o drone em relação ao caminhão, maior a economia obtida pelo SDVRPD, validando o investimento em tecnologias mais avançadas. O exemplo numérico com custos reais confirma esse potencial, mostrando que o modelo gera economias significativas ao transformar a logística de um problema de “menor caminho” em uma decisão estratégica sobre qual veículo deve atender cada entrega, de forma a minimizar os custos de transporte na entrega das mercadorias.

Figura 7 – Solução SDVRPD onde o drone atende o cliente distante.



Fonte: Autoria própria

Tabela 6 – Resumo Comparativo em Escala para Diferentes Valores de α

Fator de Custo α	Custo Total SDVRPD	Economia	(%)
1,0 (Custo Igual)	385,68	6,53	1,7%
0,4 (Drone 60% mais eficiente)	328,10	64,11	16,3%
0,3 (Drone 70% mais eficiente)	318,50	73,71	18,8%
0,2 (Drone 80% mais eficiente)	308,90	83,31	21,2%

4 HEURÍSTICAS E METAHEURÍSTICAS

Serão apresentadas neste capítulo as técnicas heurísticas e metaheurísticas, os algoritmos utilizados para aplicação ao problema do trabalho.

4.1 Algoritmo Genético

O Algoritmo Genético foi proposto por Holland (1975). Assim como a classe de algoritmos baseados na natureza, ele é baseado no processo natural de evolução, usando o conceito de sobrevivência do mais apto para encontrar as melhores soluções para os indivíduos. No Algoritmo Genético, soluções para problemas de otimização combinatória são geradas em populações como indivíduos. Esses indivíduos, como possíveis soluções, são interpretados pelo campo dos algoritmos evolucionários como de cromossomos, e representados por vetores, com posições como genes e os valores dessas posições como os alelos.

O PCV, para ser solucionado pelo Algoritmo Genético, recebe a representação de uma possível solução como: um cromossomo como rota ou circuito hamiltoniano; os genes como a sequência ou ordem; e os alelos como as cidades ou localizações. Em um caso PCV, com um número de n cidades, cada uma representada por um índice e com todos os custos de distância entre cada par dessas cidades, o Algoritmo Genético utiliza como população o grupo de possíveis arranjos na ordem dessas cidades.

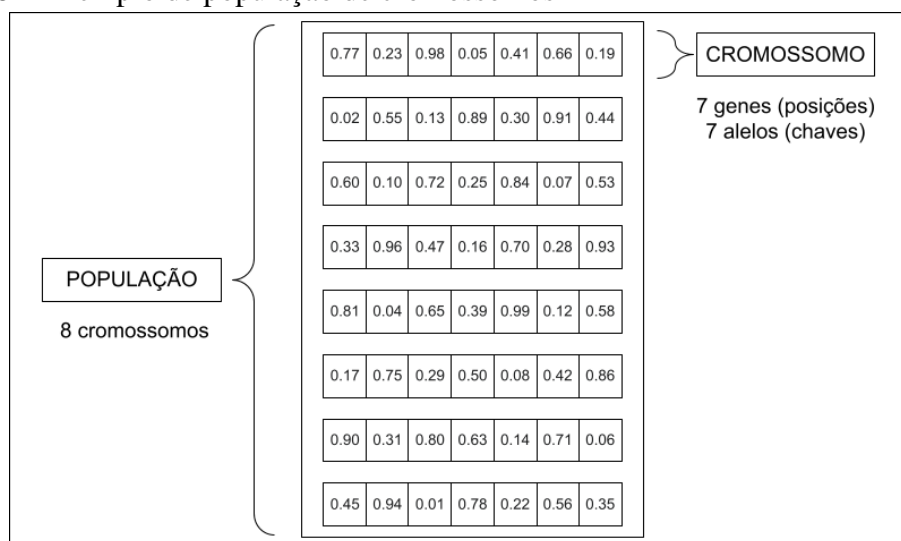
Tendo um conjunto inicial de cromossomos (geração inicial), os passos para gerar novos cromossomos se dão por meio de novas gerações. O algoritmo realiza os cruzamentos entre cromossomos para gerar novos cromossomos, novas combinações de rotas. Nesses passos, os termos '*crossover*', '*offspring*' e '*mutantes*' são referentes ao processo evolucionário do Algoritmo Genético.

4.2 *Biased Random-Key Genetic Algorithm*

O Algoritmo Genético com Chaves Aleatórias, tradução literal de *Random-Key Genetic Algorithm* (RKGA), é uma variante do Algoritmo Genético, e foi introduzido por Bean (1994). No RKGA, as populações das gerações são representadas por cromossomos de valores aleatórios, as chaves. O algoritmo gera, como passo inicial, um conjunto de cromossomos como vetores de números reais aleatórios, no intervalo $[0, 1]$. Um exemplo

visual de uma população gerada é mostrado na figura 8.

Figura 8 – Exemplo de população de cromossomos



Fonte: Autoria própria

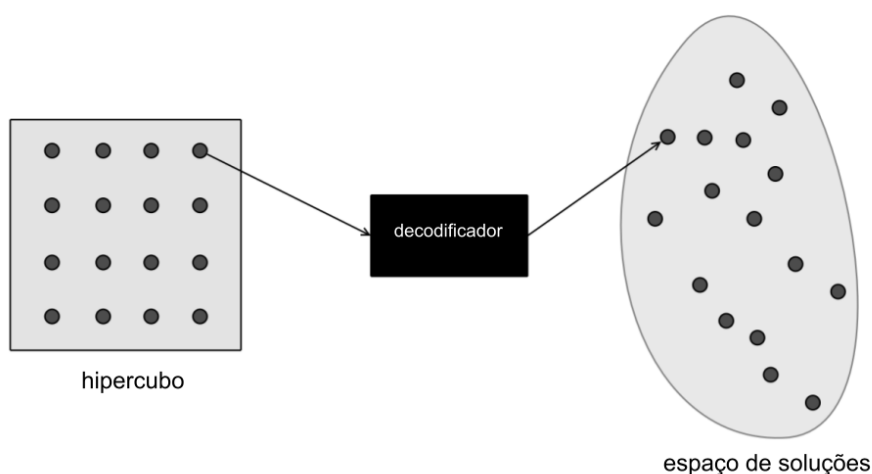
Após esse passo inicial, um algoritmo determinístico inicia a decodificação dos vetores de chaves aleatórias, realizando associação das posições de um cromossomo, os genes, com os índices equivalentes dos vértices que representam o conjunto de um problema. Em seguida, o decodificador classifica as chaves aleatórias do cromossomo em ordem crescente, o que recombina suas posições. Com essa recombinação, os índices associados às chaves aleatórias assumem as mesmas posições dessas chaves em um vetor. Quando todos os cromossomos de chaves aleatórias são decodificados, o custo de uma solução, que é o valor *fitness* de cada cromossomo, é calculado e o algoritmo classifica os cromossomos, do melhor para o pior *fitness*.

No RKGA, cada população (geração) é formada por p cromossomos de chaves aleatórias. Quando o valor de *fitness* desses cromossomos é calculado, o algoritmo classifica os cromossomos da população em dois grupos: um grupo elite, que são os indivíduos com melhor *fitness* e um grupo não-elite, que é o restante dos indivíduos com *fitness* menores. O valor de *fitness* de cada cromossomo é calculado fazendo as associações aos índices da instância de um problema (figura 9).

Uma vez concluída a primeira população, com grupos de elite e não-elite, está pronta para gerar a próxima população. A primeira população é a população k e será usada para gerar a população $k+1$. A geração da população $k+1$ se dará em três processos:

No primeiro processo o grupo de elite da população k é copiado para a próxima

Figura 9 – Dados da Instância: Hypercube



Fonte: (GONÇALVES; RESENDE, 2011)

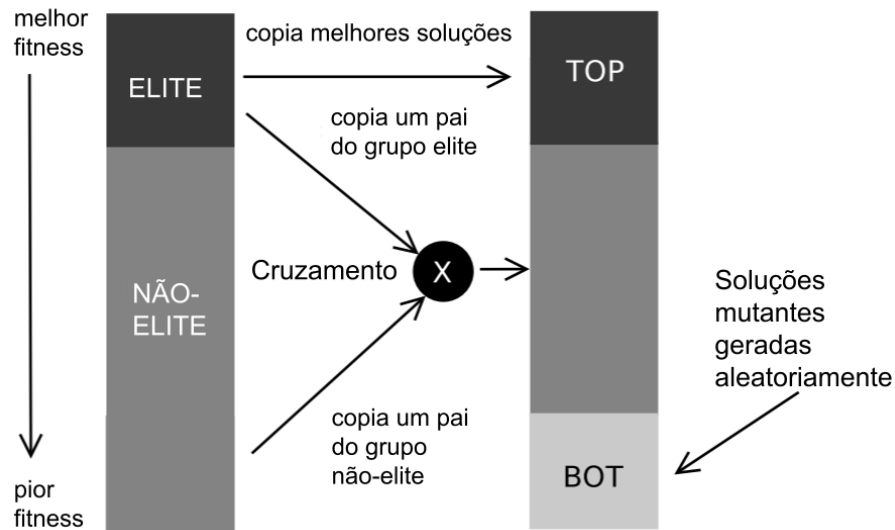
população $k+1$, representando a característica elitista; no segundo processo, cromossomos mutantes são adicionados na população $k+1$ através da geração de novos vetores de chaves aleatórias, da mesma forma que na população inicial; o último processo é o cruzamento, no qual cada novo cromossomo denominado *offspring* é gerado por dois pais selecionados aleatoriamente da população k . Quando os três processos são concluídos, o algoritmo decodifica a nova população e a aptidão é calculada. Em seguida, os cromossomos são agrupados em elite e não elite.

