

UNIVERSIDADE FEDERAL FLUMINENSE

MICHELANGELO JOSÉ DE MELLO LIMA E SOUZA LOPES

**Problema de Roteirização de Veículos
Multi-Depósito com Drones: replicabilidade,
revalidação e novas estratégias**

NITERÓI

2026

MICHELANGELO JOSÉ DE MELLO LIMA E SOUZA LOPES

**Problema de Roteirização de Veículos
Multi-Depósito com Drones: replicabilidade,
revalidação e novas estratégias**

Dissertação de Mestrado apresentada ao Programa de Pós-Graduação em Computação da Universidade Federal Fluminense como requisito parcial para a obtenção do Grau de Mestre em Computação. Área de concentração: Ciência da Computação.

Orientador:
Fábio Protti

Coorientador:
Luiz Satoru Ochi

NITERÓI

2026

Ficha catalográfica automática - SDC/BEE
Gerada com informações fornecidas pelo autor

L864p Lopes, Michelangelo José de Mello Lima e Souza
 Problema de Roteirização de Veículos Multi-Depósito com
 Drones: replicabilidade, revalidação e novas estratégias /
 Michelangelo José de Mello Lima e Souza Lopes. - 2026.
 94 f.: il.

 Orientador: Fábio Protti.
 Coorientador: Luiz Satoru Ochi.
 Dissertação (mestrado)-Universidade Federal Fluminense,
 Instituto de Computação, Niterói, 2026.

 1. Problema de Roteirização de Veículos. 2. Veículo
 Aéreo Não Tripulado (VANT). 3. Meta-heurística. 4.
 Otimização (Computação). 5. Produção intelectual. I.
 Protti, Fábio, orientador. II. Ochi, Luiz Satoru,
 coorientador. III. Universidade Federal Fluminense. Instituto
 de Computação. IV. Título.

CDD - XXX

MICHELANGELO JOSÉ DE MELLO LIMA E SOUZA LOPES

Problema de Roteirização de Veículos Multi-Depósito com Drones: replicabilidade,
revalidação e novas estratégias

Dissertação de Mestrado apresentada ao Programa de Pós-Graduação em Computação da Universidade Federal Fluminense como requisito parcial para a obtenção do Grau de Mestre em Computação. Área de concentração: Ciência da Computação.

Aprovada em julho de 2026.

BANCA EXAMINADORA

Prof. Fábio Protti, Orientador, UFF

Prof. Luiz Satoru Ochi, Co-orientador, UFF

Prof. Igor Coelho, UFF

Prof. Luidi Simonetti, UFRJ

Niterói

2026

À Daniela, pelo amor, pela paciência e por estar ao meu lado nesta jornada.

Agradecimentos

Agradeço, em primeiro lugar, a Deus, por me dar forças e clareza nos momentos em que mais precisei ao longo desta jornada.

À minha esposa Daniela, por ter vivido de perto cada fase desta jornada. Quando o mestrado começou, ainda éramos namorados; ao defendê-lo, somos casados e construímos nosso lar juntos. Você acompanhou cada adiamento, cada recomeço, e nunca deixou de acreditar no final desta história.

Ao meu pai, Walter, pelo incentivo a fazer o mestrado desde que eu era pequeno, e à minha mãe, Silvana, por todo o carinho e dedicação. A vocês dois, agradeço pelo lar que me ofereceram durante a fase mais intensa do mestrado, onde enfrentei os longos trajetos diários e aprendi a conciliar trabalho e aulas ao mesmo tempo. Ao meu irmão mais novo, Nicholas, agradeço pelo apoio silencioso na nossa convivência cotidiana. Essa base, feita de amor, carinho e exemplo, é o que me sustentou para chegar até aqui.

Ao meu orientador, Prof. Fábio Protti, pela orientação precisa e, sobretudo, pela enorme paciência que teve comigo. Não fui o orientando mais fácil: as demandas do trabalho muitas vezes falaram mais alto, e foram necessárias três prorrogações até chegarmos ao ponto atual. Ainda assim, nunca deixou de acreditar na conclusão deste trabalho, e por isso sou profundamente grato.

Ao meu co-orientador, Prof. Luiz Satoru Ochi, pela orientação na fase inicial desta pesquisa e pelo retorno precioso na fase final, com indicações de ajustes e dicas valiosas para a apresentação. Sua contribuição foi decisiva no processo de escolha e delimitação do tema, ajudando a moldar a direção que este trabalho tomou.

Resumo

Esta dissertação investiga o Problema de Roteirização de Veículos Multi-Depósito com Drones (MDVRP-D), variante logística que integra caminhões e drones para entregas de última milha. O trabalho conduz uma replicação e um estudo de ablação do algoritmo AACO-NC-D (*Adaptive Ant Colony Optimization with Node Clustering for Drones*), estado da arte para o problema. Durante a replicação, identificou-se que a restrição de autonomia do drone admite duas interpretações operacionais: a Variante A contabiliza o tempo de serviço do caminhão no ponto de recuperação como consumo de bateria do drone (correspondente à formalização matemática original); a Variante B não o faz. A auditoria computacional das 1.150 soluções publicadas confirmou que estas são aderentes à Variante B. Com base nessa distinção, a pesquisa estabelece os primeiros *baselines* comprovadamente válidos sob a Variante A para as instâncias de médio e grande porte do problema.

Como contribuições metodológicas, investigam-se dois eixos principais. O primeiro abrange a clusterização de nós: formaliza-se a Setorização Angular Gulosa descrita de forma ambígua no modelo original e propõe-se a Partição de Voronoi com mecanismo de diversidade adaptativa, que demonstrou superioridade na qualidade média do *makespan* em 22 das 23 instâncias avaliadas. O segundo eixo investiga a decomposição do problema em sub-instâncias por depósito sob três modos operacionais: decomposição pura (executada de forma autônoma), decomposição como semente (inicializando as matrizes de feromônio no espaço completo) e decomposição como refinamento (polimento pós-convergência global).

Os resultados experimentais confirmam que a Partição de Voronoi e a decomposição como refinamento proporcionam melhorias significativas na qualidade do *makespan* e no desempenho computacional. O trabalho estabelece uma base metodológica rigorosa e reproduzível para futuras pesquisas no MDVRP-D.

Palavras-chave: MDVRP-D, Roteirização com Drones, Meta-heurísticas, ACO, Partição de Voronoi, Decomposição.

Abstract

This dissertation investigates the Multi-Depot Vehicle Routing Problem with Drones (MDVRP-D), a logistics variant that integrates trucks and drones for last-mile deliveries. The work conducts a replication study and ablation analysis of the AACO-NC-D (*Adaptive Ant Colony Optimization with Node Clustering for Drones*), the current state-of-the-art algorithm. During replication, it was identified that the drone battery constraint admits two operational interpretations: Variant A accounts for truck service time at the rendezvous node as drone endurance consumption, consistent with the original mathematical formulation; Variant B does not. A computational audit of the 1,150 published solutions confirmed their adherence to Variant B. Based on this distinction, the research establishes the first provably valid baselines under Variant A for the medium- and large-scale instances of the problem.

As methodological contributions, two main axes are investigated. The first encompasses node clustering: formalizing the Greedy Angular Sectorization described ambiguously in the original model and proposing the Voronoi Partition with adaptive diversity mechanism, which demonstrated superiority in average *makespan* quality across 22 out of 23 evaluated instances. The second axis investigates problem decomposition into subinstances per depot across three operational modes: pure decomposition (executed independently), decomposition as a seed (initializing pheromone matrices in the complete space), and decomposition as refinement (post-global convergence polishing).

Experimental results confirm that the Voronoi Partition and decomposition as refinement provide significant improvements in *makespan* quality and computational performance. The work establishes a rigorous and reproducible methodological foundation for future research on the MDVRP-D.

Keywords: MDVRP-D, Drone Routing, Meta-heuristics, ACO, Voronoi Partition, Decomposition.

Lista de Figuras

1	Cadeia de suprimentos.	12
2	Linha do tempo dos problemas de roteirização.	14
3	Comparação entre as variantes de problemas de roteirização.	18
4	Comparação entre as rotas geradas pelo algoritmo de Clarke-Wright e pelo algoritmo de varredura.	20
5	Conceitos operacionais do MDVRP-D.	23
6	Minimização do makespan no MDVRP-D.	24
7	Dinâmica de sincronização caminhão-drone.	26
8	Fluxo de otimização do algoritmo AACO-NC-D.	34
9	Setorização Angular Gulosa na instância p01-D.	38
10	Partição de Voronoi por depósito na instância p01-D.	40
11	Listas de candidatos de um nó de borda na instância p01-D para diferentes valores de α : (a) $\alpha = 0$; (b) $\alpha = 0,5$; (c) $\alpha = 1,0$	42
12	Listas de candidatos de um nó de centro na instância p01-D: a estabilidade das listas demonstra que o mecanismo de diversidade não gera candidatos extras para nós distantes das fronteiras.	42
13	Exemplo de violação de autonomia na instância p20-D: (a) visão geral da rota; (b) detalhe da viagem do drone entre os nós 6, 20 e 22.	62

Lista de Tabelas

1	Complexidade das fases do AACO-NC-D	35
2	Parâmetros de controle do AACO-NC-D	37
3	Síntese da notação e variáveis dos algoritmos	37
4	Tamanho percentual do cluster por instância MDVRP-D	40
5	Comparação de complexidade: instância completa vs. decomposição	52
6	Especificações das instâncias de teste MDVRP-D	58
7	Parâmetros operacionais dos caminhões e drones	58
8	Parâmetros de controle do AACO-NC-D	59
9	Comparação dos ambientes computacionais	60
10	Revalidação por instância	63
11	Análise de sensibilidade da constante de diversidade α	65
12	Comparativo entre a Setorização Angular e a Partição de Voronoi com $\alpha = 1,0$	65
13	Comparativo entre AACO-NC-D sobre o espaço completo ($\alpha = 0$) e decomposição por sub-instâncias VRP-D.	67
14	Comparativo entre o AACO-NC-D sobre o espaço completo ($\alpha = 1,0$) e a Decomposição como Semente.	68
15	Comparativo entre o AACO-NC-D sobre o espaço completo ($\alpha = 1,0$) e a Decomposição como Refinamento.	68
16	Comparativo entre a Decomposição Pura e a Decomposição como Refinamento.	69
17	Variação entre depósitos e estratégia vencedora por instância.	70
18	Impacto da busca local na Decomposição Pura.	71

19	Impacto da busca local na Decomposição como Refinamento.	72
20	Comparativo entre MDVRP (sem drones) e MDVRP-D (com drones) sob Partição de Voronoi com $\alpha = 1,0$ e busca local.	73
21	Comparativo entre MDVRP (sem drones) e MDVRP-D (com drones) sob Decomposição Pura com busca local.	73
22	Comparativo entre MDVRP (sem drones) e MDVRP-D (com drones) sob Decomposição como Refinamento com busca local.	74
23	Impacto do relaxamento da autonomia do drone (Variante A vs. Variante B) na Partição de Voronoi com $\alpha = 1,0$	75
24	Impacto do relaxamento da autonomia do drone (Variante A vs. Variante B) na Decomposição Pura com busca local.	75
25	Impacto do relaxamento da autonomia do drone (Variante A vs. Variante B) na Decomposição como Refinamento com busca local.	76
26	Comparativo de desempenho sob a Variante A estrita para as instâncias p01-D a p07-D.	78
27	Comparativo direto sob a Variante B entre a abordagem proposta e os melhores resultados publicados por Stodola e Kutěj (2024)	79
28	Primeiras soluções válidas sob a Variante A para as instâncias p08-D a p23-D e custo da viabilidade em relação à Variante B do próprio trabalho.	80

Lista de Abreviaturas e Siglas

AACO-NC-D Adaptive Ant Colony Optimization with Node Clustering for Drones

ACO Ant Colony Optimization

ALNS Adaptive Large Neighborhood Search

B2B Business to Business

B2C Business to Customer

FSTSP Flying Sidekick Travelling Salesman Problem

GRASP Greedy Randomized Adaptive Search Procedure

MDVRP Multi-Depot Vehicle Routing Problem

MDVRP-D Multi-Depot Vehicle Routing Problem with Drones

OpenMP Open Multi-Processing

PDSTSP Parallel Drone Scheduling Travelling Salesman Problem

PO Pesquisa Operacional

SA Simulated Annealing

TSP Travelling Salesman Problem

TSP-D Travelling Salesman Problem with Drone

VANT Veículo Aéreo Não Tripulado

VND Variable Neighborhood Descent

VNS Variable Neighborhood Search

VRP Vehicle Routing Problem

VRP-D Vehicle Routing Problem with Drones

Sumário

1	Introdução	12
1.1	A última milha	12
1.2	Os problemas de roteirização	13
1.3	Proposta da dissertação	15
1.4	Organização do trabalho	16
2	Fundamentação teórica	17
2.1	O problema TSP	17
2.2	O problema VRP	18
2.3	O problema MDVRP	19
2.4	O problema VRP-D	21
2.5	O problema MDVRP-D	22
2.5.1	Definições e modelo	22
2.5.2	Equações e restrições	24
2.6	Otimização por colônia de formigas	27
3	Revisão da literatura	28
3.1	Uso de drones em problemas de otimização combinatória	28
3.2	Algoritmos para problemas de roteirização com drones	29
3.2.1	Variantes e dinâmicas recentes de roteamento com drones	30
3.2.2	Abordagens ACO para o VRP-D	31
3.3	O estado da arte: AACO-NC-D para o MDVRP-D	32

4	Metodologia	34
4.1	O algoritmo AACO-NC-D	34
4.1.1	Fundamentação do ACO adaptativo e parâmetros de controle . . .	36
4.1.2	Pré-processamento: clusterização de nós	36
4.1.2.1	Setorização angular gulosa	38
4.1.2.2	Partição de Voronoi com vizinhança adaptativa	39
4.1.3	Construção de solução por uma formiga	43
4.1.4	Feromônio e evaporação adaptativa	46
4.1.5	Busca local	48
4.2	Decomposição do problema	51
4.3	Estratégias de integração da decomposição	53
4.3.1	Decomposição como semente	53
4.3.2	Decomposição como refinamento	55
4.4	Propostas de estudo de ablação	55
5	Experimentos computacionais	57
5.1	Configuração experimental e instâncias	57
5.1.1	Métricas de avaliação e ambiente computacional	59
5.1.2	Análise crítica das restrições físico-operacionais das instâncias . . .	60
5.2	Revalidação computacional das soluções publicadas	61
5.2.1	Análise por perfil de instância	63
5.3	Estudo de ablação dos componentes	64
5.3.1	Impacto da estratégia de clusterização	64
5.3.2	Impacto da decomposição	66
5.3.2.1	Decomposição Pura	66
5.3.2.2	Decomposição como Semente	67
5.3.2.3	Decomposição como Refinamento	68

5.3.2.4	Síntese: Decomposição Pura vs. Refinamento	69
5.3.3	Impacto da otimização local	70
5.4	Impacto dos drones na roteirização multi-depósito	72
5.5	Sensibilidade à restrição de autonomia dos drones	74
5.6	Discussão dos resultados	77
5.6.1	Comparação de desempenho e custo da viabilidade	77
5.6.1.1	Comparativo sob a Variante A (estrita)	77
5.6.1.2	Comparativo direto sob a Variante B (relaxada)	78
5.6.1.3	Primeiras soluções válidas sob a Variante A	78
5.6.2	Relação entre <i>makespan</i> e distância total no MDVRP-D	79
5.6.3	Síntese das contribuições	80
6	Conclusão	82
6.1	Contribuições principais	82
6.2	Limitações	83
6.3	Trabalhos futuros	84
	REFERÊNCIAS	85

1 Introdução

1.1 A última milha

A partir do início dos anos 2000, ocorreu um crescimento acelerado do comércio eletrônico (*e-commerce*), o que transformou o panorama do varejo global. Impulsionada por gigantes como a Amazon (com forte presença na América do Norte e Europa), o Alibaba (líder na Ásia) e o Mercado Livre (dominante na América Latina), essa transformação digital alterou não apenas como os consumidores compram, mas, principalmente, como esperam receber seus produtos. Conforme apontam [Lim, Jin e Srai \(2018\)](#), a expectativa por entregas no mesmo dia (*same-day delivery*) ou no dia seguinte (*next-day delivery*) deixou de ser um diferencial competitivo para se tornar uma exigência básica do consumidor. Essa pressão logística converge para um ponto crucial da cadeia de suprimentos (*supply chain*) ilustrada na Figura 1: a última milha (*last mile*).

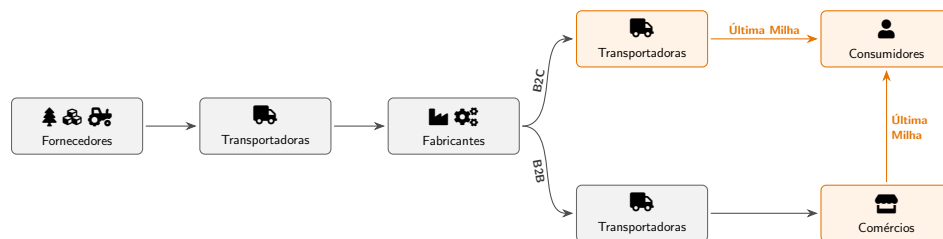


Figura 1: Cadeia de suprimentos.

A última milha refere-se à etapa final do processo de entrega de mercadorias: o trajeto do centro de distribuição local ou loja até a porta do consumidor final (B2C, *Business to Customer*). Este é o segmento mais ineficiente de toda a cadeia logística, podendo representar 50% dos custos logísticos totais ([MANGIARACINA et al., 2019](#)).

Diversos fatores contribuem para essa ineficiência econômica, como informado por [Mangiaracina et al. \(2019\)](#). Diferente do transporte de carga (B2B, *Business to Business*), que movimenta grandes volumes entre poucos pontos, a última milha é caracterizada por uma baixa densidade de paradas, com pequenos volumes sendo entregues em

vários endereços residenciais normalmente dispersos. Além disso, operações em centros urbanos densos sofrem com congestionamentos, restrições de tráfego, dificuldade de estacionamento e falhas na entrega (quando o cliente não está em casa, por exemplo), o que exige novas tentativas e eleva ainda mais o custo operacional.

Neste ambiente de alta complexidade, a eficiência logística é o motor de lucratividade do varejo digital. A redução do tempo de entrega impacta diretamente os custos operacionais, acionando o chamado Efeito de Alavancagem do Lucro (*Profit Leverage Effect*), onde pequenas economias logísticas geram impactos financeiros massivos no lucro líquido (BALLOU, 2004; TOTH; VIGO, 2014). Dessa forma, a Pesquisa Operacional (PO) estabelece-se como fundamental, oferecendo os modelos e algoritmos que permitem organizar e tornar esse problema complexo em algo matematicamente solucionável.

1.2 Os problemas de roteirização

O pilar fundamental de toda a otimização de rotas é o Problema do Caixeiro-Viajante (TSP, *Traveling Salesman Problem*). O TSP estabelece o desafio conceitual de encontrar a rota mais curta para um viajante visitar um conjunto de cidades e retornar à origem (DANTZIG; FULKERSON; JOHNSON, 1954).

O Problema de Roteirização de Veículos (VRP, *Vehicle Routing Problem*), que é o ponto de partida central para este campo de pesquisa, é a generalização mais famosa do TSP. Sua formulação original por Dantzig e Ramser (1959) expandiu o desafio de um único agente para uma frota de veículos com capacidade limitada, formalizando o desafio de criar rotas ótimas a partir de um único depósito.

No entanto, o modelo VRP clássico mostrou-se limitado para operações de larga escala, que dependem de uma logística descentralizada. Essa necessidade levou ao Problema de Roteirização de Veículos Multi-Depósito (MDVRP, *Multi-Depot Vehicle Routing Problem*). Este problema, introduzido em trabalhos como o de Gillett e Miller (1974), adicionou uma nova camada de complexidade: além da roteirização, era preciso definir a alocação de clientes, ou seja, qual depósito atenderia qual cliente.

Paralelamente a essa expansão de escala, o cenário da última milha foi transformado por uma inovação tecnológica: o uso de Veículos Aéreos Não Tripulados (VANTs), ou drones. Embora os drones ofereçam vantagens claras de velocidade, suas limitações de autonomia e capacidade levaram a modelos colaborativos.

O trabalho fundador desta área, o Problema do Carteiro Voador (FSTSP, *The Flying Sidekick Traveling Salesman Problem*) introduzido por [Murray e Chu \(2015\)](#), estabeleceu as bases para a classe de problemas conhecida como Problema de Roteirização de Veículos com Drones (VRP-D, *Vehicle Routing Problem with Drones*). Isso foi feito a partir de uma estratégia onde caminhões e drones trabalham em paralelo para realizar entregas, demonstrando ganhos significativos de tempo.

Com a evolução da literatura, apresentam-se dois eixos de complexidade: a escala operacional referente ao MDVRP e a inovação tecnológica relacionada ao VRP-D. A convergência natural desses dois eixos, conforme demonstrada na Figura 2, leva ao Problema de Roteirização de Veículos Multi-Depósito com Drones (MDVRP-D, *Multi-Depot Vehicle Routing Problem with Drones*), que é o problema-foco desta dissertação.

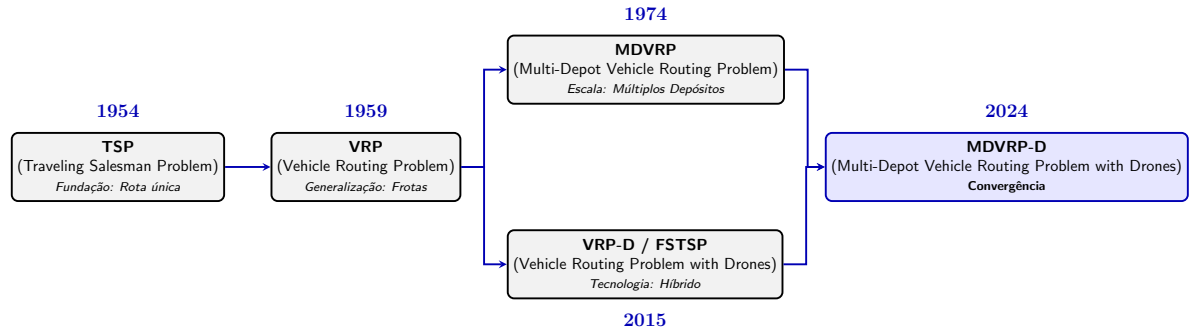


Figura 2: Linha do tempo dos problemas de roteirização.

Este problema modela o cenário operacional complexo de empresas que possuem múltiplos centros de distribuição e buscam implementar frotas híbridas. Ele combina os desafios de alocação depósito-cliente do MDVRP com os complexos desafios de coordenação e sincronização das frotas híbridas do VRP-D. Dentre estes desafios, destaca-se a divergência de objetivos entre a minimização do tempo de conclusão da rota mais longa (*makespan*) e a minimização da distância total percorrida. Enquanto o primeiro busca o equilíbrio de carga de trabalho entre os depósitos e a rapidez de atendimento global, o segundo prioriza a economia no consumo agregado de combustível terrestre e de energia aérea. Esses objetivos conflitantes geram *trade-offs* estruturais relevantes para a tomada de decisão logística.

Sendo o MDVRP-D um problema de introdução recente na literatura, o tema ainda não é tão explorado. O trabalho original de [Stodola e Kutěj \(2024\)](#) foi responsável por formalizar esta variante na literatura, propondo um modelo baseado em Otimização por Colônia de Formigas (ACO, *Ant Colony Optimization*). Por representar o início do estado da arte, este trabalho serve como o modelo-base para a análise de replicação e o estudo

de ablação conduzido nesta dissertação, conforme detalhado no Capítulo 4.

1.3 Proposta da dissertação

Apesar dos resultados promissores apresentados por [Stodola e Kutěj \(2024\)](#), sua proposta compartilha de um desafio crônico na área de meta-heurísticas: a replicabilidade. Artigos científicos, por limitação de espaço e foco, frequentemente omitem detalhes de implementação, estratégias de calibração de parâmetros ou otimizações de código de baixo nível que são extremamente importantes para o desempenho final do algoritmo.

Uma análise preliminar e uma tentativa de replicação do modelo AACO-NC-D (*Adaptive Ant Colony Optimization with Node Clustering for Drones*) revelaram que a descrição original possuía ambiguidades na especificação de seus componentes-chave e omitia o tratamento do tempo de serviço do caminhão no cálculo da autonomia do drone. Essa lacuna descritiva levou à identificação de duas variantes operacionais para a restrição de autonomia: a Variante A, que contabiliza o tempo de serviço do caminhão no ponto de recuperação como consumo de bateria do drone (correspondente à formalização matemática original), e a Variante B, que não o faz. A auditoria computacional revelou que as soluções publicadas por [Stodola e Kutěj \(2024\)](#) são aderentes à Variante B. Diante disso, esta dissertação tem como objetivo conduzir uma auditoria rigorosa e estabelecer os primeiros *baselines* de comparação válidos sob a Variante A para as instâncias de médio e grande porte do problema.

O objetivo central deste trabalho não é propor uma meta-heurística inteiramente nova, mas realizar um estudo de replicação e o estudo de ablação do modelo AACO-NC-D. O trabalho inicia com a replicação do modelo-base, documentando as discrepâncias encontradas em relação aos resultados da literatura. Na sequência, propõem-se estudos de ablação em componentes-chave do algoritmo, a fim de avaliar o impacto de cada um na qualidade das soluções obtidas.

As investigações propostas abrangem duas frentes principais. A primeira avalia a estratégia de clusterização de nós: a Partição de Voronoi por depósito, com vizinhança adaptativa controlada pelo parâmetro α , é proposta e comparada com a Setorização Angular Gulosa, analisando seu impacto na cobertura de fronteiras entre depósitos e na qualidade das rotas. Como observação metodológica associada, aborda-se a ambiguidade terminológica sobre o *k-opt* modificado presente na literatura de referência.

A segunda frente propõe e avalia três modos de uso da decomposição do problema

em sub-problemas VRP-D independentes: a decomposição pura, a inicialização guiada por decomposição (em que a solução decomposta semeia as matrizes de feromônio do AACO-NC-D global) e o refinamento por decomposição (em que o AACO-NC-D global é executado primeiro para identificar a atribuição de clientes a depósitos, e a decomposição é então aplicada como etapa de refinamento sobre essa estrutura fixada).

Com essas investigações, esta dissertação estabelece novos patamares de comparação para o MDVRP-D, estabelecendo os primeiros *baselines* para a Variante A e resultados competitivos sob ambas as variantes operacionais, e propondo uma metodologia algorítmica mais robusta e transparente. A metodologia adotada para conduzir essa investigação é detalhada no Capítulo 4.

1.4 Organização do trabalho

Para atingir os objetivos propostos, este documento está organizado da seguinte maneira. O Capítulo 2 expõe conceitos fundamentais para os problemas de roteirização e a formulação matemática completa do MDVRP-D. O Capítulo 3 apresenta uma revisão aprofundada da literatura, cobrindo as principais abordagens de solução para o VRP-D e o MDVRP-D. O Capítulo 4 descreve a metodologia de replicação do modelo AACO-NC-D e detalha as adaptações implementadas. O Capítulo 5 apresenta e discute os resultados computacionais obtidos, incluindo comparações com o algoritmo de referência. Finalmente, o Capítulo 6 apresenta as conclusões desta dissertação, suas limitações e aponta direções para trabalhos futuros.

2 Fundamentação teórica

Este capítulo estabelece os fundamentos teóricos necessários para a compreensão do problema central desta dissertação. A otimização de rotas é um campo construído ao longo de décadas de pesquisa, em que cada novo problema costuma ser uma generalização ou extensão de seu predecessor.

Para compreender o problema central, é essencial primeiro analisar seus componentes fundamentais. Inicia-se com a definição do problema-raiz de todas as roteirizações, o TSP. Em seguida, apresenta-se sua generalização mais famosa, o VRP, e sua extensão para múltiplos depósitos, o MDVRP.

Por fim, detalha-se a inovação tecnológica da frota híbrida, formalizada pelo FSTSP (VRP-D), e conclui-se com a definição formal e matemática do MDVRP-D. O capítulo encerra com a apresentação da Otimização por Colônia de Formigas (ACO), a meta-heurística que fundamenta o método proposto por [Stodola e Kutěj \(2024\)](#) e estudado neste trabalho.

2.1 O problema TSP

As raízes do problema datam do século XIX, com os estudos de Hamilton e Kirkman sobre ciclos hamiltonianos em grafos. A forma geral de otimização para minimizar a distância total foi estudada pela primeira vez na década de 1930 por Karl Menger, que considerou o uso de força bruta e observou a não-otimalidade da heurística do vizinho mais próximo. A resolução computacional desses problemas concretizou-se com os métodos de [Dantzig, Fulkerson e Johnson \(1954\)](#).

O TSP pode ser definido da seguinte forma: dado um conjunto de cidades que o caixeiro-viajante deve visitar, e conhecendo o custo de viagem (distância, tempo ou dinheiro) entre cada par de cidades, o objetivo é encontrar a rota de custo mínimo que visite cada cidade exatamente uma vez e retorne à cidade de origem, formando um ciclo

Hamiltoniano.

Embora conceitualmente simples, o TSP é um problema de otimização combinatória NP-difícil, servindo como a base para a maioria dos problemas de roteirização.

2.2 O problema VRP

O VRP é uma generalização direta do TSP. A formulação original é atribuída a [Dantzig e Ramser \(1959\)](#), que abordaram um problema real da indústria petrolífera: a distribuição eficiente de gasolina para postos de serviço.

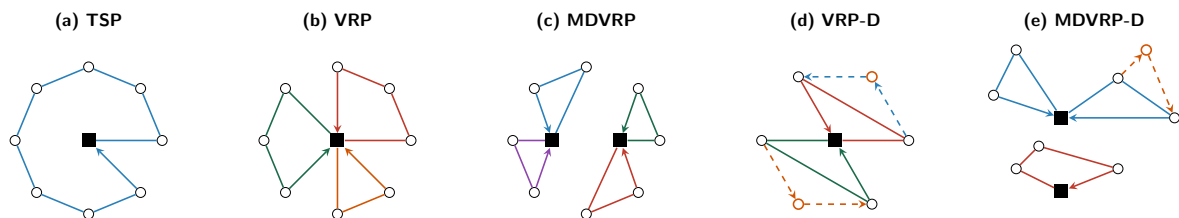


Figura 3: Comparação entre as variantes de problemas de roteirização.

O VRP difere do TSP em aspectos cruciais. Em vez de um único viajante, o VRP considera um depósito central, uma frota de veículos com capacidade de carga limitada e um conjunto de clientes com demandas específicas de mercadorias.

Visualmente, essa diferença é intuitiva ao se comparar as soluções: o TSP forma um grande ciclo único, enquanto o VRP assemelha-se a um conjunto de pétalas que convergem para o depósito. Conforme ilustrado nos painéis (a) e (b) da Figura 3 para um cenário de 8 clientes, o TSP pressupõe que um único veículo possui capacidade para visitar todos os pontos em uma única viagem. Já no VRP, as restrições de capacidade tornam o ciclo único inviável, exigindo a criação de múltiplas rotas que saem e retornam ao centro de distribuição.

O objetivo do VRP é, portanto, determinar um conjunto de rotas de custo mínimo (usualmente distância ou tempo) para a frota de veículos, de modo que todos os clientes sejam atendidos, as restrições de capacidade dos veículos sejam respeitadas e cada rota inicie e termine no depósito central.

Logo após a formulação original, [Clarke e Wright \(1964\)](#) propuseram o algoritmo de economias (*savings algorithm*), mais conhecido como algoritmo de Clarke-Wright, que transformou o VRP de um problema teórico intratável em uma aplicação prática viável.

Para isso, foi introduzido o conceito heurístico de economia. Considerando que cada cliente é atendido por um caminhão exclusivo que vai e volta do depósito, calcula-se a economia de unir dois clientes em uma mesma rota, dado que uma viagem de ida e uma de volta ao depósito não serão mais necessárias. O algoritmo calcula essa economia para todos os pares e prioriza a fusão daqueles com maior ganho, respeitando a capacidade de carga dos veículos.

Com isso, o algoritmo permitiu a resolução de instâncias com centenas de clientes, demandando baixo custo computacional e qualidade de solução aceitável. Dessa forma, empresas de logística passaram a utilizar computadores para planejar rotas do dia a dia e, por conta dessa visibilidade, o VRP consolidou-se como um campo central da PO.

2.3 O problema MDVRP

O modelo VRP clássico, com sua premissa de um único depósito, mostrou-se limitado para representar operações logísticas de larga escala, como as encontradas em grandes centros urbanos, onde a realidade logística é descentralizada. Tal limitação impulsionou o desenvolvimento do MDVRP, que introduz uma complexidade adicional: antes da roteirização propriamente dita, é imperativo resolver a alocação dos clientes aos múltiplos depósitos. O painel (c) da Figura 3 ilustra essa configuração: dois depósitos independentes, cada um responsável por atender um subconjunto geograficamente próximo de clientes.

Para lidar com esse desafio, a literatura consagrou estratégias de decomposição conhecidas como *Cluster-First, Route-Second* (que envolvem agrupar primeiro, para roteirizar depois). O precursor fundamental dessa abordagem é o algoritmo de varredura (*sweep algorithm*), proposto originalmente por Gillett e Miller (1974).

Enquanto Clarke e Wright (1964) focaram na construção de rotas por meio da fusão de trajetos baseada em custos, Gillett e Miller (1974) abriram o caminho para o uso da estratégia de decomposição geométrica. A importância histórica do algoritmo de varredura reside em dois pilares: a aceitação visual da solução obtida e a eficiência computacional.

A aceitação visual refere-se a uma grande vantagem do método: a geração de rotas que, devido à lógica angular, tendem a não se cruzar. Para os despachantes da época, essas soluções pareciam mais lógicas e organizadas do que as geradas pelo algoritmo de Clarke-Wright, que por vezes criavam emaranhados visuais, ainda que fossem matematicamente eficientes. Essa diferença é ilustrada na Figura 4.