O Algoritmo Genético com Chaves Aleatórias Viciadas, do inglês *Biased Random-Key Genetic Algorithm* (BRKGA), é semelhante ao RKGA em quase todos os processos, diferindo na forma como os cromossomos pais são selecionados para o *crossover*. Enquanto no RKGA os pais são selecionados aleatoriamente da população inteira, no BRKGA, os cromossomos são selecionados de forma que um cromossomo seja do grupo de elite e outro seja do grupo não-elite (figura 10).

O pseudocódigo do algoritmo BRKGA, contendo seus principais passos é apresentado no algoritmo 1.

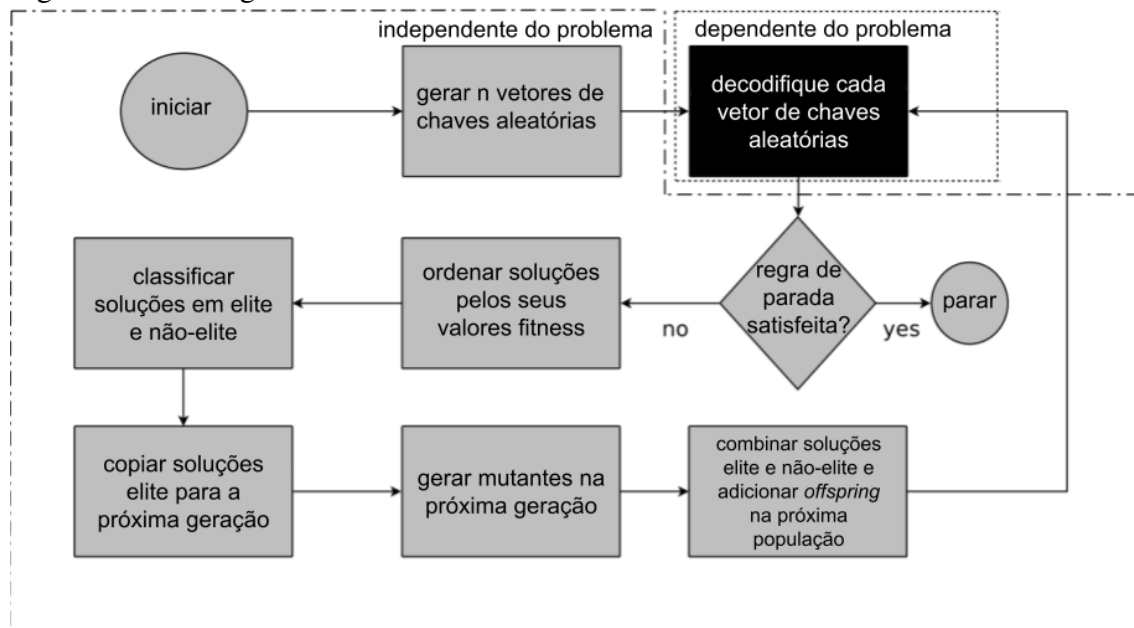
A seguir, o fluxograma que mostra todo os procedimentos do BRKGA, desde o início, com a geração de vetores de chaves aleatórias, até a classificação da próxima população para gerar novas, é apresentado na figura 11.

Figura 10 – Dinâmica entre Populações no BRKGA



Fonte: (GONÇALVES; RESENDE, 2011)

Figura 11 – Fluxograma do BRKGA



Fonte: (GONÇALVES; RESENDE, 2011)

4.3 Lin-Kernighan-Helsgaun

Dentre as várias heurísticas utilizadas para solucionar problemas de roteamento como o PCV, uma que tem se mostrado bastante efetiva nos testes com grandes instâncias é a LKH (HELGAUN, 2000). Sendo uma versão modificada do algoritmo *Lin-Kernighan* (LK), LKH representa uma extensão das técnicas originais, que são da classe de métodos de busca local. No algoritmo original LK, proposto por Lin e Kernighan (1973), a busca

Fonte: (RESENDE, 2013)

Entrada : $|P|, |P_e|, |P_m|, n, \rho_a$
Saída : Solução ótima χ^*
 BRKGA($|P|, |P_e|, |P_m|, n, \rho_a$);
Data: Inicialize o valor da melhor solução encontrada: $f^* \leftarrow \infty$;
while critério de parada não for satisfeito **do**
 Gere a população P com vetores de n chaves aleatórias;
 while critério de reinicialização não for satisfeito **do**
 Avalie o custo de cada solução nova em P ;
 Particione P em dois conjuntos: P_e e $\overline{P_e}$;
 Inicialize a população da próxima geração: $P^+ \leftarrow P_e$;
 Gere o conjunto de mutantes P_m , cada mutante com n chaves aleatórias;
 Adicione P_m à população da próxima geração: $P^+ \leftarrow P^+ \cup P_m$;
 for $i \leftarrow 1$ **to** $|P| - |P_e| - |P_m|$ **do**
 Escolha o pai a aleatoriamente de P_e ;
 Escolha o pai b aleatoriamente de $\overline{P_e}$;
 for $j \leftarrow 1$ **to** n **do**
 Jogue uma moeda viciada com probabilidade $\rho_a > 0.5$ de dar cara;
 if jogada der cara **then**
 $c[j] \leftarrow a[j]$;
 end
 else
 $c[j] \leftarrow b[j]$;
 end
 end
 Adicione o filho c à população da próxima geração: $P^+ \leftarrow P^+ \cup \{c\}$;
 end
 Atualize a população: $P \leftarrow P^+$;
 Ache a melhor solução χ^+ em P : $\chi^+ \leftarrow \arg \min \{f(\chi) | \chi \in P\}$;
 if $f(\chi^+) < f^*$ **then**
 $\chi^* \leftarrow \chi^+$;
 $f^* \leftarrow f(\chi^+)$;
 end
 end
end
return retorna χ^* ;

local se baseia nas técnicas *2-opt* e *3-opt*, que são utilizadas para realizar as trocas de arestas em uma rota (solução), o que gera uma combinação diferente na sequência de vértices, uma nova solução.

O LK mostrou ser bastante efetivo na otimização de problemas com dimensão de até 50 cidades (vértices), com probabilidade muito próxima a 100% de busca de solução ótima em teste único. Porém, a probabilidade cai cerca de 20% e 30% em problemas com 100 cidades, tal probabilidade aumentando quando são feitos alguns testes a mais

(HELGAUN, 2000). Isso mostra um certo limite na aplicação desse algoritmo para instâncias maiores (com mais de 100 cidades), e essa limitação vem de algumas restrições nos critérios de aceitação dessa heurística na escolha das arestas de troca. Como explicado em Helgaun (2000), os critérios de troca sequencial, viabilidade, ganho positivo, dentre outros, podem restringir a busca em soluções ótimas locais em determinadas condições.

O LKH veio com a proposta de revisar tais critérios do LK original e modificar as técnicas de trocas para possibilitar uma flexibilidade na busca de soluções em grandes instâncias. Uma das modificações que implicam na melhora da busca é remover a restrição de escolha de incluir até 5 vizinhos de uma dada cidade. Isso é feito introduzindo o conceito de *1-tree*, uma árvore geradora mínima que permite uma maior abrangência nos esquemas de candidatos. A modificação mais importante do LK feita pelo LKH é fazer os movimentos de troca base serem *5-opt* ou mais. Essa modificação generaliza a busca e permite melhores soluções globais.