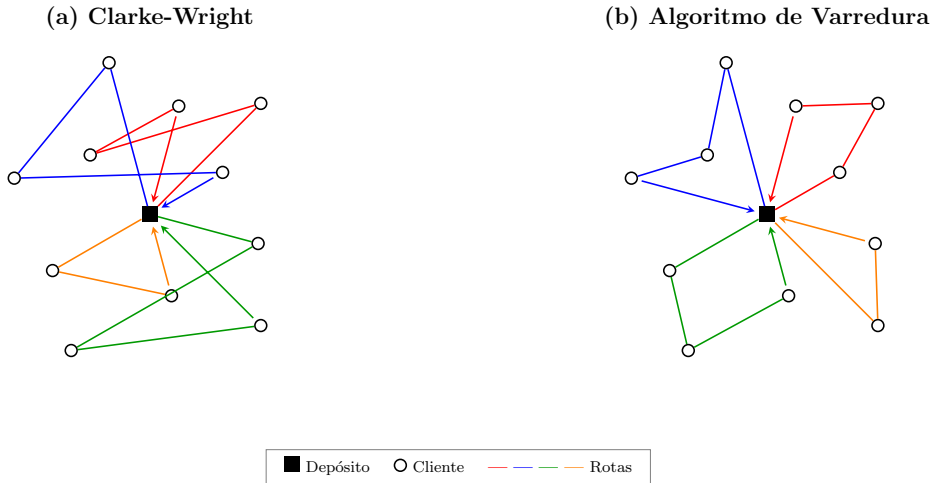


Figura 4: Comparação entre as rotas geradas pelo algoritmo de Clarke-Wright e pelo algoritmo de varredura.

A eficiência computacional, por sua vez, refere-se à utilização de coordenadas polares para segmentar o problema em sub-regiões independentes, permitindo o tratamento de instâncias com milhares de nós com um custo computacional extremamente baixo de $O(n \log n)$ (TOTH; VIGO, 2014), pois sua complexidade depende basicamente da ordenação dos ângulos. Esse valor é ainda mais impressionante quando comparado com a complexidade de $O(n^2 \log n)$ (TOTH; VIGO, 2014) do algoritmo de Clarke-Wright.

O funcionamento do algoritmo baseia-se na transformação do problema 2D (coordenadas cartesianas) em uma lista 1D ordenada, convertendo a localização dos clientes em coordenadas polares e usando o ângulo em relação ao depósito como chave de ordenação. Simulando um radar girando em torno do depósito, varre-se a área angularmente (geralmente no sentido anti-horário). À medida que o raio avança, os clientes são agrupados em clusters até que a capacidade do veículo seja preenchida, momento em que um novo grupo é iniciado.

Embora concebido inicialmente para o VRP, essa lógica de agrupamento geométrico tornou-se a base para resolver a etapa de alocação do MDVRP. Trabalhos pioneiros, como a extensão proposta por Gillett e Johnson (1976), adaptaram o algoritmo decompondo o MDVRP em duas etapas. Primeiramente, os clientes são alocados aos depósitos baseando-se na proximidade euclidiana, e o MDVRP é então quebrado em vários VRPs independentes, permitindo que o algoritmo de varredura seja aplicado individualmente para cada depósito.

Essa contribuição foi crucial para formalizar a dualidade do problema, que exige a resolução simultânea da alocação (determinar qual depósito atende um conjunto de clien-

tes aglomerados geometricamente) e da roteirização (definir a sequência ótima de visita dentro de cada grupo).

No cenário brasileiro, destacam-se trabalhos importantes produzidos no Instituto de Computação da Universidade Federal Fluminense (IC-UFF). A tese de [Subramanian \(2012\)](#) e a tese de [Penna \(2013\)](#), ambas orientadas por Luiz Satoru Ochi, estabeleceram resultados que superaram o estado da arte para o MDVRP e mais de 25 variantes do VRP à época.

2.4 O problema VRP-D

O avanço tecnológico comercial das últimas décadas viabilizou a introdução de Veículos Aéreos Não Tripulados (VANTs), comumente referidos como drones, na logística civil. A utilização de drones para entregas promete ganhos significativos de agilidade, especialmente por sua capacidade de operar sem as restrições da malha viária terrestre e de evitar congestionamentos urbanos. Entretanto, os drones comercialmente viáveis possuem limitações severas de capacidade de carga unitária e autonomia de bateria.

Para viabilizar essas operações, a literatura propôs a utilização colaborativa entre veículos terrestres e aéreos. O trabalho fundador dessa abordagem foi o *Flying Sidekick Traveling Salesman Problem* (FSTSP), introduzido por [Murray e Chu \(2015\)](#). Nesse paradigma, o caminhão funciona não apenas para entregas convencionais, mas também atua como um depósito móvel e plataforma de lançamento e recuperação para o drone.

Essa sincronização possibilita explorar o que há de melhor em cada veículo: o caminhão possui grande capacidade de carga e alcance, enquanto o drone possui alta velocidade e capacidade de realizar entregas em paralelo ao percurso terrestre. A dinâmica de uma operação típica no VRP-D, denominada viagem do drone, envolve o lançamento do drone pelo caminhão em um nó específico (que pode ser o depósito ou a localização de um cliente). No FSTSP clássico, essa operação é definida por um voo de três vértices: o local de partida, o local de entrega e o local de retorno, que deve ser um vértice subsequente na rota do caminhão.

O painel (d) da Figura 3 ilustra essa dinâmica: o caminhão percorre rotas de entrega (linhas sólidas), enquanto o drone é lançado em nós intermediários para atender clientes adicionais por caminhos aéreos (linhas tracejadas).

Apesar da vantagem operacional, o VRP-D eleva consideravelmente a dificuldade do

modelo matemático. A roteirização passa a exigir, além do trajeto ótimo para o veículo, a sincronização rigorosa no fluxo de tempo e espaço entre o caminhão e o drone. As restrições do modelo precisam assegurar que o limite de tempo de voo da bateria não seja excedido, contemplam tempos técnicos para acoplamento e desacoplamento, coordenam o tempo de espera do caminhão no vértice de recuperação e garantem que o drone transporte cargas compatíveis. É importante notar que o tempo de espera no local de retorno, caso o drone chegue antes do caminhão, é considerado tempo de voo e consome bateria. Além disso, variações posteriores, como o *Traveling Salesman Problem with Drones* (TSP-D) proposto por [Agatz, Bouman e Schmidt \(2018\)](#), expandiram o modelo permitindo que o caminhão permaneça estacionado enquanto aguarda o drone.

2.5 O problema MDVRP-D

A convergência natural dos dois eixos de desenvolvimento, a descentralização operacional do MDVRP e a inovação tecnológica das frotas híbridas do VRP-D, resulta no Problema de Roteirização de Veículos Multi-Depósito com Drones (MDVRP-D, *Multi-Depot Vehicle Routing Problem with Drones*), ilustrado no painel (e) da Figura 3. Esse modelo integra, em uma única formulação matemática, as dificuldades simultâneas de seus predecessores: o desafio de alocação de múltiplos depósitos e os desafios de coordenação e sincronização de frota híbrida.

Tratando-se de uma variação NP-difícil de introdução recente na literatura de PO, o MDVRP-D tem seu embasamento central nesta pesquisa direcionado pelo estudo de [Stodola e Kutěj \(2024\)](#). Esta seção apresenta a formulação matemática completa da variante abordada neste trabalho.

2.5.1 Definições e modelo

Seja G um grafo cujos vértices compreendem todos os depósitos e clientes: $V = D \cup C$, onde $D = \{D_1, D_2, \dots, D_m\}$ é o conjunto de depósitos, $C = \{C_1, C_2, \dots, C_n\}$ o conjunto de clientes e m, n seus tamanhos. Associados a D , há um conjunto de caminhões $T = \{T_1, T_2, \dots, T_m\}$ e um conjunto de drones $U = \{U_1, U_2, \dots, U_m\}$. Cada caminhão $T_i \in T$ é equipado com um drone $U_i \in U$ e está localizado em seu respectivo depósito $D_i \in D$, cuja posição 2D é denotada por D_i^y . Cada cliente C_i possui localização C_i^y , demanda C_i^{demand} e deve ser atendido exatamente uma vez.

Os caminhões são caracterizados por quatro parâmetros: velocidade média de deslo-

camento $T_i^{velocity}$, capacidade máxima de carga $T_i^{capacity}$, duração máxima de uma viagem $T_i^{maxroute}$ e tempo de serviço ao cliente $T_i^{service}$. Os drones, por sua vez, são definidos por seis parâmetros: velocidade média de voo $U_i^{velocity}$, capacidade máxima de carga por viagem do drone $U_i^{capacity}$, autonomia máxima de voo por viagem do drone $U_i^{endurance}$, tempo de serviço ao cliente $U_i^{service}$, tempo de preparação e lançamento U_i^{launch} e tempo de recuperação e acoplamento $U_i^{recovery}$.

Quando o cliente C_i é atendido por um drone, este deve ser lançado a partir de um vértice $V_a \in V$ e recuperado pelo caminhão em um vértice $V_b \in V$. Define-se uma viagem do drone como a tripla $P_i = \langle V_a, V_c, V_b \rangle$, onde V_a é o vértice de lançamento, $V_c \in C$ é o cliente atendido pelo drone e V_b é o vértice de encontro, com $V_a \neq V_b$, $V_a \neq V_c$ e $V_b \neq V_c$.

Uma solução do MDVRP-D é representada pelo par (R, S) , em que R e S denotam, respectivamente, o conjunto de rotas dos caminhões e o de viagens dos drones. Para alinhar formalmente o trabalho com a literatura de roteirização, adotam-se três definições conceituais básicas.

A primeira delas é a *rota*, que corresponde ao caminho completo de um caminhão T_i ao longo de sua jornada de entregas, denotada por $R^i = (r_1^i, r_2^i, \dots, r_{n_i}^i)$. A segunda é a *viagem*, que define cada ciclo individual de saída e retorno do caminhão ao depósito, em que a rota do veículo é composta por uma ou mais viagens consecutivas, respeitando as restrições de capacidade e duração de cada viagem. Por fim, a *viagem do drone* representa o trajeto aéreo a partir de seu lançamento pelo caminhão, passando pelo cliente atendido e retornando para sua posterior recuperação.

A Figura 5 ilustra os conceitos operacionais do MDVRP-D. Nela, observa-se a rota completa de um caminhão composta por duas viagens terrestres independentes (que iniciam e terminam no depósito), além de uma viagem de drone executada em paralelo a um dos trajetos terrestres.

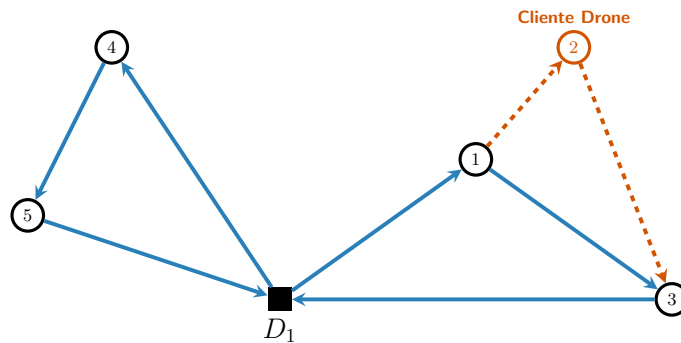


Figura 5: Conceitos operacionais do MDVRP-D.

A sequência de viagens do drone U_i é denotada por $S^i = (p_1^i, p_2^i, \dots, p_{m_i}^i)$, com $m_i \geq 0$. Cada viagem do drone $p_j^i = (l_j^i, s_j^i, f_j^i)$ é composta pelo vértice de lançamento $l_j^i \in R^i$, pelo cliente atendido $s_j^i \in C$ (com $s_j^i \notin R^i$, ou seja, o cliente do drone não pode constar na rota terrestre) e pelo vértice de recuperação $f_j^i \in R^i$, com o lançamento precedendo a recuperação na sequência de R^i . Define-se $B^i = \{f_1^i, f_2^i, \dots, f_{m_i}^i\}$ como o conjunto de todos os vértices de recuperação das viagens do drone U_i .

A função objetivo do MDVRP-D é a minimização do tempo máximo de conclusão entre todos os caminhões, denominado *makespan*, formalizado como a busca por minimizar $f(R, S) = \max(t_{n_1}^1, t_{n_2}^2, \dots, t_{n_m}^m)$, onde $t_{n_i}^i$ é o tempo em que o caminhão T_i (e seu drone) terminam a operação no último vértice $r_{n_i}^i$ (o depósito). Este objetivo visa distribuir uniformemente os clientes entre as frotas dos diferentes depósitos, tratando-as como recursos paralelos, conforme ilustrado na Figura 6.

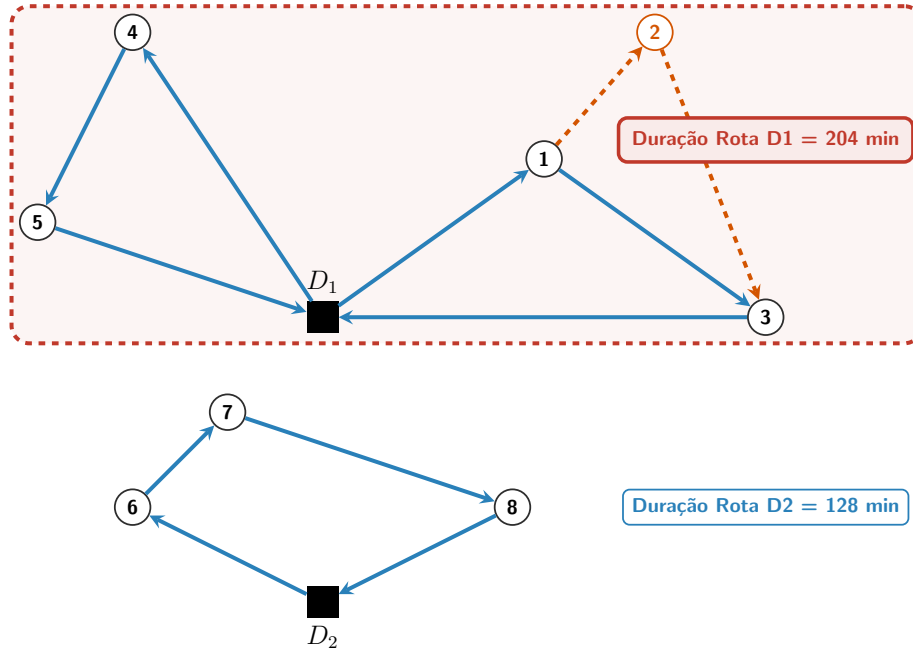


Figura 6: Minimiza  o do makespan no MDVRP-D.

2.5.2 Equa  es e restri  es

As equa  es de tempo e sincroniza  o a seguir s o reproduzidas de (STODOLA; KUT J, 2024) e fundamentam a avalia  o de solu  es vi veis do MDVRP-D.

O tempo de deslocamento do caminh o ao v rtice r_j^i , incluindo o atendimento ao cliente,   dado por:

$$d_{r_j^i}^i = \frac{|r_j^i - r_{j-1}^i|}{T_i^{velocity}} + \begin{cases} 0 & \text{se } r_j^i \in D \\ T_i^{service} & \text{se } r_j^i \in C \end{cases} \quad (2.1)$$

O tempo de duração de cada viagem do drone p_k^i , desde o lançamento até a recuperação, é:

$$d_{p_k^i}^i = U_i^{launch} + \frac{|s_k^i - r_{a_k^i}^i|}{U_i^{velocity}} + U_i^{service} + \frac{|r_{b_k^i}^i - s_k^i|}{U_i^{velocity}} + U_i^{recovery} \quad (2.2)$$

Os tempos de partida dos caminhões em cada vértice da rota são calculados pelas equações de sincronização abaixo. O tempo de partida do primeiro vértice (o depósito) é zero para todos os caminhões:

$$t_1^i = 0 \quad \text{para todo } T_i \in T \quad (2.3)$$

Para os demais vértices, o tempo de partida considera se o caminhão deve aguardar o retorno do drone, conforme a Equação 2.4. Nessa equação, $t_{b_k^i}^i = t_{a_k^i}^i + d_{p_k^i}^i$ é o instante em que o drone está pronto para ser recuperado. A sincronização garante que o caminhão aguarda o drone quando necessário (e vice-versa). O conceito de encontro sincronizado no vértice de resgate é ilustrado na Figura 7. O drone realiza a viagem aérea enquanto o caminhão segue pelo percurso terrestre atendendo clientes intermediários; o tempo de espera do drone em voo estacionário no nó de encontro é contabilizado para o limite de autonomia da bateria.

$$t_j^i = \begin{cases} t_{j-1}^i + d_{r_j^i}^i & \text{se } r_j^i \notin B^i \\ \max(t_{j-1}^i + d_{r_j^i}^i + U_i^{launch} + U_i^{recovery}, t_{b_k^i}^i) & \text{caso contrário } (b_k^i = j) \end{cases} \quad (2.4)$$

Conforme definido por Stodola e Kutěj (2024), qualquer solução viável (R, S) deve satisfazer as restrições operacionais do problema. A restrição de cobertura exige que todos os clientes sejam atendidos: $\bigcup_{i=1}^m ((R^i \cap C) \cup (S^i \cap C)) = C$.