Essas e as demais mudanças se mostraram efetivas e o algoritmo LKH conseguiu apontar tamanha efetividade na otimização de rotas em uma variedade de instâncias na literatura, se tornando o método "estado-da-arte" para o PCV e suas variações.

Na mesma pesquisa de sua modificação (HELGAUN, 2000), o autor do LKH inovou e desenvolveu um *solver* (*framework / software*) de código aberto que implementa o LKH para poder ser utilizado ou integrado para testes computacionais e oferece uma documentação que define os modelos de dados de entrada e saída e as possíveis opções de configurações para adaptação nas soluções de instâncias. Inicialmente direcionado para o PCV, o *solver* evoluiu e ganhou algumas atualizações e versões ao longo dos anos. Na sua versão mais recente, o *LKH-3*, o *solver* ganhou adaptação para os problemas e PRV (HELGAUN, 2017).

4.4 ILS-RVND

Esta técnica híbrida é uma combinação baseada na metaheurística ILS e na RVND, que adapta a *Variable Neighborhood Descent* com *Random Neighborhood Ordering* (RVND). A é proposta pelo trabalho de Penna *et al.* (2013) e se baseia, de modo geral, em busca local com a aplicação da ILS como um *framework*.

A RVND é implementada dentro do *framework* para a melhoria da qualidade da solução. Algumas regras são utilizadas durante a execução de cada iteração do *framework*,

como o critério de aceitação *Hill Climbing* para determinar se uma solução foi aceita. Nessa regra também são aceitos movimentos que não apresentam melhora na solução local, desde que, na solução completa final, o valor objetivo seja menor que o valor da melhor solução mais recente de iterações anteriores.

Durante os demais passos do *framework* são utilizadas várias técnicas, que se classificam como estruturas de vizinhança intra-rota, estruturas de vizinhança inter-rota e mecanismos de perturbação. Tais técnicas aplicam trocas de clientes nas sequências das sub-rotas, entre sub-rotas e no conjunto completo da solução, todas as sub-rota juntas.

O recente trabalho, *AlkaidSD*, realizado por Lin *et al.* (2025), descreve o desenvolvimento de um *solver open-source* (código aberto) para o desafio *VRP with Split Deliveries* do *Dimacs*, disponibilizado em: <http://dimacs.rutgers.edu/programs/challenge/vrp/vrpsd/>. O *solver* se destacou entre os demais candidatos, tendo os melhores resultados entre as instâncias utilizadas no desafio. Os autores descrevem que o algoritmo principal do *solver* é uma metaheurística *multistart* baseada na ILS e fazem uma implementação híbrida da heurística com o RVND. Os autores demonstram que se baseiam, fortemente, nos passos de integração das heurísticas descritos pelo trabalho de Penna *et al.* (2013) para a construção de solução inicial e os mecanismos de busca local e perturbação de sequências de vértices e trocas entre sub-rotas.

5 METODOLOGIA

Neste capítulo serão descritos os procedimentos metodológicos para a aplicação das heurísticas e metaheurísticas aos problemas de roteamento. Cada problema descrito nos objetivos terá sua solução com implementação das heurísticas e metaheurísticas em módulos, tais sendo de desenvolvimento próprio e com a integração de módulos de *frameworks* de terceiros. Ao final do projeto, esses módulos estarão reunidos em um *framework* que irá solucionar o problema do objetivo geral deste trabalho.

Os módulos com os algoritmos das heurísticas e metaheurísticas, com suas adaptações para a resolução dos problemas, serão implementados na linguagem C++ e, como descrito, serão integrados, recebendo, processando, e enviando os dados das soluções, havendo um fluxo sequencial. Ao final, a solução gerada, seu custo e tempo, do SDVRP com drones (SDVRPD) será o resultado do *framework* desenvolvido neste trabalho.

5.1 Instâncias

As instâncias utilizadas nos testes computacionais deste trabalho são as 11 instâncias propostas por Christofides e Eilon (1969). Tais instâncias são referentes ao *Capacited Vehicle Routing Problem* (CVRP) e têm uma vasta abordagem na literatura em trabalhos no contexto do PRV. As instâncias são disponibilizadas em uma variedade de repositórios. Para este trabalho, as instâncias foram adquiridas em dois formatos: o formato do repositório da biblioteca *TSPLIB 95* (REINHELT, 2014) e o formato utilizado pelo trabalho do *solver* (LIN *et al.*, 2025), disponibilizado pelo desafio SDVRP do *12th DIMACS Implementation Challenge*, disponível em: <http://dimacs.rutgers.edu/programs/challenge/vrp/vrpsd/>.

Cada instância CVRP representa um arquivo que contém os dados necessários para as definições para um caso do problema. Na implementação, cada arquivo é lido e seus dados são coletados de cada linha e armazenados em constantes, variáveis, vetores e matrizes. Nesses arquivos, os dados a serem utilizados na implementação estão nas linhas que representam *< tipodedado >: < valor >* e nas sessões, como o conjunto de vértices e seus formatos, as demandas e a indicação de qual vértice representa o depósito.

Na implementação é utilizada a representação dos vértices como uma matriz de distâncias. No processo de leitura de cada arquivo, um algoritmo verifica a sessão do conjunto de vértices, lê suas coordenadas euclidianas (bidimensional, no caso) e utiliza a

fórmula de distância entre pontos para construir a matriz, que tem a dimensão $n \times n$, com n sendo o número de vértices da instância. Cada linha da matriz representa um vértice (linha 0 à $n - 1$), e as posições em cada linha, as colunas, contêm os valores de distâncias entre esse vértice e cada um dos demais vértices (coluna 0 à $n - 1$). Essa representação matricial das distâncias permite que os valores sejam calculados uma única vez e acessados diretamente a cada vez que for necessário utilizar as distâncias.

5.2 Representação de solução

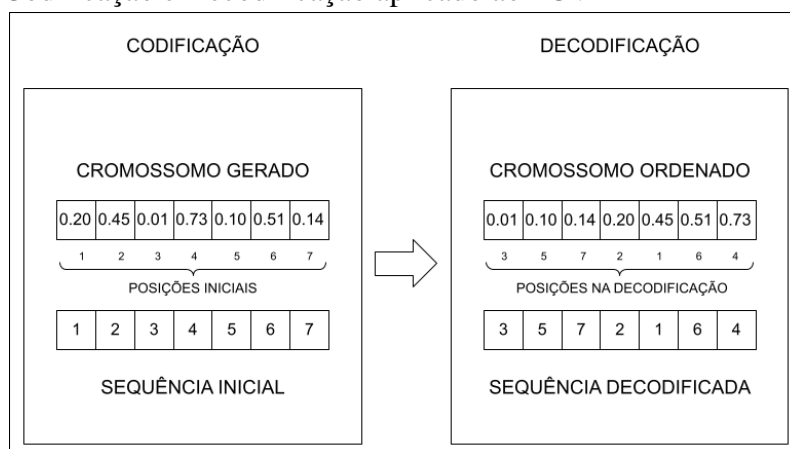
Durante todo o projeto computacional deste trabalho, a escolha de utilização de formatos de representação de solução são de crucial importância, pois os algoritmos que compõem o projeto recebem uma solução, fazem o processo e retornam uma nova solução. Dessa forma, os algoritmos devem estar adaptados para trabalharem com um formato específico. Para este trabalho a representação de uma solução para o CVRP é feita utilizando os seguintes formatos: um vetor do tipo inteiro com valores referentes aos clientes de uma instância, em que suas posições representam uma sequência (sub-rota); uma matriz com n vetores referentes às rotas e um vetor que representa a grande rota, com as sequências dos valores referentes aos clientes representando as sub-rotas, essas sendo intercaladas pelo valor do depósito. Durante todo o projeto uma função para calcular distâncias é utilizada, recebendo como entrada um vetor com as sequências de clientes e retornando o valor da distância total desse vetor, que representa uma rota. O algoritmo da função que realiza o cálculo recebe cada sequência de dois valores do vetor e os utiliza como índices de linha e coluna, na busca pela posição entre os vértices na matriz de distâncias.

5.3 BRKGA aplicado ao Problema do Caixeiro Viajante

Inicialmente serão utilizadas instâncias simples para solucionar o PCV. As instâncias serão representadas por matrizes de distâncias nas quais as localizações serão indicadas pelos índices das linhas e as distâncias para as localizações adjacentes de cada linha serão indicadas pelos índices das colunas. As localizações serão as representações de dois tipos para o problema: o depósito e os clientes. Como exemplo, para uma instância com 7 vértices, uma possível solução gerada através da codificação e decodificação do

BRKGA pode ser vista na figura 12.

Figura 12 – Codificação e Decodificação aplicado ao PCV



Fonte: Autoria própria

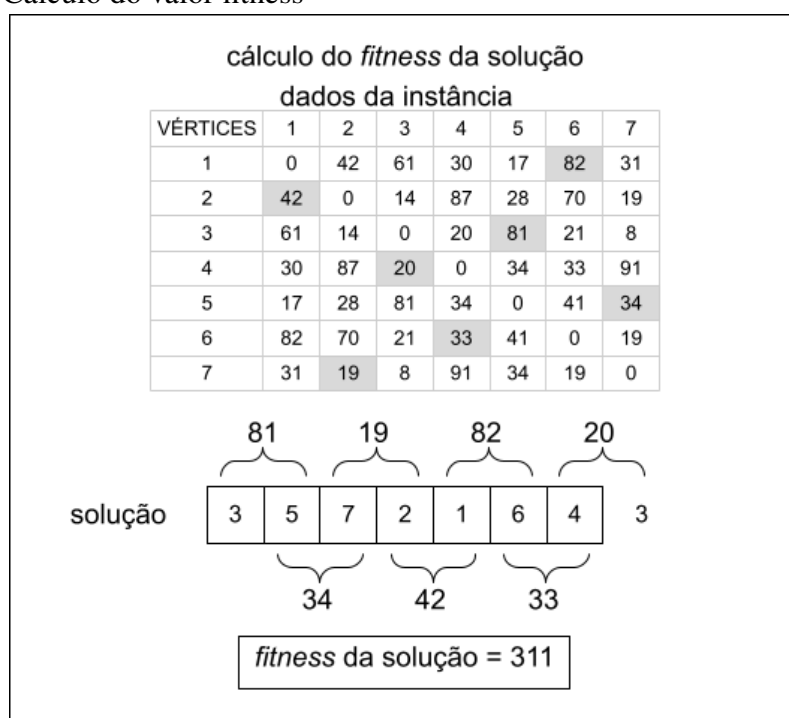
O metaheurística *Biased Random-Key Genetic Algorithm* (BRKGA) iniciará a solução do PCV, primeiramente, gerando a população inicial de p vetores (cromossomos) de chaves aleatórias. Então, o algoritmo associará os índices das posições de cada vetor de chaves aleatórias aos índices das respectivas localizações da instância, os índices do depósito e dos clientes. O algoritmo realizará a decodificação aplicando uma troca de posições para que os valores sejam ordenados de forma crescente. Os índices equivalentes das localizações também são trocados de posições e ordenados de acordo com a ordem das respectivas chaves. Ao final da decodificação da população inicial, os p cromossomos, com as chaves ordenadas, são associados aos índices e indicarão p possíveis soluções de rotas do PCV.

Cada vetor de chaves aleatórias, indicando a sequência de vértices de uma rota, será uma possível candidata a melhor solução. Na aplicação ao problema, é considerado que o início de cada rota seja, referente ao conjunto de um grafo, o vértice que representa o depósito. Nesse caso, o valor da primeira posição para todas as possíveis soluções. Os demais valores representam os clientes nas sequências das posições.

O próximo passo do algoritmo é calcular o custo (ou distância total) de cada rota (ciclo hamiltoniano) com base no vetor de chaves que define a sequência, buscando os valores correspondentes entre cada par de vértices da instância. Quando todas distâncias são somadas o valor de custo (*fitness*) total de cada rota é definido para cada vetor de chaves aleatórias da população. A partir desse momento o algoritmo ordena todos os vetores da população com relação às melhores rotas primeiro. Quanto menor custo, ou melhor

fitness, melhor será a classificação de um cromossomo. O vetor de chaves aleatórias com rota de menor custo associado será o primeiro da população e o vetor com pior custo, o último. Um exemplo de cálculo do *fitness* do PCV para 7 vértices feito com a sua matriz de distâncias é mostrado na figura 13.

Figura 13 – Cálculo do valor *fitness*



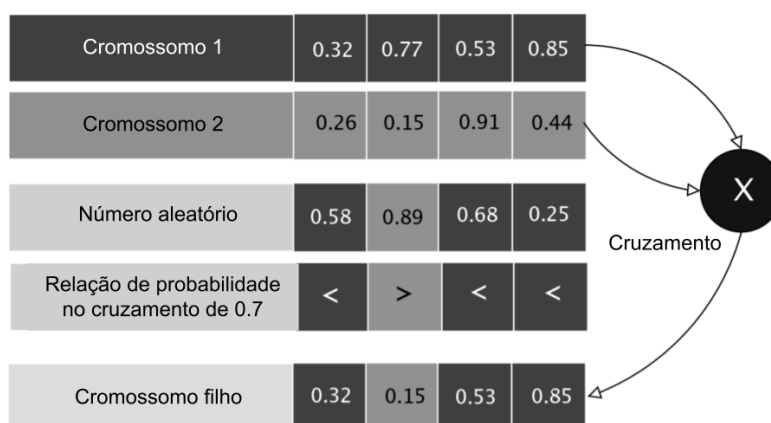
Fonte: Autoria própria

Quando a população estiver ordenada, com a classificação de *fitness*, o algoritmo BRKGA irá dividir a população em dois grupos: grupo elite e não-elite. Com isso, a população inicial está pronta.

Para a construção da próxima população $k + 1$, e das demais, o grupo elite da população inicial (ou atual k) será copiado para a população $k + 1$. Com o grupo elite copiado, os demais vetores (cromossomos não mutantes) da população serão gerados pelo cruzamento parametrizado entre um cromossomo elite e um cromossomo não elite da população k . A figura 14 mostra um exemplo de cromossomo filho gerado do cruzamento de um pai elite e um pai não elite, tais cromossomos tendo 4 genes.

Então, com todos os cruzamentos realizados, os demais cromossomos da nova população são gerados da mesma maneira que a população inicial, pm cromossomos com números aleatórios. Quando a população $k + 1$ estiver pronta, ela passa a ser a população atual k e, o algoritmo retorna ao procedimento de classificação dos vetores associados às

Figura 14 – Cruzamento Parametrizado Uniforme



Fonte: (GONÇALVES; RESENDE, 2011)

rotas. Para a nova população atual (k), divide os vetores de chaves em elite e não elite. Isso pois, depois do cruzamento anterior os vetores elite podem não continuar com os melhores custos. O algoritmo repetirá os procedimentos, gerando novas populações de acordo com os critérios informados até atingir uma quantidade específica para encontrar a melhor rota para o PCV.

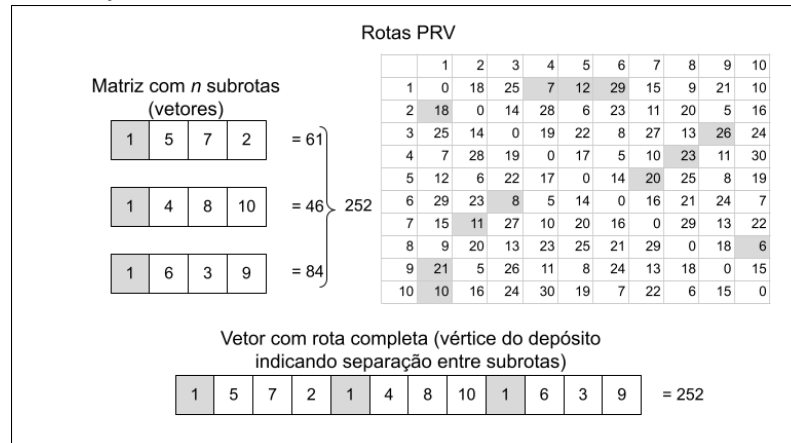
5.4 Aplicação ao Problema de Roteamento de Veículos

Sendo similar ao PMCV, o PRV varia na adição de restrições de quantidade de veículos disponíveis, suas capacidades e as demandas dos clientes. Como descrito anteriormente, o PRV se define como: uma frota de veículos e suas capacidades, um conjunto de clientes e suas demandas associadas. A solução do problema consiste em encontrar as rotas de menor custo para cada veículo. Mas, como as características de capacidades e demandas são levadas em consideração, o problema também busca otimizar a utilização dos veículos para a possível redução de custos de operação. Então, as demandas dos clientes, nos dados das instâncias, são fator determinante para verificar a disponibilização de um veículo de acordo com sua capacidade. Dessa forma, com a somatória de todas as demandas dos clientes será possível definir quantos veículos serão necessários para realizar as entregas de todas as demandas. Isso que possibilita, além da otimização das rotas, a otimização para a utilização de uma quantidade mínima de veículos para as entregas.