Adicionalmente, a restrição de unicidade garante que cada cliente seja atendido exatamente uma vez, de forma que o cliente de uma viagem do drone não pode constar na rota terrestre do caminhão ($s_j^i \notin R^i$). A exclusividade de depósito define que cada caminhão T_i visita apenas seu próprio depósito D_i , enquanto a restrição de início e fim da viagem

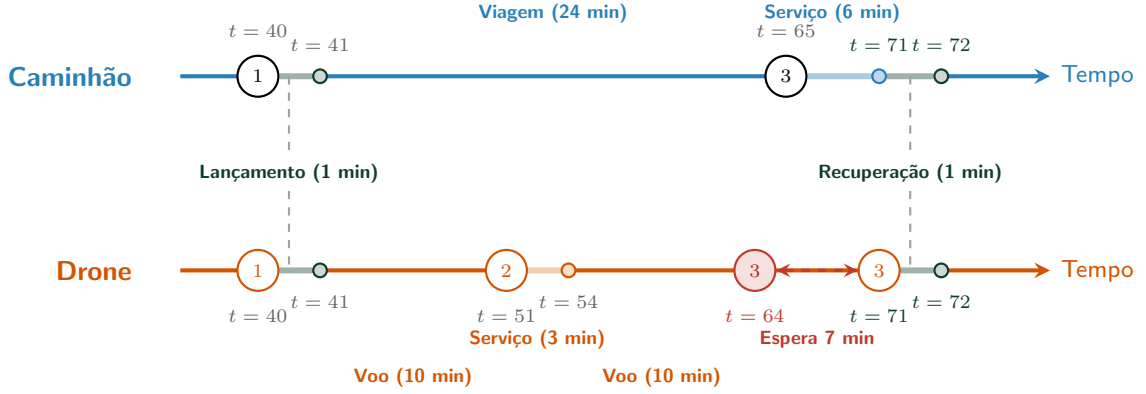


Figura 7: Dinâmica de sincronização caminhão-drone.

estabelece que $r_1^i = r_{n_i}^i = D_i$ para todo T_i .

Quanto à coordenação dos veículos, a sequência de viagens do drone determina que os vértices de lançamento e recuperação de cada viagem do drone devem aparecer nessa ordem na rota do caminhão, com o vértice de recuperação de uma viagem p_j^i precedendo obrigatoriamente o lançamento da viagem subsequente p_{j+1}^i , garantindo a recuperação do drone antes de um novo lançamento.

A restrição de duração de viagem impõe que a duração de cada viagem individual do caminhão não ultrapasse o limite $T_i^{maxroute}$. No que se refere à capacidade, a capacidade do caminhão restringe a soma das demandas de todos os clientes atendidos em uma viagem (tanto de forma terrestre quanto aérea pelo drone associado) ao limite $T_i^{capacity}$. Já a capacidade do drone limita a demanda do cliente atendido em uma viagem aérea a $U_i^{capacity}$. Por fim, a restrição de autonomia do drone impede que a duração de sua viagem aérea exceda a autonomia $U_i^{endurance}$ da bateria.

A sincronização no vértice de recuperação determina a duração efetiva da viagem do drone. Na formalização matemática proposta por Stodola e Kutěj (2024), a autonomia da bateria contabiliza o voo e o tempo de espera no ponto de resgate (incluindo o tempo de serviço do caminhão no nó de encontro), conforme a Equação 2.5, onde t_{launch}^i é o instante de partida do caminhão do nó de lançamento V_a e t_j^i é o instante de partida do nó de resgate V_b , integrando $T_i^{service}$ via Equação 2.4. Conforme discutido nos Capítulos 4 e 5, essa contabilização diferencia o modelo estrito (Variante A) do modelo relaxado (Variante B).

$$t_j^i - t_{launch}^i \leq U_i^{endurance} \quad (2.5)$$

2.6 Otimização por colônia de formigas

A Otimização por Colônia de Formigas (ACO, *Ant Colony Optimization*), introduzida por [Dorigo, Maniezzo e Colormi \(1996\)](#), é uma meta-heurística populacional de busca estocástica inspirada no comportamento de formigas reais na busca por alimento. Formigas depositam feromônio no solo à medida que se deslocam. Ao se depararem com bifurcações, as formigas escolhem caminhos probabilisticamente, influenciadas pela intensidade do feromônio ali depositado. Como caminhos mais curtos são percorridos mais rapidamente, o fluxo de retorno de formigas por esses trajetos é maior, resultando em um acúmulo acelerado de feromônio que atrai novas formigas em um ciclo de realimentação positiva.

O ACO simula esse comportamento para resolver problemas de otimização combinatória. Uma população de formigas artificiais constrói soluções incrementalmente, adicionando um elemento por vez ao percurso em construção. A cada passo, a formiga k localizada no nó i seleciona o próximo nó j a visitar dentro do seu conjunto de vizinhos viáveis N_i^k de acordo com a regra de transição probabilística descrita na Equação 2.6, onde τ_{ij} denota a intensidade da trilha de feromônio no arco (i, j) , $\eta_{ij} = 1/d_{ij}$ representa a informação heurística (normalmente, o inverso da distância), α^{ACO} pondera a importância relativa do feromônio e β regula a influência da informação heurística.

$$p_{ij}^k = \frac{[\tau_{ij}]^{\alpha^{ACO}} \cdot [\eta_{ij}]^{\beta}}{\sum_{l \in N_i^k} [\tau_{il}]^{\alpha^{ACO}} \cdot [\eta_{il}]^{\beta}} \quad \text{se } j \in N_i^k \quad (2.6)$$

Após todas as formigas concluírem a construção de suas soluções, a matriz de feromônio é atualizada pela combinação dos processos de evaporação e reforço, conforme a Equação 2.7, onde $\rho \in (0, 1]$ é a taxa de evaporação e $\Delta\tau_{ij}^k$ representa a quantidade de feromônio depositada pela formiga k no arco (i, j) , sendo proporcional à qualidade da solução produzida por essa formiga (ou nula se o arco não foi utilizado na solução).

$$\tau_{ij} \leftarrow (1 - \rho) \cdot \tau_{ij} + \sum_{k=1}^{n_{ants}} \Delta\tau_{ij}^k \quad (2.7)$$

Esse ciclo de construção e atualização é repetido até que um critério de parada seja atingido. O equilíbrio entre evaporação e reforço é o que confere ao ACO sua capacidade de explorar o espaço de busca de forma ampla nas fases iniciais e de convergir progressivamente para soluções de alta qualidade. Essa estrutura geral é o ponto de partida sobre o qual o AACO-NC-D, descrito no Capítulo 4, constrói suas adaptações específicas para o MDVRP-D.

3 Revisão da literatura

O Capítulo 2 estabeleceu a formulação teórica dos problemas centrais de roteirização, desde o VRP clássico até o MDVRP-D. Uma vez que todos esses problemas são classificados como NP-Difíceis, sua solução exata é computacionalmente inviável para instâncias de tamanho realista. Por essa razão, a literatura científica tem se concentrado massivamente no desenvolvimento de heurísticas e meta-heurísticas. Este capítulo analisa o estado da arte das abordagens de solução propostas para os problemas abordados nesta dissertação.

3.1 Uso de drones em problemas de otimização combinatória

Partindo dos modelos de roteirização estabelecidos no Capítulo 2, a ideia de utilizar drones em cooperação com veículos terrestres surgiu no trabalho pioneiro de [Murray e Chu \(2015\)](#). Os autores definiram duas variantes: o PDSTSP (*Parallel Drone Scheduling TSP*), no qual drones partem diretamente do depósito, e o FSTSP (*Flying Sidekick TSP*), no qual o drone é lançado pelo caminhão durante a rota e recuperado em um vértice subsequente. Esta última abordagem, por estender o alcance efetivo do drone sem exigir retorno ao depósito, tornou-se o paradigma dominante na literatura subsequente.

Diversas extensões do modelo original foram propostas; [Agatz, Bouman e Schmidt \(2018\)](#) introduziram o TSP-D (*Traveling Salesman Problem with Drones*), permitindo que o caminhão aguarde o drone no mesmo cliente de lançamento e considerando uma velocidade do drone como múltiplo da velocidade do caminhão. Em seguida, [Ha et al. \(2018\)](#) propuseram uma variante cujo objetivo é minimizar o custo operacional total, composto pelo custo de transporte e pelo custo de tempo de espera entre os veículos.

A transição do cenário de um único caminhão para frotas ocorreu no estudo de [Wang, Poikonen e Golden \(2017\)](#), que introduziram o VRP-D com o objetivo de minimizar a duração máxima das rotas, demonstrando teoricamente as economias obtidas pelo uso de drones; [Poikonen, Wang e Golden \(2017\)](#) estenderam o problema incorporando a limitação de bateria dos drones e duas matrizes de distâncias distintas: uma para caminhões e outra

para drones. Uma formulação matemática detalhada do VRP-D, com análise de restrições de capacidade e duração máxima de rota, foi proposta por [Sacramento, Pisinger e Ropke \(2019\)](#). Uma revisão abrangente dos problemas de roteirização com drones pode ser encontrada em [Liang e Luo \(2022\)](#).

Todos os estudos mencionados consideram frotas partindo de um único depósito. Até onde se tem conhecimento, nenhum trabalho anterior ao de [Stodola e Kutěj \(2024\)](#) abordou o cenário em que caminhões equipados com drones estão distribuídos em múltiplos depósitos, configurando o MDVRP-D.

No contexto de roteamento de VANTs, destacam-se os trabalhos de [Junior et al. \(2020\)](#), [Júnior, Coelho et al. \(2022\)](#) e [Júnior, Ochi et al. \(2023\)](#), que propõem metaheurísticas multiobjetivo para o Problema de Roteamento de Grade de VANTs Verdes (*Green UAV Grid Routing Problem*), com aplicações em cidades inteligentes. Esses trabalhos demonstram a aplicabilidade de abordagens metaheurísticas ao roteamento de VANTs em cenários de escala real, contribuindo para a literatura de referência da área.

3.2 Algoritmos para problemas de roteirização com drones

A otimização combinatória tem sido objeto de intensa pesquisa ao longo das últimas décadas, e uma vasta gama de algoritmos determinísticos e estocásticos foi proposta para o TSP, VRP, MDVRP e suas variantes. Por se tratarem de problemas NP-difíceis, abordagens heurísticas e meta-heurísticas são amplamente adotadas ([HALIM; ISMAIL, 2019](#)). Entre as abordagens aplicadas especificamente ao roteamento de VANTs, [Junior et al. \(2020\)](#), [Júnior, Coelho et al. \(2022\)](#) e [Júnior, Ochi et al. \(2023\)](#) empregam metaheurísticas multiobjetivo combinadas com estratégias de busca local, obtendo resultados competitivos em instâncias de larga escala.

Para o FSTSP, destaca-se o trabalho de [Ponza \(2016\)](#), que desenvolveu um algoritmo baseado no princípio do *Simulated Annealing*. A Busca em Vizinhança Adaptativa Grande (ALNS, *Adaptive Large Neighborhood Search*) foi aplicada ao FSTSP por [Windras Mara, Rifai e Sopha \(2022\)](#). Uma Busca em Vizinhança Variável (VNS, *Variable Neighborhood Search*) foi proposta por [Freitas e Penna \(2019\)](#), utilizando a solução ótima do TSP como ponto de partida. Um algoritmo de *branch-and-bound* para pequenas instâncias (até 15 clientes) foi desenvolvido por [Dell’Amico, Montemanni e Novellani \(2021\)](#).

No VRP-D, [Euchi e Sadok \(2021\)](#) propuseram um algoritmo genético híbrido combinado com técnicas de varredura. O ALNS foi proposto por [Sacramento, Pisinger e Ropke](#)

(2019) como solução para o VRP-D, e seus resultados tornaram-se referência na área; Tamke e Buscher (2021) desenvolveram um algoritmo de *branch-and-cut*. Por sua vez, Schermer, Moeini e Wendt (2019) introduziram um algoritmo híbrido VNS/Busca Tabu para a variante VRPDERO, na qual drones podem ser lançados e recuperados em pontos intermediários.

3.2.1 Variantes e dinâmicas recentes de roteamento com drones

A pesquisa sobre o roteamento cooperativo entre caminhões e drones tem explorado novas dinâmicas de operação, que ampliam os modelos clássicos do FSTSP e do VRP-D, conforme sintetizado na revisão da literatura de Liang e Luo (2022). A seguir serão descritas as principais variações observadas na literatura recente.

A primeira delas é a sincronização em movimento (POIKONEN; WANG; GOLDEN, 2017). Nessa dinâmica, o caminhão não precisa parar para lançar ou recolher o drone. A operação é feita enquanto o veículo está em deslocamento na malha rodoviária. A razão para isso é que a eliminação das paradas reduz o tempo técnico de espera, o que tende a minimizar o *makespan* global da operação. Por outro lado, essa abordagem exige um controle físico de voo mais complexo. No contexto nacional, Costa (2023) propuseram um Algoritmo Memético combinado com a heurística Lin-Kernighan-Helsgott para um problema de entrega fracionada com caminhões atuando como plataformas móveis de lançamento e recuperação de drones, modelado sob as premissas de Cidades Inteligentes.

A segunda variante é a sincronização contínua, também conhecida como sincronização em arcos (SCHERMER; MOEINI; WENDT, 2019). Enquanto a sincronização discreta limita os eventos de lançamento e recuperação aos vértices dos clientes, a sincronização contínua permite que tais eventos ocorram em qualquer ponto ao longo de uma via. Com isso, o sistema ganha flexibilidade geométrica para encontrar pontos de encontro que minimizam o consumo de bateria do drone.

Outro paradigma relevante é o de modelos com drones desacoplados (MURRAY; CHU, 2015). Nesses sistemas, os drones operam de forma independente e partem diretamente de depósitos ou de micro-hubs urbanos para realizar entregas de ida e volta (como no modelo PDSTSP), sem utilizar o caminhão como base móvel de reabastecimento.

Por fim, destacam-se as propostas de terceirização logística híbrida, baseadas em *crowdsourcing* (MANGIARACINA et al., 2019). Esse conceito refere-se ao aproveitamento de uma rede sob demanda de entregadores autônomos ou prestadores terceirizados

para realizar entregas de última milha. No contexto de VANTs, drones autônomos ou operadores independentes são integrados dinamicamente à rede de distribuição para absorver picos de demanda. Conforme apontado na revisão de inovações de última milha por [Mangiaracina et al. \(2019\)](#), essa estratégia evita o custo fixo de manutenção de frotas subutilizadas fora dos horários de pico.

Uma dimensão adicional na classificação dessas variantes é a topologia da rota aérea e o grau de autonomia do drone. Do ponto de vista topológico, as soluções da literatura organizam-se em duas famílias: a topologia em estrela, na qual o drone parte e retorna ao depósito de forma independente (modelo PDSTSP), e a topologia em ciclo, na qual o drone é lançado e recuperado pelo caminhão em vértices distintos da sua rota, formando um ciclo tripártite (modelo FSTSP e suas variantes, incluindo o MDVRP-D). Do ponto de vista operacional, os VANTs empregados na logística de última milha são normalmente classificados em autônomos (operam por planejamento de voo pré-programado, sem piloto humano em tempo real) e pilotados remotamente (requerem operação ativa por um operador). Essa distinção tem impacto direto nos modelos de roteirização: drones autônomos admitem planejamento de rota integrado à otimização, enquanto drones pilotados introduzem variáveis humanas que limitam a previsibilidade do tempo de resposta e aumentam o custo operacional.

Sendo assim, embora essas variantes representem o estado da arte na logística de *last mile* ([LIANG; LUO, 2022](#)), elas exigem modelos matemáticos substancialmente mais complexos. Dito isso, este trabalho segue focado no modelo MDVRP-D de [Stodola e Kutěj \(2024\)](#), que utiliza a sincronização discreta nos vértices com parada obrigatória do caminhão.

3.2.2 Abordagens ACO para o VRP-D

O ACO não tem sido extensivamente utilizado em problemas combinatórios com drones em cooperação com caminhões; [Huang et al. \(2022\)](#) propuseram um algoritmo ACO para o VRP-D, no qual as transições entre clientes seguem o princípio ACO padrão, mas o drone é lançado com uma probabilidade constante definida como parâmetro de controle, ou seja, a decisão de lançamento não é influenciada pelas trilhas de feromônio. Além disso, uma única matriz de feromônio é usada tanto para caminhões quanto para drones, o que ignora as diferentes características de cada veículo.

Outros trabalhos utilizam matrizes de feromônio distintas para caminhões e drones ([DAS et al., 2021](#)), mas geram as rotas em fases sequenciais desacopladas, limitando

a integração das decisões de lançamento.

Dessa forma, a revisão da literatura revelou que o princípio de lançamento de drones guiado pela intensidade das trilhas de feromônio, desenvolvido no algoritmo AACO-NC-D de [Stodola e Kutěj \(2024\)](#), não havia sido publicado anteriormente. Além disso, não há registro na literatura de um algoritmo ACO aplicado ao problema de roteirização com drones em múltiplos depósitos.

3.3 O estado da arte: AACO-NC-D para o MDVRP-D

O estado da arte do problema é o MDVRP-D, formalmente introduzido por [Stodola e Kutěj \(2024\)](#). Este trabalho diferenciou-se de seus predecessores ao utilizar sequências de vértices como variáveis de decisão, resultando em um modelo matemático mais conciso: 10 restrições, em contraste com as 30 do modelo VRP-D de [Sacramento, Pisinger e Ropke \(2019\)](#).

Para resolver o problema, foi proposto o algoritmo AACO-NC-D (*Adaptive Ant Colony Optimization with Node Clustering for Drones*), que incorpora um conjunto de inovações estruturais em relação ao ACO clássico. A primeira delas é a manutenção de matrizes de feromônio independentes para caminhões (τ) e drones (τ'), reconhecendo que as regras de rentabilidade diferem entre os veículos. Em segundo lugar, a decisão de lançamento do drone ocorre de forma probabilística via $p(C_{drone}) = 1 - e^{-\tau_{\max}}$, ajustando a taxa de voo estocasticamente conforme a força das trilhas acumuladas pela colônia.