As melhores soluções PCV geradas pela metaheurística BRKGA são transformadas em soluções PRV, com base nas capacidades e demandas do clientes da instância. Essas soluções PRV podem ser representadas nos formatos de vetor (rota completa) e matriz

(sub-rotas), sendo enviadas para o módulo do LKH-3 para soluções iniciais. O LKH-3, ao ler essas soluções iniciais, irá realizar os procedimentos de otimização. Os cálculos do *fitness* de cada possível solução com o LKH-3 é apresentado na figura 15.

Figura 15 – Cálculo *fitness* das Rotas PRV



Fonte: Autoria própria

5.5 Aplicação ao Problema de Roteamento de Veículos com Entregas Fracionadas

A solução com melhor valor de *fitness* gerada ao final das gerações do BRKGA, e otimizada pelo LKH-3, é lida como solução inicial pelo *solver AlkaidSD*, o que substitui a heurística de construção de solução inicial descrita no trabalho original (LIN *et al.*, 2025). Dessa forma, o *solver* já recebe, como entrada, uma solução sub-ótima ou muito boa da instância do CVRP e, então, segue com as iterações do ILS-RVND. Ao substituir o algoritmo de construção de solução inicial do *solver* pela solução gerada pelo BRKGA com LKH, o custo computacional, com relação ao tempo de processamento, tem a possibilidade de ser menor do que utilizando a heurística de construção do *solver* na geração de uma solução.

No processo do *framework*, uma solução passa por todos os passos da heurística híbrida do ILS-RVND e pelas estruturas de trocas e perturbações descritas anteriormente. Ao final das iterações do *framework*, o *solver* retorna, como solução, a última solução de melhor valor objetivo (*fitness*). Se houve possibilidade de a solução ter tido o fracionamento, então o valor objetivo, que é a soma das distâncias das sub-rotas, é retornado. Se não, o valor objetivo normal da solução CVRP é mostrado. Ao final, o *solver* retorna, além do valor objetivo, um arquivo contendo a melhor solução encontrada, com as sequências

de sub-rotas completas.

5.6 Aplicação ao Problema de Roteamento de Veículos com Drones

O módulo do *framework* deste trabalho que adiciona os drones recebe, como entrada, a melhor solução encontrada pelo *solver AlkaidSD*. O próximo passo é a aplicação da heurística de busca local LKH para realizar a otimização da solução lida, escolhendo quais vértices de cada sub-rotas serão atendidos pelos drones. Em uma sub-rotas, a heurística seleciona uma quantidade de vértices e realiza a troca para sub-rotas ramificadas. Neste passo, é utilizada a restrição de que, um drone, ao atender um cliente, deverá voltar ao veículo para que possa atender a outro cliente. Ou seja, um drone apenas pode atender uma demanda de cada vez. As trocas, ao serem feitas, fazem com que os vértices escolhidos sejam removidos da sub-rotas, o que significa uma reconexão entre as sequências, reduzindo a distância total daquela sub-rotas. A redução na distância se dá pois a rota de um drone é paralela à principal, o que faz com que o drone atenda sua demanda enquanto o veículo continua atendendo outras.

A cada iteração do LKH a sub-rotas escolhida é reiniciada e a heurística refaz as trocas dos vértices que serão atendidos por drone. Se em uma iteração o valor da distância da sub-rotas com veículo e drone é menor que o valor encontrado antes, os vértices escolhidos são salvos como melhores candidatos. Ao final das iterações, o melhor valor de *fitness* encontrado é salvo e a sub-rotas e as rotas ramificadas com os vértices escolhidos referentes a esse valor são atualizadas.

O processo descrito repete-se para todas as sub-rotas descritas. Assim que todos os passos são finalizados, o algoritmo que contém a heurística LKH retorna o conjunto de rotas como melhor solução e seu valor objetivo, que é a distância total, soma das distâncias das sub-rotas. O valor objetivo e o tempo referente para sua solução serão os valores utilizados de cada instância neste trabalho para realizar comparações com os demais trabalhos relacionados nos testes computacionais.

A adição de drones ao contexto do problema possibilita, ainda mais, a melhora na otimização das rotas, pois os drones farão parte das entregas enquanto um veículo está realizando as demais. Drones têm limites de capacidade e tempo de carga de bateria. Porém, como rapidez e condução aérea, além de poderem realizar parte ou todas as entregas de mercadorias pequenas ou leves, podem atingir áreas de difícil acesso ou que tornariam

a entrega por veículos impossível, como lugares que só podem ser visitados por vias aquáticas. A aplicação de drones terá as representações de tempo e capacidade nos dados para serem utilizados pelo algoritmo.

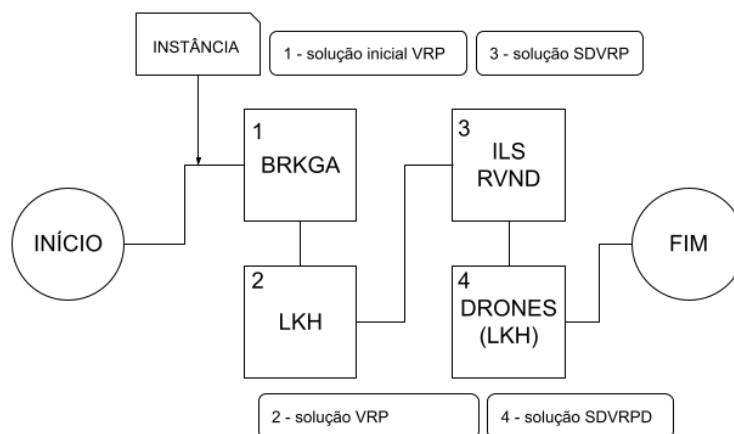
As aplicações seguirão a seguinte ordem:

- a) Aplicação do BRKGA ao PCV;
- b) Gerar soluções para o PRV;
- c) Otimizar as soluções PRV com LKH;
- d) Gerar soluções SDVRP com ILS-RVND
- e) A partir das soluções SDVRP, gerar soluções com drones com o LKH.

Os dados das instâncias, os vetores e matrizes de valores nos tipos inteiro, ponto flutuante e demais tipos serão necessários para representar as quantidades de veículos, drones, clientes, bem como seus respectivos valores de capacidades, demandas.

A figura 16 apresenta o fluxograma da implementação deste trabalho, mostrando a sequência entre os módulos e o as soluções que cada módulo realiza.

Figura 16 – Fluxograma deste Projeto



Fonte: Autoria própria

6 RESULTADOS

Neste capítulo são apresentados os resultados dos testes computacionais realizados do projeto deste trabalho com as instâncias escolhidas. Os testes são baseados nos valores de melhor *fitness* e o tempo ao qual esses valores foram encontrados. Os resultados dos testes estão dispostos em tabelas, que apresentam comparações entre as heurísticas, metaheurísticas e com os resultados de outros trabalhos relacionados.

6.1 Ambiente de testes

A implementação deste projeto e os testes realizados foram feitos no sistema operacional *Microsoft Windows 11 - 64 bit*. As configurações do *hardware* são: processador *12th Gen Intel(R) Core(TM) i5-12450H - 2.00 GHz*; memória RAM de 8,00 GB. Para a codificação do projeto na linguagem C++, foi utilizado o *Visual Studio 2022*, ambiente de desenvolvimento integrado (IDE) da *Microsoft*.

6.2 Dados e Parâmetros

As instâncias *eilon* utilizadas para os testes computacionais representam um total de 11 arquivos do tipo *.vrp*, classificados para o CVRP, com dimensões de vértices variando entre 22 e 101. Cada arquivo tem os atributos específicos que informam: quantidade de vértices (DIMENSION), tipo de aresta (EDGE_WEIGHT_TYPE), capacidade do veículo (CAPACITY), sessão com coordenadas dos vértices (NODE_COORD_SECTION), sessão com as demandas dos vértices (DEMAND_SECTION) e a sessão qual vértice representa o depósito (DEPOT_SECTION).

O *framework* do projeto deste trabalho utiliza os arquivos *.vrp* em quase todos os seus módulos, com exceção do módulo *solver AlkaidSD* (ILS/RVND), que utiliza o tipo *.sd*, disponibilizado pelo desafio do DIMACS.

As instâncias lidas pelos módulos e os valores de *capacidade*, *demandas* e vértice do *depósito* são armazenados, além das coordenadas euclidianas (tipo de aresta) de todos vértices, que são utilizadas no cálculo da matriz de distâncias.

Seguindo o fluxograma do projeto, o primeiro módulo implementa metaheurística BRKGA aplicada ao PRV, primeiramente gerando chaves aleatórias referentes à quantidade de vértices clientes ($clientes = dimensao - deposito$), adicionadas em cromossomos já

iniciados com a chave 0, referente ao depósito. Com os valores gerados e a decodificação, um algoritmo soma as demandas e define cada rota até a restrição da capacidade. Esse processo continua até todos os vértices clientes terem suas demandas somadas.

Os parâmetros utilizados para o BRKGA nos testes variam de acordo com a dimensão de cada instância. Quanto maior a dimensão de uma instância, maior deve ser os valores p para a população para que a metaheurística tenha bons resultados. Assim, são utilizados os valores dentro dos intervalos recomendados por Gonçalves e Resende (2011), onde:

- p (tamanho da população) = an , sendo a uma constante e n dimensão da instância;
- pe (tamanho da população elite) = $0.25p$;
- pm (tamanho da população mutante) = $0.30p$;
- pe (probabilidade de escolha de alelo elite) = 0.7 .

Como as instâncias têm dimensões que variam de 22 a 101 vértices, os valores utilizados para a constante a variam de 2.0 a 17, indicando o cálculo do tamanho p das populações.

Os parâmetros utilizados para o LKH-3 estão definidos no arquivo *.par*, nos atributos: INITIAL_TOUR_FILE, MAX_TRIALS, MOVE_TYPE, TIME_LIMIT, RUNS. O parâmetro INITIAL_TOUR_FILE indica o arquivo *.tour* com uma solução gerada pelo módulo do BRKGA.

O módulo *AlkaidSD* (ILS-RVND) teve o código original editado para receber a solução do LKH-3 como solução inicial. Essa solução inicial é recebida como um parâmetro junto ao arquivo de configuração do projeto original para cada instância.

6.3 Testes computacionais

A seguir estão apresentados os resultados dos testes computacionais nas tabelas. Os resultados incluem os testes unitários de cada módulo e com a utilização híbrida BRKGA + LKH-3, BRKGA + LKH-3 + AlkaidSD e, por último, o teste completo com a adição dos drones com LKH. Na tabela 7 são apresentados os resultados unitários dos módulos BRKGA e LKH-3, além do resultado híbrido BRKGA + LKH-3.

Com os resultados da tabela 7 mostrados, algumas instâncias têm o tempo de solução *fitness* bem reduzido para o LKH-3 quando já recebe uma solução inicial do

Tabela 7 – Testes com Metaheurísticas

Instância	BRKGA		LKH-3		BRKGA + LKH-3	
	fitness	tempo	fitness	tempo	fitness	tempo
eil22	425	0.30	375	0.15	375	0.83
eil23	704	0.34	569	0.08	569	0.12
eil30	634	0.73	534	3.07	534	1.65
eil33	1020	0.84	535	0.97	535	22.82
eil51	785	3.45	821	5.69	821	8.67
eilA76	1068	12.09	836	67.31	836	60.64
eilB76	1429	11.15	1032	78.07	1036	35.53
eilC76	1150	14.29	736	37.29	736	10.62
eilD76	1259	11.04	682	1.36	682	6.55
eilA101	1246	25.49	819	16.77	815	22.51
eilB101	1429	39.02	1083	85.47	1080	31.68

BRKGA, com relação em apenas o próprio módulo. São exemplos as instâncias *eilB76*, *eilC76* e *eilB101*.

Os resultados dos testes unitários do módulo *AlkaidSD* e os testes com a utilização híbrida BRKGA + LKH-3 + *AlkaidSD* são mostrados na tabela .

Tabela 8 – Comparação entre Metaheurísticas

Instância	<i>AlkaidSD</i>		BRKGA + LKH-3 + <i>AlkaidSD</i>	
	fitness	tempo	fitness	tempo
eil22	375	0.001	375	0.010
eil23	569	0.0195	569	0.001
eil30	503	0.203	503	0.022
eil33	835	0.058	835	0.002
eil51	521	0.186	521	0.001
eilA76	818	46.156	818	5.763
eilB76	1002	23.109	1002	9.933
eilC76	733	3.884	733	3.884
eilD76	679	35.859	679	30.877
eilA101	814	5.140	814	0.010
eilB101	1061	33.378	1060	52.915

Os resultados da tabela 8 indicam que o tempo necessário para encontrar o *fitness* foi menor com a abordagem híbrida para maioria das instâncias. O tempo das instâncias *eilA76* e *eilA101*, por exemplo, foi bastante reduzido com a utilização híbrida dos módulos.

A seguir, na tabela 9 são apresentados os resultados dos testes computacionais deste trabalho em comparação com os trabalhos de Silva *et al.* (2015) e Costa (2023) para o SDVRP (sem drones).

A tabela 9 mostrou que o trabalho de Silva *et al.* (2015) teve melhor resultado em

Tabela 9 – Testes Computacionais SDVRP (sem drones)

Instância	Silva (2013)		Costa (2023)		Este trabalho	
	fitness	tempo	fitness	tempo	fitness	tempo
eil22	375	0,13	375	0,3975	375	0.010
eil23	569	0,09	569	0,4050	569	0.001
eil30	510	0,30	510	0,4275	510	0.022
eil33	835	0,39	830*	0,4200	835	0.002
eil51	521	1,63	521	1,4350	521	0.001
eilA76	818	25,68	818	7,7840	818	5.736
eilB76	1002	38,05	1002	16,5350	1002	9.933
eilC76	733*	15,17	735	6,3700	733*	3.884
eilD76	681	11,02	681	6,2160	679*	30.887
eilA101	815	32,70	813*	17,4840	814	0.010
eilB101	1061	75,43	1061	18,8825	1060*	52.915

apenas 1 instância comparado ao trabalho de Costa (2023), a instância *eilC76*. O trabalho de Costa (2023), por sua vez, conseguiu ter melhor resultado para 2 instâncias, *eil33* e *eilA101*, com relação a Silva *et al.* (2015) e a este trabalho. Este trabalho teve melhor resultado para 2 instâncias com relação ao trabalho de Silva *et al.* (2015) e para 3 instâncias, com relação ao trabalho de Costa (2023). Foram as instâncias *eilC76*, *eilD76* e *eilB101*.

Os próximos resultados incluem os drones ao SDVRP, se tornando o SDVRPD. As comparações são feitos utilizando o trabalho de Costa (2023). Os resultados são mostrados na tabela 10.

Tabela 10 – Testes Computacionais SDVRPD (com drones)

Instância	Costa (2023)		Este trabalho	
	fitness	tempo	fitness	tempo
eil22	267,18	0,78	252.677*	0.014
eil23	405,41	0,86	382.391*	0.004
eil30	337,02	0,92	326.872*	0.024
eil33	563,4	0,88	677.349	0.005
eil51	309,61	2,01	302.49*	0.005
eilA76	485,09	9,89	462.109*	5.340
eilB76	405,49	21,07	609.611	9.939
eilC76	401,32	7,64	398.435*	3.887
eilD76	371,82	7,12	408.765	30.890
eilA101	415,6	19,59	490.852	0.019
eilB101	508,48	20,40	592.828	52.921

De acordo com a tabela 10, este trabalho teve melhor resultados nos testes para 6 das 11 instâncias, com relação ao trabalho de Costa (2023). Foram as instâncias *eil22*, *eil23*, *eil30*, *eil51*, *eilA76* e *eilC76*. Assim, este trabalho teve melhor resultado para

maioria das instâncias entre 22 e 51 vértices, apenas duas instâncias com 76 vértices e nenhuma instância com 101 vértices.

Sobre as comparações com o SDVRP, na tabela 9, é possível notar que há uma maior tendência de fracionamento de acordo com a dimensão da instância. Ou seja, quanto maior quantidade de vértices, maior a possibilidade de fracionamento, o que é visto em todos os trabalhos. Nos resultados apenas com o SDVRP, as técnicas utilizadas neste trabalho demonstraram bons resultados em relações aos outros dois trabalhos, visto uma pequena vantagem para algumas instâncias.

Com a análise dos resultados da tabela 10, é correto afirmar que as técnicas utilizadas neste trabalho promovem bons resultados na inclusão dos drones em relação ao outro trabalho, tendo apenas menor eficiência com relação às instâncias com dimensão de 101 vértices. Mesmo assim, os resultados são satisfatórios e podem indicar avanços e possíveis ajustes em trabalhos futuros.