Adicionalmente, adota-se a clusterização de nós (*node clustering*) para reduzir a complexidade computacional de construção de $O(n^2)$ para $O(n \cdot n_{prim} \cdot n_{size})$ por passo. Por fim, a evaporação é regulada adaptativamente pela entropia de Shannon, acelerando a convergência em populações diversas e preservando a estrutura em populações similares, integrada a uma busca local especializada equipada com operadores de troca entre rotas terrestres e viagens de drone.

Os experimentos computacionais de [Stodola e Kutěj \(2024\)](#) demonstraram que o AACO-NC-D supera o ALNS em 17 dos 28 grupos de instâncias VRP-D testados, com *gap* variando de -1% a $+3\%$. Nas instâncias MDVRP-D, o uso de drones proporcionou uma economia média de $24,7\%$ no tempo total de operação em comparação com o MDVRP puro, com impacto variando de $4,5\%$ a $49,5\%$ dependendo da configuração das instâncias. A busca local mostrou-se essencial: sem ela, o *gap* médio em relação ao melhor resultado cresce de $2,4\%$ para $10,2\%$.

Além do desempenho competitivo, o AACO-NC-D tem etapas claramente separadas: clusterização, construção por formiga e busca local, cada uma com interface bem definida. Essa separação é propícia para um estudo de ablação, em que o impacto de cada componente pode ser avaliado de forma isolada e o algoritmo pode ser estendido por adição de novos módulos, como a decomposição proposta nesta dissertação.

Após a introdução do MDVRP-D, a pesquisa ramificou-se para aplicações de nicho, como logística humanitária (YÜKSEL; ELIGÜZEL; METE, 2024) e resposta a emergências (WENG et al., 2024). Contudo, nenhum desses trabalhos subsequentes revisitou as restrições do modelo original nem conduziu uma avaliação sistemática dos componentes do algoritmo, o que motiva a proposta metodológica deste trabalho, apresentada no Capítulo 4.

4 Metodologia

Este capítulo descreve a metodologia utilizada para conduzir o estudo de replicação e o estudo de ablação do algoritmo AACO-NC-D (*Adaptive Ant Colony Optimization with Node Clustering for Drones*). A abordagem proposta fundamenta-se na auditoria rigorosa dos parâmetros operacionais da literatura, na investigação da clusterização de nós com sobreposição adaptativa e no desenvolvimento de estratégias de decomposição do problema.

4.1 O algoritmo AACO-NC-D

O AACO-NC-D é um algoritmo híbrido proposto por [Stodola e Kutěj \(2024\)](#) que combina pré-processamento de clusterização, uma meta-heurística ACO adaptativa e refinamento por busca local. A arquitetura geral e o fluxo de execução do modelo estão ilustrados na Figura 8, que demonstra as etapas sequenciais desde o processamento da instância até o refinamento das soluções.

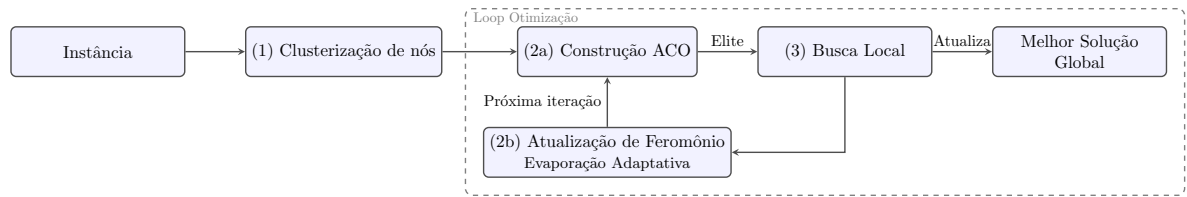


Figura 8: Fluxo de otimização do algoritmo AACO-NC-D.

A estrutura lógica e a sequência detalhada de operações do AACO-NC-D estão formalizadas no Algoritmo 1.

A Tabela 1 resume a complexidade das fases do AACO-NC-D. O pré-processamento de clusterização é executado uma única vez. A clusterização de nós reduz a complexidade por iteração de $O(n^2)$ para $O(n \cdot m \cdot n_{prim} \cdot n_{size})$, onde $m \cdot n_{prim} \cdot n_{size} \ll n$.

Os componentes do AACO-NC-D exercem papéis distintos no balanço entre diversi-

Algoritmo 1 AACO-NC-D

```

1: Inicializar  $\tau_{j,k}^i = \tau_{j,k}'^i = 1$  para todo  $V_j, V_k \in V, D_i \in D$ 
2: Para cada  $v \in V$ : construir lista de clusters  $K^{(v)}$ 
3:  $R, S \leftarrow$  construir solução inicial (1 formiga)
4: while critério de parada não atingido (iter / tempo / entropia / estagnação) do
5:   for  $a = 1$  to  $n_{ants}$  do
6:      $R_a, S_a \leftarrow$  AntSolution ( $V, D, C, T, U, K, \tau, \tau'$ )
7:   end for
8:   Validar soluções; acumular ocorrências de arestas para entropia  $H$ 
9:   Selecionar a melhor formiga por custo (formiga elite)
10:  if  $iter \bmod n_{freq} = 0$  then
11:     $R_{best}, S_{best} \leftarrow$  LocalOptimization ( $R_{best}, S_{best}$ )
12:  end if
13:  Atualizar  $R, S$  (aceitar eventual piora via SA global com temperatura  $T_{update}$ )
14:  Calcular  $H$ ; ajustar  $\rho \in [\rho_{min}, \rho_{max}]$  conforme Eq. 4.3
15:  Reforçar  $\tau$  e  $\tau'$  com base em  $R^+, S^+$ 
16:  Evaporar  $\tau$  e  $\tau'$  por fator  $(1 - \rho)$ 
17:   $iter \leftarrow iter + 1$ 
18: end while
19: return  $R, S$ 

```

Tabela 1: Complexidade das fases do AACO-NC-D

Fase	Complexidade
Clusterização	$O(n^2 \cdot n_{sect})$
Construção por formiga	$O(n \cdot m \cdot n_{prim} \cdot n_{size})$
Feromônio	$O(m \cdot n^2)$
Busca local	$O(n^2 \cdot k_{max})$
Loop global	$O(iter \cdot n_{ants} \cdot n \cdot m \cdot n_{prim} \cdot n_{size})$

ficação e intensificação, eixo central da eficácia de metaheurísticas ACO. A construção probabilística das soluções, conduzida pelos operadores de seleção de lista de vizinhos e de cliente (*SelectCluster* e *SelectCustomer*), utiliza as trilhas de feromônio τ para guiar estocasticamente as formigas em direção a arcos de alta qualidade descobertos em iterações anteriores, constituindo o mecanismo central do ACO.

A intensificação é realizada pela Busca Local (*LocalOptimization*), que explora de forma determinística a vizinhança do melhor indivíduo da geração, refinando a solução corrente. A diversificação, por sua vez, é promovida pela evaporação adaptativa, em que ρ é proporcional à entropia H , acelerando o esquecimento de rotas ruins em momentos de baixa diversidade, e pela aceitação de pioras controladas via parâmetro T_{update} , que impede a convergência prematura da busca. O equilíbrio entre esses mecanismos é regulado pelo critério de entropia mínima ($H < H_{thresh}$), que encerra a otimização quando a diver-

tidade cai abaixo de um limiar configurável, sinalizando que novas iterações dificilmente produzirão melhorias adicionais.

4.1.1 Fundamentação do ACO adaptativo e parâmetros de controle

Em contraste com o ACO clássico descrito na Seção 2.6, o AACO-NC-D de Stodola e Kutěj (2024) incorpora mecanismos adaptativos específicos para gerenciar a complexidade do MDVRP-D. O primeiro elemento diferenciador é a dualidade de trilhas de feromônio, na qual são mantidas duas matrizes independentes (τ para caminhões e τ' para drones), reconhecendo as dinâmicas operacionais distintas de cada frota. Além disso, a seleção do próximo cliente a ser atendido é restringida do conjunto de vizinhos viáveis da formiga para listas de vizinhos pré-calculadas de tamanho n_{size} , o que mitiga o custo combinatorial. Cada vértice possui um conjunto de listas de vizinhos, sendo as primeiras n_{prim} listas classificadas como primárias e consultadas na seleção probabilística do ACO; dessa forma, a formiga avalia no máximo $n_{prim} \cdot n_{size}$ candidatos ativos a cada passo de construção.

O lançamento de drones ocorre de forma dinâmica e probabilística via $p(C_{drone}) = 1 - e^{-\tau_{max}}$, que cresce conforme a colônia consolida trajetórias eficientes. Além disso, a taxa de evaporação ρ é regulada adaptativamente pela entropia de Shannon H , retardando a convergência prematura em momentos de estagnação. Por fim, a atualização de feromônio aceita soluções da iteração piores que a melhor global com probabilidade estocástica a temperatura constante $T_{update} = 0,1$ (Seção 4.1.4), atuando como um critério de aceitação no estilo Metropolis que permite à colônia escapar de ótimos locais.

O equilíbrio e a intensidade desses mecanismos são regulados pelo conjunto de parâmetros de controle detalhados na Tabela 2. Os valores numéricos empíricos desses parâmetros são apresentados na Seção 5.1. A notação algorítmica e as variáveis utilizadas na formalização dos procedimentos são sintetizadas na Tabela 3.

4.1.2 Pré-processamento: clusterização de nós

Sem restrição ao conjunto de candidatos, cada formiga avalia todos os n clientes não visitados a cada passo de construção, impondo custo $O(n^2)$ por geração, incompatível com as instâncias de larga escala do MDVRP-D. A clusterização de nós resolve esse gargalo pré-computando, para cada vértice, listas de vizinhos de tamanho fixo, reduzindo o custo de construção de $O(n^2)$ para $O(n \cdot m \cdot n_{prim} \cdot n_{size})$. Embora essa delimitação reduza o tempo computacional por iteração ao focar a busca em vizinhanças promissoras, listas

Tabela 2: Parâmetros de controle do AACO-NC-D

Parâmetro	Descrição
n_{ants}	Determina o tamanho da população de formigas por iteração.
n_{freq}	Intervalo de iterações para ativação da busca local.
n_{prim}	Quantidade de listas de vizinhos consideradas na seleção probabilística.
n_{size}	Número máximo de vértices candidatos contidos em cada lista de vizinhos.
n_{sect}	Divisões angulares usadas na Setorização Angular Gulosa.
T_{update}	Temperatura constante de aceitação de soluções de piora.
α_{update}	Taxa de decaimento de temperatura (fixada em 1,0).
ρ_{min}	Limite inferior da taxa de evaporação de feromônio.
ρ_{max}	Limite superior da taxa de evaporação de feromônio.
δ	Coefficiente multiplicativo para o reforço de trilha.
α^{ACO}	Peso da intensidade do feromônio na probabilidade.
β	Peso da informação heurística (visibilidade) na probabilidade.

Tabela 3: Síntese da notação e variáveis dos algoritmos

Variável	Descrição
V	Conjunto total de vértices da instância ($V = D \cup C$)
D	Conjunto de depósitos ($ D = m$)
C	Conjunto de clientes ($ C = n$)
R^i	Rota terrestre do caminhão associada ao depósito $i \in D$
S^i	Conjunto de viagens de drone associadas ao depósito $i \in D$
p_j^i	Viagens do drone expressas por (l_j^i, s_j^i, f_j^i)
l_j^i, f_j^i	Nó de lançamento e nó de recuperação da viagem do drone
s_j^i	Cliente atendido pela viagem do drone
τ, τ'	Matriz de feromônio para caminhões e para drones
$K^{(v_i)}$	Estrutura de listas de vizinhos associada ao vértice v_i
n_{total}	Número total de sub-listas de vizinhos ($n_{total} = K^{(v_i)} $)
$U_{endurance}$	Autonomia máxima de voo do drone (tempo de bateria)
$T^{service}$	Tempo de serviço/atendimento do caminhão no cliente
Q_{truck}, Q_{drone}	Capacidade máxima de carga do caminhão e do drone
α	Parâmetro de diversidade e sobreposição de fronteira

excessivamente rígidas correm o risco de excluir conexões de alta qualidade entre depósitos. A estratégia proposta neste trabalho incorpora sobreposição de fronteira controlada pelo parâmetro α (Equação 4.1) para mitigar esse risco.

Cabe destacar a relação entre os procedimentos de clusterização: a etapa *Voronoi-Clusters* (Algoritmo 3) substitui a etapa *CreateClusters* (Algoritmo 2) no fluxo geral do algoritmo AACO-NC-D (Algoritmo 1) quando a estratégia de Partição de Voronoi com Vizinhança Adaptativa (proposta original deste trabalho) é adotada no lugar da Setorização

Angular Gulosa da literatura.

Para cada vértice $v \in V$, é construída uma estrutura organizada em n_{total} listas de vizinhos de tamanho máximo n_{size} (onde $n_{total} = |K^{(v)}|$). As primeiras n_{prim} listas são denominadas primárias e são consultadas na seleção probabilística; as demais funcionam como reserva de contingência.

4.1.2.1 Setorização angular gulosa

Cabe observar que [Stodola e Kutěj \(2024\)](#) não especifica a heurística geométrica utilizada para delimitar os setores angulares: o artigo indica apenas que o vértice mais próximo em cada setor é selecionado, sem definir como os setores são particionados (ângulo fixo de $360^\circ/n_{sect}$, setores adaptativos à densidade local, ou outro critério). Esta omissão impede a replicação exata do mecanismo de setorização; neste trabalho, preenche-se essa lacuna definindo formalmente a partição uniforme de $360^\circ/n_{sect}$ graus por setor, centrada no vértice candidato e alinhada ao eixo horizontal. A etapa de pré-processamento *Create-Clusters* (Algoritmo 2) apresenta os detalhes operacionais da setorização angular proposta na literatura, em que cada sub-lista de tamanho fixo é populada sequencialmente a partir de varreduras direcionais. A Figura 9 ilustra o resultado dessa setorização na instância p01-D.

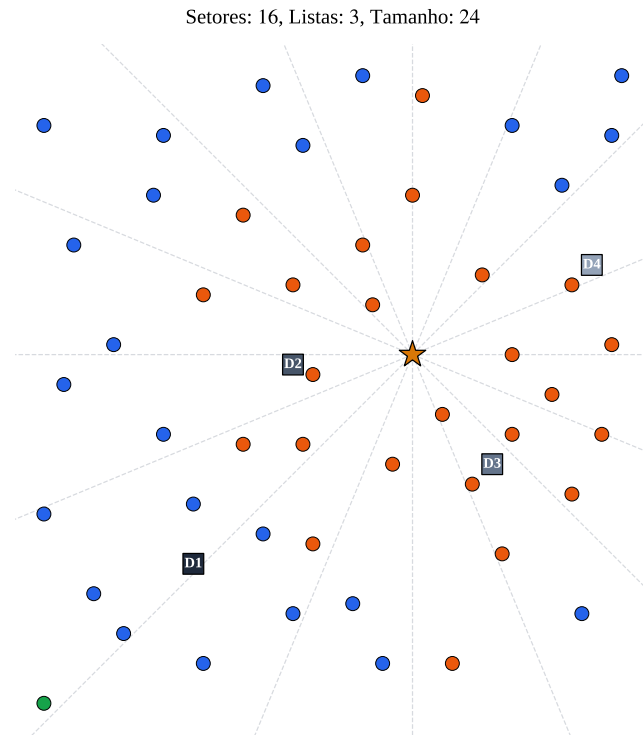


Figura 9: Setorização Angular Gulosa na instância p01-D.

Algoritmo 2 CreateClusters**Require:** $V, v_i, n_{size}, n_{sect}$ **Ensure:** $K^{(v_i)}$

```

1:  $id \leftarrow 1$ 
2:  $K_{id}^{(v_i)} \leftarrow \emptyset$ 
3:  $V_{free} \leftarrow V \setminus \{v_i\}$ 
4: for  $j = 1$  to  $n_{sect}$  do
5:   Encontrar o vértice  $v \in V_{free}$  mais próximo de  $v_i$  no setor  $j$ 
6:    $K_{id}^{(v_i)} \leftarrow K_{id}^{(v_i)} \cup \{v\}$ 
7:    $V_{free} \leftarrow V_{free} \setminus \{v\}$ 
8: end for
9: while  $V_{free} \neq \emptyset$  do
10:  if  $|K_{id}^{(v_i)}| \geq n_{size}$  then
11:     $id \leftarrow id + 1$ 
12:     $K_{id}^{(v_i)} \leftarrow \emptyset$ 
13:  end if
14:  Encontrar o vértice  $v \in V_{free}$  mais próximo de  $v_i$ 
15:   $K_{id}^{(v_i)} \leftarrow K_{id}^{(v_i)} \cup \{v\}$ 
16:   $V_{free} \leftarrow V_{free} \setminus \{v\}$ 
17: end while
18: return  $K^{(v_i)}$ 

```

4.1.2.2 Partição de Voronoi com vizinhança adaptativa

A Partição de Voronoi com Vizinhança Adaptativa é uma estratégia original desenvolvida neste trabalho para superar a rigidez e as limitações geométricas da setorização angular. Em vez de criar fatias angulares, a alocação dos clientes aos depósitos fundamenta-se no conceito de Partição de Voronoi por proximidade euclidiana pura, associando cada cliente ao depósito geograficamente mais próximo. A Figura 10 ilustra a partição espacial resultante na instância p01-D.

O comportamento do mecanismo de diversidade varia conforme a posição geográfica do cliente no cluster. Para um cliente situado no núcleo do cluster, os $P\%$ vizinhos mais próximos estarão contidos no mesmo território, de modo que o filtro resulte em poucos candidatos extras. Por outro lado, para um cliente localizado na fronteira (clientes cujas vizinhanças mais próximas cruzam as divisórias entre territórios), esses vizinhos mais próximos dividem-se entre clusters distintos, permitindo a inclusão de candidatos externos e ampliando a lista para cobrir rotas que cruzam fronteiras quando a diversidade é ativada ($\alpha > 0$).

A eficácia do percentual de diversidade P é sensível ao grau de fragmentação da instância. Em cenários com poucos depósitos (ex: $m = 2$, onde cada partição ocupa cerca

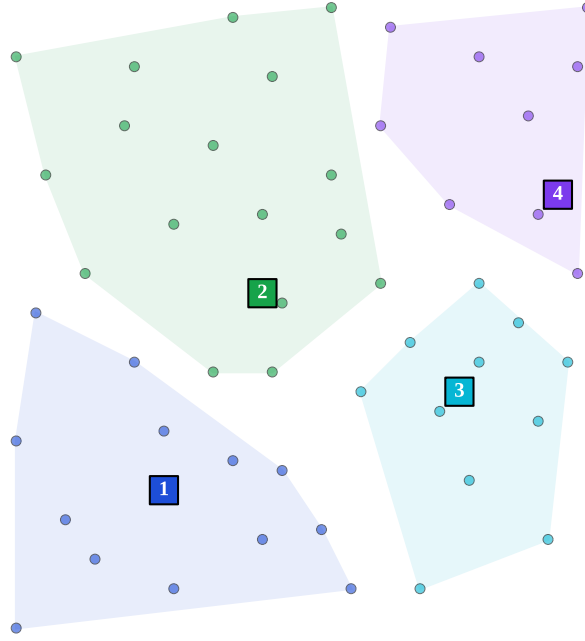


Figura 10: Partição de Voronoi por depósito na instância p01-D.

de 50% do mapa), um percentual fixo de $P = 10\%$ pode ser insuficiente para transpor as fronteiras geográficas e capturar vizinhos externos. Inversamente, em instâncias com muitos depósitos (ex: $m = 9$), esse mesmo valor pode abranger a totalidade de um cluster adjacente, tornando a busca excessivamente ampla. A Tabela 4 ilustra essa variação: para cada grupo de instâncias, o tamanho médio do cluster (n/m) e a fração correspondente ($1/m$) definem o “raio natural” do território de cada depósito.

Tabela 4: Tamanho percentual do cluster por instância MDVRP-D

Instância	Clientes (n)	Depósitos (m)	n/m	Tamanho do cluster ($1/m$)
p01-D e p02-D	50	4	12,5	25,0%
p03-D	75	5	15,0	20,0%
p04-D e p05-D	100	2	50,0	50,0%
p06-D	100	3	33,3	33,3%
p07-D	100	4	25,0	25,0%
p08-D	249	2	124,5	50,0%
p09-D	249	3	83,0	33,3%
p10-D	249	4	62,2	25,0%
p11-D	249	5	49,8	20,0%
p12-D a p14-D	80	2	40,0	50,0%
p15-D a p17-D	160	4	40,0	25,0%
p18-D a p20-D	240	6	40,0	16,7%
p21-D a p23-D	360	9	40,0	11,1%

Para garantir uma cobertura de fronteira consistente independentemente do número de depósitos, adota-se a Hipótese da Vizinhança Adaptativa: o parâmetro P é escalonado proporcionalmente ao tamanho médio de um cluster, controlado por uma constante α , conforme a Equação 4.1, onde α é uma constante configurável. Para $\alpha = 0,5$, o raio de cobertura equivale à metade do tamanho médio de um cluster; para $\alpha = 1,0$, equivale a um cluster inteiro, garantindo cobertura consistente de fronteira independentemente do número de depósitos. A etapa de pré-processamento *VoronoiClusters* (Algoritmo 3) formaliza a Partição de Voronoi com Vizinhança Adaptativa.

$$P = \alpha \cdot \frac{100\%}{m} \quad (4.1)$$

Algoritmo 3 VoronoiClusters

Require: $V, D, C, n_{size}, n_{prim}, \alpha$

Ensure: $K^{(v)}$ para cada $v \in V$

```

  {Etapa 1 - Partição de Voronoi por depósito mais próximo}
  1: for cada  $c \in C$  do
  2:    $region(c) \leftarrow \arg \min_{D_i \in D} \text{dist}(c, D_i)$ 
  3: end for
  {Etapa 2 - Parâmetro de diversidade adaptativo}
  4:  $n \leftarrow |C|$ ;  $m \leftarrow |D|$ 
  5:  $k_{cross} \leftarrow \lfloor \alpha \cdot n/m \rfloor$ 
  {Etapas 3 e 4 - Lista de candidatos para cada vértice}
  6: for cada  $v \in V$  do
  7:    $r_v \leftarrow \arg \min_{D_i \in D} \text{dist}(v, D_i)$ 
  8:    $C_{same} \leftarrow \{c \in C \mid region(c) = r_v, c \neq v\}$ 
  9:   Ordenar  $C_{ext} = \{c \in C \mid region(c) \neq r_v\}$  por  $\text{dist}(v, c)$  crescente
  10:   $C_{cross} \leftarrow$  primeiros  $k_{cross}$  elementos de  $C_{ext}$ 
  11:   $V_{cand} \leftarrow C_{same} \cup C_{cross}$ 
  12:  Ordenar  $V_{cand}$  por  $\text{dist}(v, c)$  crescente
  13:   $id \leftarrow 1$ ;  $K_{id}^{(v)} \leftarrow \emptyset$ 
  14:  for cada  $c \in V_{cand}$  (em ordem crescente de distância) do
  15:    if  $|K_{id}^{(v)}| \geq n_{size}$  then
  16:       $id \leftarrow id + 1$ ;  $K_{id}^{(v)} \leftarrow \emptyset$ 
  17:    end if
  18:     $K_{id}^{(v)} \leftarrow K_{id}^{(v)} \cup \{c\}$ 
  19:  end for
  20:   $K^{(v)} \leftarrow (K_1^{(v)}, K_2^{(v)}, \dots)$ 
  21: end for
  22: return  $K^{(v)}$  para todo  $v \in V$ 

```

A Figura 11 ilustra o efeito do parâmetro α sobre as listas de candidatos de um nó de borda na instância p01-D, comparando três configurações: $\alpha = 0$ (sem diversidade),

$\alpha = 0,5$ (raio de cobertura equivalente a metade do cluster) e $\alpha = 1,0$ (raio equivalente a um cluster inteiro).

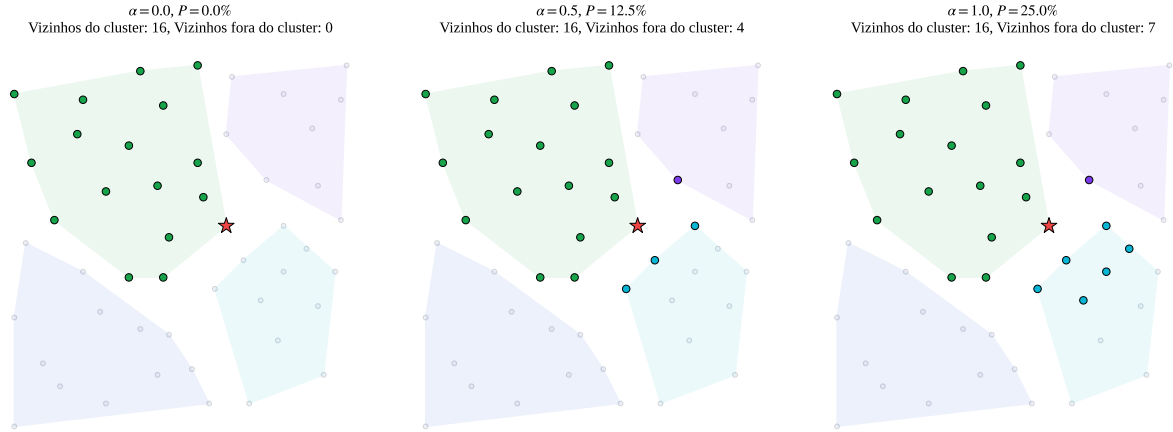


Figura 11: Listas de candidatos de um nó de borda na instância p01-D para diferentes valores de α : (a) $\alpha = 0$; (b) $\alpha = 0,5$; (c) $\alpha = 1,0$.

Em contraste, a Figura 12 apresenta as listas de candidatos para um nó de centro, localizado no interior geográfico de seu cluster. Por estar afastado das divisórias entre depósitos, a vizinhança do top $P\%$ para este nó permanece integralmente contida dentro dos limites de seu próprio território, resultando em listas estáveis independentemente do aumento do parâmetro α .

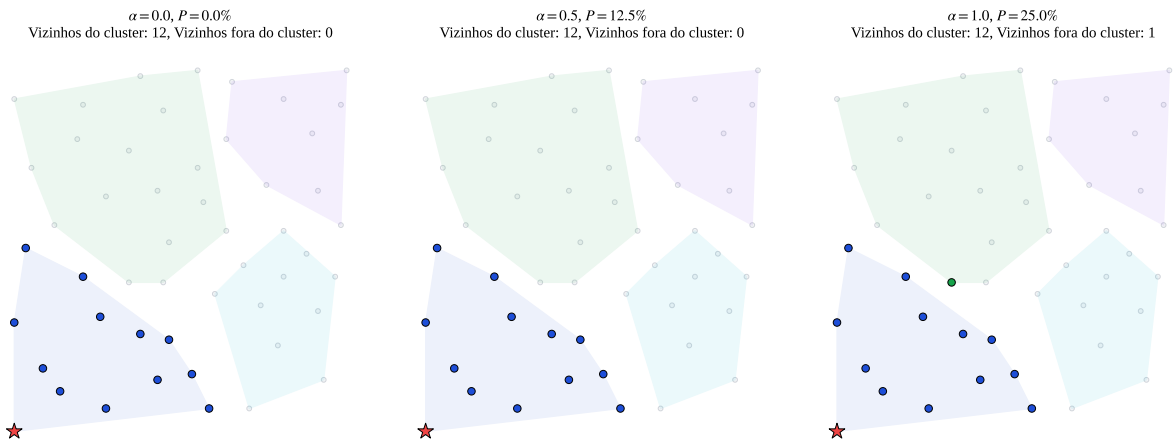


Figura 12: Listas de candidatos de um nó de centro na instância p01-D: a estabilidade das listas demonstra que o mecanismo de diversidade não gera candidatos extras para nós distantes das fronteiras.

As vantagens da Partição de Voronoi sobre a Setorização Angular Gulosa (Figura 9) decorrem dessas propriedades estruturais. Enquanto a setorização angular gera fatias radiais rígidas a partir de cada nó, correndo o risco de ignorar vértices espacialmente próximos situados em setores adjacentes ou em depósitos vizinhos, a Partição de Voro-

noi (Figura 10) estabelece fronteiras naturais alinhadas à proximidade dos depósitos. A inclusão controlada de vizinhos de fronteira pelo parâmetro α (Figuras 11 e 12) permite transpor dinamicamente as barreiras entre clusters para os nós das bordas, sem comprometer a compacidade das listas para os nós do interior.

4.1.3 Construção de solução por uma formiga

O método *AntSolution* (Algoritmo 4) é o responsável por construir soluções para cada formiga a cada iteração do ACO. Esse método distribui os clientes não visitados nas rotas terrestres de caminhão e nas viagens aéreas de drone, buscando um balanceamento operacional de carga.

Algoritmo 4 AntSolution

```

1:  $V_{free} \leftarrow C$ 
2: for cada  $D_i \in D$  do
3:    $R^i \leftarrow \{D_i\}$ ;  $S^i \leftarrow \emptyset$ ;  $R_{last}^i \leftarrow D_i$ ;  $q_{route}^i \leftarrow 0$ ;  $t_{route}^i \leftarrow 0$ 
4: end for
5: while  $V_{free} \neq \emptyset$  do
6:    $D_i \leftarrow \text{SelectDepot}(\tau)$ 
7:    $K_j^{(R_{last}^i)} \leftarrow \text{SelectCluster}(K^{(R_{last}^i)}, \tau)$ 
8:    $V_{candidates} \leftarrow V_{free} \cap K_j^{(R_{last}^i)}$ 
9:    $C_{truck} \leftarrow \text{SelectCustomer}(V_{candidates}, \tau, \alpha, \beta)$ 
10:  if  $q_{route}^i + C_{truck}^{demand} > T_i^{capacity}$  or  $t_{route}^i + d_{C_{truck}}^i + d_{D_i}^i > T_i^{maxroute}$  then
11:     $R^i \leftarrow R^i + \{D_i\}$ ;  $q_{route}^i \leftarrow 0$ ;  $t_{route}^i \leftarrow 0$ 
12:  end if
13:   $R^i \leftarrow R^i + \{C_{truck}\}$ ;  $q_{route}^i \leftarrow q_{route}^i + C_{truck}^{demand}$ ;  $V_{free} \leftarrow V_{free} \setminus \{C_{truck}\}$ 
14:   $C_{drone} \leftarrow \text{SelectDrone}(D_i, T_i, U_i, R_{last}^i, C_{truck}, V_{free}, \tau')$ 
15:  if  $C_{drone} \neq \emptyset$  then
16:     $p \leftarrow (R_{last}^i, C_{drone}, C_{truck})$ ;  $S^i \leftarrow S^i + \{p\}$ 
17:     $q_{route}^i \leftarrow q_{route}^i + C_{drone}^{demand}$ ;  $V_{free} \leftarrow V_{free} \setminus \{C_{drone}\}$ 
18:     $t_{route}^i \leftarrow t_{route}^i + \max(d_{C_{truck}}^i + U_i^{launch} + U_i^{recovery}, d_p^i)$ 
19:  else
20:     $t_{route}^i \leftarrow t_{route}^i + d_{C_{truck}}^i$ 
21:  end if
22:   $R_{last}^i \leftarrow C_{truck}$ 
23: end while
24: for cada  $D_i \in D$  do
25:    $R^i \leftarrow R^i + \{D_i\}$ 
26: end for
27: return  $R_a = \{R^1, \dots, R^m\}$ ,  $S_a = \{S^1, \dots, S^m\}$ 

```

A seleção de cliente de caminhão segue a regra probabilística ACO padrão, ponderando a intensidade do feromônio e a informação heurística, conforme a Equação 4.2. O

parâmetro α^{ACO} regula o peso do feromônio e β regula a influência da informação heurística (distância inversa), em conformidade com a Equação 2.6 da Seção 2.6, sendo α^{ACO} distinto do parâmetro de vizinhança adaptativa α da Seção 4.1.2.2. Quando a seleção probabilística não encontra candidato viável, o sistema recorre ao vizinho viável mais próximo por critério determinístico, garantindo que soluções completas sempre sejam produzidas.

$$p(C_k) = \frac{\tau_{R_{last}, C_k}^{\alpha^{ACO}} \cdot \left(\frac{1}{\text{dist}(R_{last}, C_k)} \right)^\beta}{\sum_{C_l \in V_{cand}} \tau_{R_{last}, C_l}^{\alpha^{ACO}} \cdot \left(\frac{1}{\text{dist}(R_{last}, C_l)} \right)^\beta} \quad (4.2)$$

A cadeia de seleção do *AntSolution* segue uma hierarquia de quatro decisões encadeadas: o operador *SelectDepot* (Algoritmo 5) seleciona, a partir do vértice atual, o depósito com maior potencial de feromônio considerando os clientes ainda não visitados na vizinhança do último nó visitado; em seguida, os operadores *SelectCluster* (Algoritmo 6) e *SelectCustomer* (Algoritmo 7) elegem a lista de vizinhos e o cliente do caminhão pela regra probabilística que pondera feromônio e distância inversa.

O operador de seleção do drone, *SelectDrone* (Algoritmo 8), decide se e qual cliente será atendido pela frota aérea, consultando a matriz τ' e operando *somente* pela força do feromônio, sem componente de distância. Stodola e Kutěj (2024) observa que, para o drone, a distância não determina a rentabilidade da viagem, pois clientes mais distantes podem ser mais compensadores em termos de economia de makespan; logo, ao longo da otimização, o algoritmo aprende *quais* viagens do drone são efetivamente boas, independentemente da geometria. O lançamento de drones ocorre via probabilística $p(C_{drone}) = 1 - e^{-\tau_{\max}}$: no início da otimização, as trilhas são fracas e o drone raramente é lançado; à medida que viagens eficientes são reforçadas, τ' cresce e os lançamentos tornam-se progressivamente mais frequentes, superando probabilidades constantes da literatura (HUANG et al., 2022).

Cabe notar que o *SelectDepot* percorre todos os clientes ainda não visitados para estimar a atratividade de cada depósito, sendo o componente computacionalmente mais custoso da cadeia; além disso, uma vez que um depósito acumula trilhas de feromônio fortes, as formigas tendem a reescolhê-lo nas iterações seguintes, concentrando progressivamente as soluções em torno desse depósito.