7 CONCLUSÕES

Os estudos sobre Problema de Roteamento de Veículo são de extrema importância para as áreas de Pesquisa Operacional, Teoria dos Grafos e Inteligência Artificial, pois o crescimento das inovações tecnológicas tem trazido uma gama de benefícios que possibilitam as reduções de custos e tempo. Além das abordagens clássicas do problema, como o Problema do Caixeiro Viajante, novas variantes e generalizações foram propostas ao longo dos anos. Uma dessas variantes aborda a possibilidade de redução do custo total do problema ao permitir que os clientes sejam visitados por mais de um veículo, a variante PRV com Entregas Fracionadas.

Outra variante que tem sido bastante abordada nos últimos anos, diante da evolução tecnológica, é o PRV com drones, que inclui os drones junto aos veículos para cooperarem no atendimento das demandas dos clientes. A possibilidade de redução de custos operacionais com essa nova abordagem tem sido de extrema importância na busca por propostas de otimização. O PRV com Entregas Fracionadas e drones, variante recente que integra a eficiência das duas últimas citadas, é bastante promissora para o setor de logística, pois possibilita uma maior redução dos custos. Dessa forma, o SDVRPD apresenta ser um problema promissor para a literatura.

Neste trabalho foi realizada a implementação de um *framework* que utiliza as técnicas aproximativas metaheurísticas e heurísticas BRKGA, LKH e ILS-RVND aplicadas à solução do Problema de Roteamento de Veículos com Entregas Fracionadas e Drones SDVRPD. Foram utilizadas implementações próprias com integração de *frameworks* de terceiros. Os resultados, tanto para o SDVRP (sem drones) quanto para o SDVRPD (com drones), indicam a eficiência do *framework* nas comparações entre os trabalhos relacionados utilizados.

Como o SDVRPD é uma abordagem recente na literatura, a disponibilidade de pesquisas e de dados para testes é bastante reduzida, sendo uma limitação para esta pesquisa. Este trabalho contribui para a literatura, tanto na abordagem dessa variação do PRV quanto para a implementação das heurísticas e metaheurísticas, com a utilização, integração e adaptação dos *frameworks* de terceiros para a solução do problema.

Para trabalhos futuros, os testes do problema SDVRPD podem ser testados com uma variedade de heurísticas e metaheurísticas, com adaptação e configurações de acordo com as dimensões das instâncias de teste.

REFERÊNCIAS

- AGATZ, N.; BOUMAN, P.; SCHMIDT, M. Optimization approaches for the traveling salesman problem with drone. **Transportation Science**, INFORMS, v. 52, n. 4, p. 965–981, 2018.
- APPLEGATE, D. L. **The traveling salesman problem: a computational study**. [S.l.]: Princeton University Press, 2006. v. 17.
- ARCHETTI, C.; SAVELSBERGH, M. W.; SPERANZA, M. G. Worst-case analysis for split delivery vehicle routing problems. **Transportation science**, INFORMS, v. 40, n. 2, p. 226–234, 2006.
- BEAN, J. C. Genetic algorithms and random keys for sequencing and optimization. **ORSA journal on computing**, INFORMS, v. 6, n. 2, p. 154–160, 1994.
- BEKTAS, T. The multiple traveling salesman problem: an overview of formulations and solution procedures. **omega**, Elsevier, v. 34, n. 3, p. 209–219, 2006.
- BELENGUER, J. M.; MARTINEZ, M. C.; MOTA, E. A lower bound for the split delivery vehicle routing problem. **Operations Research**, v. 48, n. 5, p. 801–810, 2000.
- BENARBIA, T.; KYAMAKYA, K. A literature review of drone-based package delivery logistics systems and their implementation feasibility. **Sustainability**, MDPI, v. 14, n. 1, p. 360, 2021.
- CHEIKHROUHOU, O.; KHOUFI, I. A comprehensive survey on the multiple traveling salesman problem: Applications, approaches and taxonomy. **Computer Science Review**, Elsevier, v. 40, p. 100369, 2021.
- CHRISTOFIDES, N.; EILON, S. An algorithm for the vehicle-dispatching problem. **Journal of the Operational Research Society**, Taylor & Francis, v. 20, n. 3, p. 309–318, 1969.
- COSTA, D. F. **Uma abordagem de meta-heurística evolutiva para a otimização de rotas de entrega fracionada de mercadorias utilizando caminhões e drones**. 53 p. Dissertação (Mestrado) — Programa de Pós-graduação em Ciência da Computação - Universidade do Estado do Rio Grande do Norte; Universidade Federal Rural do Semi-Árido, Mossoró, RN, Brasil, 2023.
- COSTA, J. G. C. **The vehicle routing problem with drones**. 124 p. Dissertação (Dissertação de Mestrado) — Instituto de Ciências Matemáticas e de Computação, Universidade de São Paulo, São Carlos, SP, Brasil, 2019.
- DANG, S.; LIU, Y.; LUO, Z.; LIU, Z.; SHI, J. A survey of the routing problem for cooperated trucks and drones. **Drones**, MDPI, v. 8, n. 10, p. 550, 2024.
- DANTZIG, G.; FULKERSON, R.; JOHNSON, S. Solution of a large-scale traveling-salesman problem. **Journal of the operations research society of America**, INFORMS, v. 2, n. 4, p. 393–410, 1954.
- DANTZIG, G. B. Linear programming and extensions. In: **Linear programming and extensions**. [S.l.]: Princeton university press, 2016.

DANTZIG, G. B.; RAMSER, J. H. The truck dispatching problem. **Management Science**, v. 6, p. 80–91, 10 1959. ISSN 0025-1909.

DORLING, K.; HEINRICHS, J.; MESSIER, G. G.; MAGIEROWSKI, S. Vehicle routing problems for drone delivery. **IEEE Transactions on Systems, Man, and Cybernetics: Systems**, v. 47, p. 70–85, 1 2017. ISSN 2168-2216.

DROR, M.; TRUDEAU, P. Savings by split delivery routing. **Transportation Science, INFORMS**, v. 23, n. 2, p. 141–145, 1989.

DROR, M.; TRUDEAU, P. Split delivery routing. **Naval Research Logistics (NRL)**, Wiley Online Library, v. 37, n. 3, p. 383–402, 1990.

ELSHAER, R.; AWAD, H. A taxonomic review of metaheuristic algorithms for solving the vehicle routing problem and its variants. **Computers & Industrial Engineering**, v. 140, p. 106242, 2 2020. ISSN 03608352.

GAREY, M. R.; JOHNSON, D. S. **Computers and intractability**. [S.l.]: freeman San Francisco, 1979. v. 174.

GONÇALVES, J. F.; RESENDE, M. G. Biased random-key genetic algorithms for combinatorial optimization. **Journal of Heuristics**, v. 17, p. 487–525, 10 2011. ISSN 13811231.

GOUVEIA, L.; LEITNER, M.; RUTHMAIR, M. Multi-depot routing with split deliveries: Models and a branch-and-cut algorithm. **Transportation Science, INFORMS**, v. 57, n. 2, p. 512–530, 2023.

GSCHWIND, T.; BIANCHESSI, N.; IRNICH, S. Stabilized branch-price-and-cut for the commodity-constrained split delivery vehicle routing problem. **European Journal of Operational Research**, Elsevier, v. 278, n. 1, p. 91–104, 2019.

HELGAUN, K. An effective implementation of the lin–kernighan traveling salesman heuristic. **European journal of operational research**, Elsevier, v. 126, n. 1, p. 106–130, 2000.

HELGAUN, K. An extension of the lin-kernighan-helgaun tsp solver for constrained traveling salesman and vehicle routing problems. **Roskilde: Roskilde University**, v. 12, p. 966–980, 2017.

HOLLAND, J. Adaptation in natural and artificial systems. **MIT Press**, 1975.

ILAVARASI, K.; JOSEPH, K. S. Variants of travelling salesman problem: A survey. In: IEEE. **International conference on information communication and embedded systems (ICICES2014)**. [S.l.], 2014. p. 1–7.

JI, B.; ZHOU, S.; SAMSON, S. Y.; WU, G. An enhanced neighborhood search algorithm for solving the split delivery vehicle routing problem with two-dimensional loading constraints. **Computers & Industrial Engineering**, Elsevier, v. 162, p. 107720, 2021.

KARP, R. M. **Reducibility among combinatorial problems**. [S.l.]: Springer, 2010.

KITJACHAROENCHAI, P.; VENTRESCA, M.; MOSHREF-JAVADI, M.; LEE, S.; TANCHOCO, J. M.; BRUNESE, P. A. Multiple traveling salesman problem with drones: Mathematical model and heuristic approach. **Computers & Industrial Engineering**, v. 129, p. 14–30, 3 2019. ISSN 03608352.