Algoritmo 5 SelectDepot**Require:** $v_d, V_{free}, D, K^{(v_d)}, \tau, n_{prim}$ **Ensure:** $d_i \in D$

```

1: for cada  $d_i \in D$  do
2:    $V_{cand} \leftarrow \emptyset$ 
3:   for  $k = 1$  to  $n_{prim}$  do
4:      $V_{cand} \leftarrow V_{cand} \cup (V_{free} \cap K_k^{(v_d)})$ 
5:   end for
6:    $p(d_i) \leftarrow \sum_{v_j \in V_{cand}} \tau_{v_d, v_j}^d$ 
7: end for
8:  $p_{sum} \leftarrow \sum_{d_i \in D} p(d_i)$ 
9: for cada  $d_i \in D$  do
10:   $p(d_i) \leftarrow p(d_i)/p_{sum}$ 
11: end for
12: Selecionar  $d_i \in D$  com base nas probabilidades  $p(d_i)$ 
13: return  $d_i$ 

```

Algoritmo 6 SelectCluster**Require:** $d, v_d, V_{free}, K^{(v_d)}, \tau, n_{prim}, \alpha, \beta$ **Ensure:** Índice k do cluster

```

1: for  $k = 1$  to  $n_{prim}$  do
2:    $V_{cand} \leftarrow V_{free} \cap K_k^{(v_d)}$ 
3:   if  $V_{cand} = \emptyset$  then
4:      $\eta_k \leftarrow 0; \tau_k \leftarrow 0$ 
5:   else
6:      $\eta_k \leftarrow |V_{cand}| \cdot \sum_{v_j \in V_{cand}} |v_d - v_j|^{-1}$ 
7:      $\tau_k \leftarrow (1/|V_{cand}|) \cdot \sum_{v_j \in V_{cand}} \tau_{v_d, v_j}^d$ 
8:   end if
9: end for
10:  $\eta_{sum} \leftarrow \sum_{k=1}^{n_{prim}} \eta_k^\alpha$ 
11:  $\tau_{sum} \leftarrow \sum_{k=1}^{n_{prim}} \tau_k^\beta$ 
12: if  $\eta_{sum} = 0$  then
13:   for  $k = n_{prim} + 1$  to  $|K^{(v_d)}|$  do
14:      $V_{cand} \leftarrow V_{free} \cap K_k^{(v_d)}$ 
15:     if  $V_{cand} \neq \emptyset$  then
16:       return  $k$ 
17:     end if
18:   end for
19: end if
20: for  $k = 1$  to  $n_{prim}$  do
21:   $p(K_k^{(v_d)}) \leftarrow (\eta_k^\alpha \cdot \tau_k^\beta) / (\eta_{sum} \cdot \tau_{sum})$ 
22: end for
23: Selecionar  $K_k^{(v_d)} \in K^{(v_d)}$  com base nas probabilidades  $p(K_k^{(v_d)})$ 
24: return  $k$ 

```

Algoritmo 7 SelectCustomer**Require:** $d, v_d, V_{cand}, \alpha, \beta, \tau$ **Ensure:** $v_i \in V_{cand}$

- 1: **for** cada $v_i \in V_{cand}$ **do**
- 2: $p(v_i) \leftarrow \frac{|v_d - v_i|^{-\alpha} \cdot (\tau_{v_d, v_i}^d)^\beta}{\sum_{v_j \in V_{cand}} |v_d - v_j|^{-\alpha} \cdot (\tau_{v_d, v_j}^d)^\beta}$
- 3: **end for**
- 4: Selecionar $v_i \in V_{cand}$ com base nas probabilidades $p(v_i)$
- 5: **return** v_i

Algoritmo 8 SelectDrone

- 1: $C_{drone} \leftarrow \emptyset$; $\tau_{\max} \leftarrow 0$
- 2: $K_j^{(R_{last}^i)} \leftarrow \text{SelectCluster}(K^{(R_{last}^i)}, \tau')$ {usa τ' em vez de τ }
- 3: $V_{cnd} \leftarrow V_{free} \cap K_j^{(R_{last}^i)}$
- 4: **for** cada $C_k \in V_{cnd}$ **do**
- 5: $p \leftarrow (R_{last}^i, C_k, C_{truck})$
- 6: **if** $C_k^{demand} > U_i^{capacity}$ **then**
- 7: Remover C_k de V_{cnd}
- 8: **end if**
- 9: **if** $q_{route}^i + C_k^{demand} > T_i^{capacity}$ **then**
- 10: Remover C_k de V_{cnd}
- 11: **end if**
- 12: **if** $d_p^i > U_i^{endurance}$ **then**
- 13: Remover C_k de V_{cnd}
- 14: **end if**
- 15: **if** $t_{route}^i + d_p^i > T_i^{maxroute}$ **then**
- 16: Remover C_k de V_{cnd}
- 17: **end if**
- 18: **if** tempo de espera do drone $> U_i^{endurance}$ **then**
- 19: Remover C_k de V_{cnd}
- 20: **end if**
- 21: **if** $C_k \in V_{cnd}$ **then**
- 22: $\tau_{\max} \leftarrow \max(\tau_{\max}, \tau_{R_{last}^i, C_k}')$
- 23: **end if**
- 24: **end for**
- 25: $p(C_{drone}) \leftarrow 1 - e^{-\tau_{\max}}$
- 26: Com probabilidade $p(C_{drone})$: $C_{drone} \leftarrow \text{SelectCustomer}(V_{cnd}, \tau', \beta)$
- 27: **return** C_{drone}

4.1.4 Feromônio e evaporação adaptativa

Duas matrizes de feromônio independentes são mantidas: τ para caminhões e τ' para drones (STODOLA; KUTĚJ, 2024). A inicialização define $\tau_{j,k}^i = \tau_{j,k}'^i = 1$ para todos os pares de vértices. Ao final de cada iteração, as matrizes são atualizadas com base na

solução R^+, S^+ , a qual pode corresponder à melhor solução da iteração (R^{best}) ou à melhor solução global encontrada (R). Esse mecanismo de seleção (denominado SA global na literatura original) aceita soluções piores da iteração com probabilidade estocástica $p(R^{best}) = e^{-\Delta/T_{update}}$, onde $\Delta = (|R^{best}| - |R|)/|R|$ (STODOLA; NOHEL, 2023). Como o fator de resfriamento é fixado em $\alpha_{update} = 1,0$, a temperatura permanece constante em $T_{update} = 0,1$. Esse comportamento atua como um critério de aceitação estocástica a temperatura constante no estilo Metropolis: soluções ligeiramente piores são aceitas frequentemente (exemplo: uma degradação de 1% é aceita com probabilidade $e^{-0,1} \approx 0,90$), enquanto uma degradação de 100% resulta em $e^{-10} \approx 0$, revertendo à melhor solução global. Dessa forma, a melhor solução global prevalece na atualização apenas sob degradação severa; nos demais casos, soluções da iteração ligeiramente piores influenciam regularmente o feromônio, mantendo a diversidade sem o resfriamento gradual característico do *Simulated Annealing* convencional.

O reforço é então aplicado aos arcos da solução R^+ . Para os caminhões, a atualização é dada por $\tau_{r_{j-1}^i, r_j^i}^i \leftarrow \tau_{r_{j-1}^i, r_j^i}^i + \delta \cdot f(R, S)/f(R^+, S^+)$. Já para os drones, aplicam-se dois reforços por viagem do drone $p_j^{+i} = (l_j^{+i}, s_j^{+i}, f_j^{+i})$: um no arco de lançamento $l \rightarrow s$ e outro no arco de recuperação $s \rightarrow f$, em proporção à qualidade global da solução.

A evaporação é controlada pelo parâmetro ρ , calculado adaptativamente conforme a Equação 4.3 (STODOLA; NOHEL, 2023), onde H é a entropia de Shannon calculada sobre as frequências de uso de arcos (tanto de caminhões quanto de drones) em toda a população de formigas; $H_{\min}$ é o valor mínimo de entropia, atingido quando todas as soluções da população são idênticas, e $H_{\max}$ é o valor máximo, atingido quando as soluções diferem o máximo possível (STODOLA; NOHEL, 2023; STODOLA; KUTĚJ, 2024). Alta entropia (soluções diversas) implica ρ alto e convergência acelerada; baixa entropia (soluções similares) implica ρ baixo e preservação da estrutura encontrada. Quando H cai abaixo de um limiar configurável, o algoritmo é encerrado, pois novas iterações dificilmente produzirão melhoria.

$$\rho = \rho_{\min} + (\rho_{\max} - \rho_{\min}) \cdot \frac{H - H_{\min}}{H_{\max} - H_{\min}} \quad (4.3)$$

Uma limitação prática do critério de término baseado em entropia é sua assimetria entre instâncias. Nos experimentos deste trabalho, instâncias pequenas (ex.: p01-D, 50 clientes) convergem rapidamente, com o coeficiente normalizado $(H - H_{\min})/(H_{\max} - H_{\min})$ caindo a valores baixos (aproximadamente 10%), tornando o critério efetivo. Em instâncias grandes (ex.: p21-D, 360 clientes), a diversidade combinatorial é estruturalmente

maior: o coeficiente estabiliza em patamares intermediários (aproximadamente 20 a 30%) mesmo após a convergência prática da solução. [Stodola e Kutěj \(2024\)](#) não discute essa assimetria nem propõe um limiar adaptativo ao tamanho da instância; na prática, o critério de estagnação por iterações sem melhoria tende a ser mais determinante do que o critério de entropia para instâncias médias e grandes.

4.1.5 Busca local

A busca local é aplicada sobre a elite da geração com frequência configurável n_{freq} . O método de busca local geral, denominado *LocalOptimization* (Algoritmo 9), orquestra a aplicação de movimentos de melhoria sobre a solução corrente em duas fases principais: a otimização de rotas terrestres (busca local intra-rota e inter-rotas) e a otimização de viagens de drone (ajustes da frota aérea).

Algoritmo 9 LocalOptimization

Require: Solução inicial R, S

Ensure: Solução otimizada R, S

```

    {Etapa 1 - Busca local intra-rota (Or-k, 2-Opt e Exchange)}
1: for cada depósito  $i \in D$  do
2:   for  $n = 1$  to  $n_{\max}$  do
3:     Testar realocação intra-rota de blocos de  $n$  nós em  $R^i$  (Or-k)
4:     Aceitar movimento se for viável e reduzir  $f(R, S)$ 
5:   end for
6:   Testar inversão de subsegmentos (2-Opt) e troca de pares (Exchange) em  $R^i$ 
7:   Aceitar movimento se for viável e reduzir  $f(R, S)$ 
8: end for
    {Etapa 2 - Busca local inter-rotas (Shift-k e Swap)}
9: for  $n = 1$  to  $n_{\max}$  do
10:  for cada par de depósitos  $i, j \in D$  ( $i \neq j$ ) do
11:    Testar transferência de blocos de  $n$  nós de  $R^i$  para  $R^j$  (Shift-k)
12:    Testar troca de pares de nós entre  $R^i$  e  $R^j$  (Swap)
13:    Aceitar movimento se for viável e reduzir  $f(R, S)$ 
14:  end for
15: end for
    {Etapa 3 - Operadores específicos de drone}
16: Executar TruckDroneOpt (Algoritmo 10)
17: Executar DroneRouteOpt (Algoritmo 11)
18: return  $R, S$ 

```

É relevante destacar uma ambiguidade terminológica observada na literatura de referência. [Stodola e Kutěj \(2024\)](#) denomina o operador de busca local *k-opt* modificado, remetendo a [Stodola \(2020\)](#) sem detalhar o funcionamento exato. O *k-opt* clássico in-

verte subsegmentos de rota pela troca de arcos; Stodola (2020) descreve, por sua vez, a remoção de blocos de n nós sucessivos e sua reinserção em outra posição da rota, com n variando de 1 a $n_{\max}$. Neste trabalho adota-se essa interpretação de realocação de blocos. O operador *Or-k* (relocação intra-rota) remove blocos de 1 a $k_{\max}$ clientes sucessivos de sua posição e os reinsere em outra posição da mesma rota (*Or-1*, *Or-2*, *Or-3*). Em contrapartida, o operador *Shift-k* (relocação inter-rotas) remove tais blocos e os reinsere em rotas distintas (inclusive entre depósitos distintos). O *2-Opt* inverte segmentos internos de uma rota para reduzir o trajeto terrestre, enquanto as trocas de posições entre pares de nós são divididas entre o *Exchange* (troca intra-rota) e o *Swap* (troca inter-rotas). A Busca Local Composta combina esses operadores clássicos de forma estruturada em uma sequência determinística de refinamento: realocação de blocos (*Or-k/Shift-k*), inversão de subsegmentos (*2-Opt*) e trocas de nós (*Exchange/Swap*).

Após o refinamento das rotas terrestres, os operadores específicos para a frota aérea realizam ajustes de sincronização entre caminhões e drones. O operador *TruckDroneOpt* (Algoritmo 10) testa mover clientes da rota terrestre para novas viagens do drone aéreas viáveis, assim como reinserir clientes do drone na sequência terrestre. Já o operador *DroneRouteOpt* (Algoritmo 11) realiza o refinamento fino deslocando os pontos de lançamento e recuperação de cada viagem ativa do drone para posições anteriores ou posteriores na rota do caminhão, reduzindo o tempo de voo e a espera aérea.

Algoritmo 10 *TruckDroneOpt*

Require: R, S, R^i, S^i, T_i, U_i

Ensure: R^i, S^i

```

1:  $R' \leftarrow R$ 
2:  $S' \leftarrow S$ 
3: for cada  $r_j^i \in R^i$  do
4:    $R'^i \leftarrow R^i \setminus \{r_j^i\}$ 
5:    $S'^i \leftarrow S^i \cup \{(r_{j-1}^i, r_j^i, r_{j+1}^i)\}$ 
6:   if  $R'^i, S'^i$  é viável and  $f(R', S') < f(R, S)$  then
7:      $R^i \leftarrow R'^i$ 
8:      $S^i \leftarrow S'^i$ 
9:   end if
10: end for
11: return  $R^i, S^i$ 

```

Estratégias baseadas em *Simulated Annealing* foram implementadas e testadas. Embora admitam movimentos de piora com probabilidade $e^{-\Delta/T}$, em teoria ampliando a exploração do espaço de soluções, os resultados obtidos não indicaram ganho sistemático sobre a Busca Local Composta dentro do orçamento de tempo utilizado. A principal limi-

Algoritmo 11 DroneRouteOpt**Require:** R, S, R^i, S^i, T, U **Ensure:** S^i

```

1:  $S' \leftarrow S$ 
2:  $b_0^i \leftarrow 1$ 
3:  $a_{m_i+1}^i \leftarrow n_i$ 
4: for cada  $p_j^i = (r_{a_j^i}^i, s_j^i, r_{b_j^i}^i) \in S^i$  do
5:    $R^{*i} \leftarrow \{r_{b_{j-1}^i}^i, r_{b_{j-1}^i+1}^i, \dots, r_{a_{j-1}^i}^i\}$ 
6:   for cada  $r_k^i \in R^{*i}$  do
7:      $p_j'^i \leftarrow (r_k^i, s_j^i, r_{b_j^i}^i)$ 
8:      $S'^i \leftarrow (S^i \setminus \{p_j^i\}) \cup \{p_j'^i\}$ 
9:     if  $S'^i$  é viável and  $f(R, S') < f(R, S)$  then
10:        $p_j^i \leftarrow p_j'^i$ 
11:        $S^i \leftarrow S'^i$ 
12:     end if
13:   end for
14: end for
15: for cada  $p_j^i = (r_{a_j^i}^i, s_j^i, r_{b_j^i}^i) \in S^i$  do
16:    $R^{*i} \leftarrow \{r_{b_j^i+1}^i, r_{b_j^i+2}^i, \dots, r_{a_{j+1}^i}^i\}$ 
17:   for cada  $r_k^i \in R^{*i}$  do
18:      $p_j'^i \leftarrow (r_{a_j^i}^i, s_j^i, r_k^i)$ 
19:      $S'^i \leftarrow (S^i \setminus \{p_j^i\}) \cup \{p_j'^i\}$ 
20:     if  $S'^i$  é viável and  $f(R, S') < f(R, S)$  then
21:        $p_j^i \leftarrow p_j'^i$ 
22:        $S^i \leftarrow S'^i$ 
23:     end if
24:   end for
25: end for
26: return  $R^i, S^i$ 

```

tação observada é a sensibilidade extrema aos hiperparâmetros de temperatura inicial e fator de resfriamento: configurações mal calibradas desperdiçam o orçamento em regiões de baixa qualidade ou, inversamente, comportam-se como busca determinística por resfriamento prematuro. O ajuste fino dessas estratégias para o MDVRP-D constitui uma direção relevante para pesquisas futuras.