KONSTANTAKOPOULOS, G. D.; GAYIALIS, S. P.; KECHAGIAS, E. P. Vehicle routing problem and related algorithms for logistics distribution: a literature review and classification. **Operational Research**, v. 22, p. 2033–2062, 7 2022. ISSN 1109-2858.

KUO, R.; LU, S.-H.; LAI, P.-Y.; MARA, S. T. W. Vehicle routing problem with drones considering time windows. **Expert Systems with Applications**, Elsevier, v. 191, p. 116264, 2022.

LAGHARI, A. A.; JUMANI, A. K.; LAGHARI, R. A.; NAWAZ, H. Unmanned aerial vehicles: A review. **Cognitive Robotics**, v. 3, p. 8–22, 2023. ISSN 26672413.

LENSTRA, J. K.; KAN, A. R. Complexity of vehicle routing and scheduling problems. **Networks**, Wiley Online Library, v. 11, n. 2, p. 221–227, 1981.

LIN, S.; KERNIGHAN, B. W. An effective heuristic algorithm for the traveling-salesman problem. **Operations research**, Informs, v. 21, n. 2, p. 498–516, 1973.

LIN, W.; HE, Z.; JIANG, S.; MA, F.; SU, Z.; LÜ, Z. Alkaid-sdvrp: An efficient open-source solver for the vehicle routing problem with split deliveries. **INFORMS Journal on Computing**, INFORMS, 2025.

LOURENÇO, H. R.; MARTIN, O. C.; STÜTZLE, T. Iterated local search. In: **Handbook of metaheuristics**. [S.l.]: Springer, 2003. p. 320–353.

MACHARET, D. G.; CAMPOS, M. F. M. A survey on routing problems and robotic systems. **Robotica**, v. 36, p. 1781–1803, 12 2018. ISSN 0263-5747.

MAHMOUDI, B.; ESHGHI, K. The multi-visit split delivery VRP with drones considering en-route launches and rendezvous: Application to post-disaster relief operations. **Computers & Industrial Engineering**, v. 206, 2025.

MLADENović, N.; HANSEN, P. Variable neighborhood search. **Computers & operations research**, Elsevier, v. 24, n. 11, p. 1097–1100, 1997.

MOHSAN, S. A. H.; KHAN, M. A.; NOOR, F.; ULLAH, I.; ALSHARIF, M. H. Towards the unmanned aerial vehicles (uavs): A comprehensive review. **Drones**, MDPI, v. 6, n. 6, p. 147, 2022.

MOHSAN, S. A. H.; KHAN, M. A.; NOOR, F.; ULLAH, I.; ALSHARIF, M. H. Towards the unmanned aerial vehicles (uavs): A comprehensive review. **Drones**, v. 6, p. 147, 6 2022. ISSN 2504-446X.

MURRAY, C. C.; CHU, A. G. The flying sidekick traveling salesman problem: Optimization of drone-assisted parcel delivery. **Transportation Research Part C: Emerging Technologies**, Elsevier, v. 54, p. 86–109, 2015.

NEWCOME, L. R. **Unmanned aviation: a brief history of unmanned aerial vehicles**. [S.l.]: Aiaa, 2004.

NGUYEN, M. A.; DANG, G. T.-H.; HÀ, M. H.; PHAM, M.-T. The min-cost parallel drone scheduling vehicle routing problem. **European Journal of Operational Research**, Elsevier, v. 299, n. 3, p. 910–930, 2022.

OSABA, E.; YANG, X.-S.; SER, J. D. Traveling salesman problem: a perspective review of recent research and new results with bio-inspired metaheuristics. **Nature-inspired computation and swarm intelligence**, Elsevier, p. 135–164, 2020.

PENNA, P. H. V.; SUBRAMANIAN, A.; OCHI, L. S. An iterated local search heuristic for the heterogeneous fleet vehicle routing problem. **Journal of Heuristics**, Springer, v. 19, n. 2, p. 201–232, 2013.

POIKONEN, S.; WANG, X.; GOLDEN, B. The vehicle routing problem with drones: Extended models and connections. **Networks**, Wiley Online Library, v. 70, n. 1, p. 34–43, 2017.

POP, P. C.; COSMA, O.; SABO, C.; SITAR, C. P. A comprehensive survey on the generalized traveling salesman problem. **European Journal of Operational Research**, Elsevier, 2023.

POP, P. C.; COSMA, O.; SABO, C.; SITAR, C. P. A comprehensive survey on the generalized traveling salesman problem. **European Journal of Operational Research**, v. 314, p. 819–835, 5 2024. ISSN 03772217.

REINHELT, G. {TSPLIB}: a library of sample instances for the tsp (and related problems) from various sources and of various types. **URL: <http://comopt.ifl.uni-heidelberg.de/software/TSPLIB95/>**, 2014.

REN, T.; LUO, T.; JIA, B.; YANG, B.; WANG, L.; XING, L. Improved ant colony optimization for the vehicle routing problem with split pickup and split delivery. **Swarm and Evolutionary Computation**, Elsevier, v. 77, p. 101228, 2023.

RESENDE, M. G. Introdução aos algoritmos genéticos de chaves aleatórias viciadas. **Simpósio Brasileiro de Pesquisa Operacional**, p. 16–19, 2013.

SACRAMENTO, D.; PISINGER, D.; ROPKE, S. An adaptive large neighborhood search metaheuristic for the vehicle routing problem with drones. **Transportation Research Part C: Emerging Technologies**, Elsevier, v. 102, p. 289–315, 2019.

SARBIJAN, M. S.; BEHNAMIAN, J. Emerging research fields in vehicle routing problem: a short review. **Archives of Computational Methods in Engineering**, Springer, v. 30, n. 4, p. 2473–2491, 2023.

SILVA, M. M.; SUBRAMANIAN, A.; OCHI, L. S. An iterated local search heuristic for the split delivery vehicle routing problem. **Computers & Operations Research**, Elsevier, v. 53, p. 234–249, 2015.

SILVA, M. M.; SUBRAMANIAN, A.; VIDAL, T.; OCHI, L. S. A simple and effective metaheuristic for the minimum latency problem. **European Journal of Operational Research**, Elsevier, v. 221, n. 3, p. 513–520, 2012.

SINGHAL, G.; BANSOD, B.; MATHEW, L. Unmanned aerial vehicle classification, applications and challenges: A review. 2018. Disponível em: <<http://revistapesquisa.fapesp.br>>.

SONG, W.; QIU, H.; LI, M. Review on vehicle routing planning for truck-unmanned vehicle collaborative distribution. In: IEEE. **2025 8th International Conference on Advanced Algorithms and Control Engineering (ICAACE)**. [S.l.], 2025. p. 856–862.

SUBRAMANIAN, A.; CABRAL, L.; OCHI, L. An efficient ils heuristic for the vehicle routing problem with simultaneous pickup and delivery. **Relatório Técnico, Universidade Federal Fluminense, disponível em [http://www. ic. uff. br/~ satoru/index. php](http://www.ic.uff.br/~satoru/index.php)**, 2008.

SUBRAMANIAN, A.; DRUMMOND, L. M.; BENTES, C.; OCHI, L. S.; FARIAS, R. A parallel heuristic for the vehicle routing problem with simultaneous pickup and delivery. **Computers and Operations Research**, v. 37, p. 1899–1911, 11 2010. ISSN 03050548.

TAMKE, F.; BUSCHER, U. A branch-and-cut algorithm for the vehicle routing problem with drones. **Transportation Research Part B: Methodological**, Elsevier, v. 144, p. 174–203, 2021.

VILORIA, D. R.; SOLANO-CHARRIS, E. L.; MUÑOZ-VILLAMIZAR, A.; MONTROYA-TORRES, J. R. Unmanned aerial vehicles/drones in vehicle routing problems: a literature review. **International Transactions in Operational Research**, Wiley Online Library, v. 28, n. 4, p. 1626–1657, 2021.

WANG, X.; POIKONEN, S.; GOLDEN, B. The vehicle routing problem with drones: several worst-case results. **Optimization Letters**, v. 11, p. 679–697, 4 2017. ISSN 1862-4472.

WANG, Z.; SHEU, J.-B. Vehicle routing problem with drones. **Transportation research part B: methodological**, Elsevier, v. 122, p. 350–364, 2019.

WINSTON, W. L. **Operations research: applications and algorithm**. [S.l.]: Thomson Learning, Inc., 2004.

WU, D.; WU, C. Research on the time-dependent split delivery green vehicle routing problem for fresh agricultural products with multiple time windows. **Agriculture**, MDPI, v. 12, n. 6, p. 793, 2022.