Em todas as fases, cada movimento candidato é validado contra capacidade, duração e viabilidade de viagens do drone antes de ser aceito. [Stodola e Kutěj \(2024\)](#), contudo, não esclarece o tratamento de vértices do caminhão que funcionam simultaneamente como ponto de lançamento ou recuperação de uma viagem do drone: mover tal vértice para outra posição na rota tende a invalidar a viagem do drone associada, pois o drone deve ser lançado e recuperado em vértices do caminhão respeitando a restrição de sequência

de viagens do drone (Seção 2.5.2). Neste trabalho, adota-se a abordagem de tentativa e validação: os movimentos são executados normalmente pelos operadores e, em seguida, a viabilidade da solução resultante é verificada. Como consequência, uma parcela expressiva dos movimentos que envolvem vértices de lançamento ou recuperação acaba descartada por violar a ordenação exigida pelas viagens do drone, o que reduz a eficácia da busca local. Estratégias alternativas, como bloquear o movimento de vértices envolvidos em viagens do drone ativas ou reconstruir dinamicamente as viagens do drone afetadas após cada movimento, constituem direções promissoras para trabalhos futuros.

4.2 Decomposição do problema

A resolução direta do MDVRP-D com m depósitos e n clientes combina dois problemas acoplados: a atribuição de clientes a depósitos e o roteamento dentro de cada depósito, com espaço combinatorial da ordem de $O(m^n \times n!)$. A decomposição proposta nesta seção apoia-se em quatro motivações: (i) o paradigma *Cluster-First, Route-Second* (GILLET; MILLER, 1974) é a abordagem clássica para o MDVRP desde 1976; a disponibilização de VANTs para logística de última milha traz a possibilidade de estendê-lo ao MDVRP-D; (ii) ao fracionar o MDVRP-D em sub-problemas VRP-D, o AACO-NC-D opera no domínio em que sua eficácia já foi estabelecida: Stodola e Kutěj (2024) demonstrou que supera o ALNS em 17 dos 28 grupos de instâncias VRP-D; (iii) a decomposição reduz as matrizes de feromônio de $O(m \times n^2)$ para $O(n^2/m)$, ganho de m^2 que permite maior profundidade de busca intra-depósito; e (iv) na estrutura multi-depósito, cada depósito tende a servir os clientes geograficamente próximos; a Partição de Voronoi formaliza essa estrutura latente, tornando as fronteiras da decomposição geograficamente naturais. A contribuição não reside na ideia de decompor, mas na avaliação sistemática de como diferentes modos de integração entre a decomposição e o AACO-NC-D sobre o espaço completo afetam a qualidade das soluções.

A Partição de Voronoi oferece uma solução natural para esse acoplamento: com o parâmetro de diversidade $\alpha = 0$ (agrupamentos disjuntos, sem vizinhos externos), os n clientes são particionados em m grupos *antes* da otimização, transformando o MDVRP-D em m sub-problemas VRP-D independentes. Essa abordagem é denominada *Decomposição Pura*. Cada sub-problema é resolvido por uma instância autônoma do AACO-NC-D, e as soluções são fundidas em pós-processamento. A estrutura de execução da estratégia *PureDecomposition* é apresentada no Algoritmo 12.

O benefício computacional da decomposição resulta de três fatores que se multiplicam entre si. O primeiro fator decorre do escalamento quadrático das matrizes de feromônio: enquanto a instância completa mantém matrizes τ e τ' de dimensão $O(m \times n^2)$, a decomposição em m sub-instâncias de n/m clientes reduz o total de entradas para $m \times (n/m)^2 = n^2/m$, representando para $m = 4$ uma redução de 16 vezes apenas nas operações de reforço e evaporação (Tabela 5). O segundo fator refere-se à redução do espaço combinatorial, uma vez que a atribuição de clientes a depósitos é resolvida pelo agrupamento prévio, cabendo ao ACO apenas o roteamento local de complexidade $O((n/m)!)$ (o que atenua a estagnação e facilita o refino estocástico). Por fim, o terceiro fator diz respeito ao custo por formiga, cujo tempo de construção por solução cai de $O(n)$ na instância global para $O(n/m)$ por sub-problema.

Tabela 5: Comparação de complexidade: instância completa vs. decomposição

Dimensão	Instância Completa	Decomposição
Matrizes de feromônio	$O(m \times n^2)$	$O(n^2/m)$
Espaço combinatorial	atribuição + roteamento	apenas roteamento
Custo por formiga	$O(n)$	$O(n/m)$

Algoritmo 12 PureDecomposition

Require: V, D, C, T, U , parâmetros do AACO-NC-D

Ensure: $R = \{R^1, \dots, R^m\}, S = \{S^1, \dots, S^m\}$

```

1: for cada  $c \in C$  do
2:    $i \leftarrow \arg \min_j \text{dist}(c, D_j)$ 
3:    $C_i \leftarrow C_i \cup \{c\}$ 
4: end for
5: for cada  $D_i \in D$  do
6:    $V_i \leftarrow C_i \cup \{D_i\}$ 
7:    $R^i, S^i \leftarrow \text{AACO-NC-D}(V_i, T_i, U_i, \text{parâmetros})$ 
8: end for
9: return  $R \leftarrow \{R^1, \dots, R^m\}, S \leftarrow \{S^1, \dots, S^m\}$ 

```

Embora a decomposição seja eficaz dentro de cada sub-problema, o agrupamento prévio impõe fronteiras fixas entre depósitos. Em instâncias com clientes geograficamente concentrados, a Partição de Voronoi pode criar sub-problemas de tamanhos muito distintos: o maior sub-problema domina o *makespan* global, anulando o ganho de performance e produzindo soluções de qualidade inferior à abordagem sobre o espaço completo. A razão entre o maior e o menor sub-cluster funciona como indicador de risco: valores próximos de 1 indicam distribuição equilibrada e são favoráveis à decomposição pura; valores elevados indicam instâncias onde a integração com o AACO-NC-D sobre o espaço completo (Seção 4.3) é preferível.

Para exemplificar, em uma instância com 100 clientes e 4 depósitos, se a partição atribuir aproximadamente 25 clientes a cada depósito, a razão entre o maior e o menor sub-cluster é próxima de 1,0; as rotas resultantes tendem a apresentar durações semelhantes, resultando em um *makespan* global equilibrado. Em contrapartida, se a distribuição geográfica concentrar 50 clientes em um único depósito e apenas 12 no menor, a razão atinge um valor elevado ($50/12 \approx 4,17$); o depósito sobrecarregado tende a ditar a duração total do sistema, enquanto os demais concluem o atendimento precocemente e permanecem ociosos.

4.3 Estratégias de integração da decomposição

A decomposição pura e o AACO-NC-D sobre o espaço completo representam dois extremos operacionais. O primeiro é computacionalmente eficiente e explora profundamente os sub-problemas locais, mas apresenta limitações na resolução do balanceamento de carga global, pois atua com fronteiras rígidas impostas pelo agrupamento prévio. O segundo, por sua vez, resolve de maneira eficaz as atribuições inter-depósito e o equilíbrio de carga, mas enfrenta restrições de convergência e desempenho temporal em instâncias de grande porte.

Para mitigar essas limitações, propõem-se duas estratégias complementares de hibridização, desenhadas para extrair os benefícios de ambas as abordagens: a *Decomposição como Semente* e a *Decomposição como Refinamento*. As duas estratégias esgotam as ordens possíveis de encadeamento: semente aplica decomposição primeiro e espaço completo depois; refinamento inverte a ordem.

4.3.1 Decomposição como semente

A inicialização guiada utiliza a decomposição para gerar um conjunto de rotas de alta qualidade em tempo reduzido. O conhecimento estrutural obtido é então transferido para o AACO-NC-D sobre o espaço completo antes da sua primeira iteração. Essa estratégia opera em duas fases sequenciais: inicialmente, a instância é dividida em sub-problemas disjuntos (Partição de Voronoi com $\alpha = 0$), os quais são otimizados de forma independente. Posteriormente, as conexões da solução particionada são utilizadas para inicializar as matrizes de feromônio, com a intensidade ajustada por um fator de reforço.

Na fase com múltiplos depósitos, o AACO-NC-D opera com parâmetros de diversidade mais altos ($\alpha > 0$), permitindo explorar as fronteiras inter-depósito negligenciadas pela decomposição inicial. Apesar da eficiência na redução do espaço de busca, essa abordagem

apresenta limitações relacionadas ao equilíbrio de carga. O particionamento puramente espacial, realizado de forma cega antes do roteamento, ignora os gargalos operacionais e a geometria real das rotas, o que frequentemente cria desequilíbrios significativos na distribuição de demandas. Cabe ressaltar que o gargalo metodológico reside exclusivamente na etapa de agrupamento e não na otimização: mesmo garantindo um tempo prolongado de convergência, o desequilíbrio estrutural entre depósitos não é corrigível pelas instâncias locais da decomposição.

Além disso, a exploração livre do ACO não assegura a manutenção do equilíbrio estrutural transferido. Na fase subsequente de otimização, a dinâmica do feromônio tende a dissolver a organização inicial. Devido a essa limitação natural, a inicialização guiada mostra-se particularmente eficaz apenas quando há um desequilíbrio geográfico ou estrutural severo prévio na instância. Nesses cenários, o agrupamento prévio fornece uma solução inicial estruturada que orienta e facilita o refinamento na etapa multi-depósito. O fluxo algorítmico da estratégia *DecompositionSeed* é detalhado no Algoritmo 13.

Algoritmo 13 *DecompositionSeed*

Require: V, D, C, T, U , parâmetros, $\delta > 0$, $\alpha_{global} > 0$

Ensure: R, S

```

{Fase 1}
1: for cada  $c \in C$  do
2:    $i \leftarrow \arg \min_j \text{dist}(c, D_j)$ ;  $C_i \leftarrow C_i \cup \{c\}$ 
3: end for
4: for cada  $D_i \in D$  do
5:    $R_i^{seed}, S_i^{seed} \leftarrow \text{AACO-NC-D}(C_i \cup \{D_i\}, T_i, U_i, \text{parâmetros})$ 
6: end for
{Fase 2}
7:  $\tau_{j,k}^i \leftarrow \tau_{j,k}^i \leftarrow 1$  para todo  $V_j, V_k \in V, D_i \in D$ 
8: for cada  $D_i \in D$  do
9:    $q_i \leftarrow \delta / f(R_i^{seed}, S_i^{seed})$ 
10:  for cada aresta  $(u, v)$  em  $R_i^{seed}$  do
11:     $\tau_{u,v}^i \leftarrow \tau_{u,v}^i + q_i$ 
12:  end for
13:  for cada viagem do drone  $p = (l, c_{drone}, f_p) \in S_i^{seed}$  do
14:     $\tau_{l, c_{drone}}^i \leftarrow \tau_{l, c_{drone}}^i + q_i$ 
15:     $\tau_{c_{drone}, f_p}^i \leftarrow \tau_{c_{drone}, f_p}^i + q_i$ 
16:  end for
17: end for
{Fase 3}
18:  $R, S \leftarrow \text{AACO-NC-D}(V, D, C, T, U, \alpha = \alpha_{global}, \tau_{init}, \tau'_{init}, \text{parâmetros})$ 
19: return  $R, S$ 

```

4.3.2 Decomposição como refinamento

Diferente da inicialização, a estratégia de refinamento inverte a ordem das operações: ela posiciona o AACO-NC-D sobre o espaço completo como a ferramenta principal para buscar o equilíbrio macroscópico da carga e repassa à decomposição o papel de intensificação local ao final do processo. Dessa forma, as duas componentes executam estritamente o tipo de busca na qual são mais eficientes.

Durante a fase com múltiplos depósitos, o ACO realiza o trabalho computacionalmente mais custoso de nivelar o desbalanceamento entre depósitos e explorar o espaço de atribuições de maneira ampla. Uma vez que o algoritmo estabiliza e as conexões de clientes por depósito encontram-se fixadas na melhor solução global encontrada, o resultado é transferido à decomposição. Nesse estágio, o particionamento não opera de modo cego; ele herda os agrupamentos já empiricamente validados pela dinâmica do feromônio.

Cada sub-problema recebe um conjunto restrito e bem definido de clientes associados exclusivamente a um depósito. O objetivo da decomposição torna-se unicamente otimizar a sequência de rotas de cada base sem alterar o balanceamento macroscópico. Como o espaço de busca intra-depósito é ordens de grandeza menor do que o problema original, o algoritmo local consegue localizar melhorias na configuração interna de viagens que o ACO sobre o espaço completo, condicionado por suas matrizes vastas e evaporação generalizada, não possuía foco para localizar de modo exato.

Em contraposição à estratégia semente, a abordagem por refinamento revela-se robusta e vantajosa independentemente do perfil geográfico da instância, gerando valor expressivo especialmente quando a busca exploratória no espaço completo deixa margem de otimização no roteamento interno individual. A estrutura de execução da estratégia *DecompositionRefinement* está esquematizada no Algoritmo 14.

4.4 Propostas de estudo de ablação

Com base na replicação do modelo-base e nas investigações propostas, alguns estudos de ablação são conduzidos. O primeiro foco avalia o impacto da clusterização por meio da comparação entre a Partição de Voronoi e a Setorização Angular, mapeando o efeito do zoneamento na qualidade final. O segundo analisa o impacto da decomposição, confrontando a otimização no espaço completo contra a resolução de sub-instâncias VRP-D independentes, investigando o equilíbrio entre tempo e qualidade em função do desba-

Algoritmo 14 DecompositionRefinement

Require: V, D, C, T, U , parâmetros globais, parâmetros locais, $\alpha_{global} > 0$ **Ensure:** R, S

```

    {Fase 1}
1:  $R^*, S^* \leftarrow \text{AACO-NC-D}(V, D, C, T, U, \alpha=\alpha_{global}, \text{parâmetros globais})$ 
    {Fase 2}
2: for cada  $D_i \in D$  do
3:    $C_i \leftarrow \{c \in C \mid c \text{ atribuído a } D_i \text{ em } R^*\}$ 
4: end for
    {Fase 3}
5: for cada  $D_i \in D$  do
6:    $R^i, S^i \leftarrow \text{AACO-NC-D}(C_i \cup \{D_i\}, T_i, U_i, \text{parâmetros locais})$ 
7: end for
8: return  $R \leftarrow \{R^1, \dots, R^m\}, S \leftarrow \{S^1, \dots, S^m\}$ 

```

lançamento geográfico. Por fim, avalia-se a integração decomposição-AACO-NC-D ao confrontar as estratégias de Decomposição como Semente e Decomposição como Refinamento para medir a velocidade de convergência e a robustez final das rotas, fixando-se o parâmetro de diversidade do espaço completo em $\alpha = 1,0$.

5 Experimentos computacionais

Este capítulo apresenta os resultados computacionais obtidos com a implementação do AACO-NC-D descrita no Capítulo 4. Os experimentos são organizados em duas etapas: avaliação nas instâncias MDVRP-D e o estudo de ablação dos componentes propostos.

5.1 Configuração experimental e instâncias

Para garantir uma avaliação independente, o algoritmo base de [Stodola e Kutěj \(2024\)](#) foi objeto de uma réplica técnica em C++. A compilação requer um compilador compatível com o padrão C++20 e suporte à biblioteca OpenMP para a avaliação paralela da colônia de formigas. Os experimentos foram conduzidos em um ambiente com processador Intel Core i5-13420H (12 threads de execução) e 16 GB de memória RAM.

A paralelização opera no nível de formigas: durante a fase de construção, a população é distribuída entre as threads disponíveis e cada thread constrói sua solução de forma totalmente independente, mantendo suas próprias listas de clientes visitados e geradores de números pseudoaleatórios. A matriz de feromônio (τ) reside em memória compartilhada; durante a construção, todas as threads apenas leem essa estrutura, dispensando bloqueios de concorrência. Ao término de cada iteração, uma barreira de sincronização garante que nenhuma thread avance antes que toda a colônia tenha concluído a construção. A deposição de feromônio é então executada de forma sequencial pela thread principal, garantindo que cada trilha τ_{ij} receba exatamente o incremento $\delta \cdot f(R,S)/f(R^+,S^+)$ calculado sobre a solução de referência. A evaporação é executada em paralelo, com as células da matriz distribuídas automaticamente entre as threads disponíveis; como cada célula é processada por exatamente uma thread, não há conflitos de escrita. Como consequência, sob uma mesma semente de números pseudoaleatórios, o comportamento estocástico da versão paralela é idêntico ao da versão serial, garantindo a reprodutibilidade dos resultados.

Para isolar o impacto individual das propostas (estratégias de clusterização e de decomposição), os experimentos das Seções 5.2 e 5.3 mantiveram os módulos de busca local

desabilitados, adotando um critério de parada de 2.000 iterações sem melhoria em 10 execuções independentes por configuração. Como [Stodola e Kutěj \(2024\)](#) não especifica o limite de estagnação adotado originalmente, fixou-se um parâmetro que assegura exploração substancial dentro de um orçamento computacional viável. Por tratar-se de avaliações relativas sob protocolos idênticos entre as variantes, a validade comparativa é preservada. O efeito da busca local é analisado especificamente na Seção 5.3.3 e mantido nos estudos das Seções 5.4 e 5.5. A comparação com a literatura adota a minimização do *makespan* como função-objetivo.

A Tabela 6 detalha o conjunto de instâncias utilizado, abrangendo diferentes escalas e distribuições geográficas. A coluna de elegibilidade indica o percentual de clientes com demanda $\leq U^{capacity} = 10$ kg, único critério que determina se um cliente pode ser atendido por drone; os demais são atendidos exclusivamente pelo caminhão.

Tabela 6: Especificações das instâncias de teste MDVRP-D

Instância	Clientes	Depósitos	Distribuição	Elegíveis por drone (%)
p01-D a p02-D	50	4	Aleatória	34,0
p03-D	75	5	Aleatória	16,0
p04-D a p07-D	100	2 a 4	Aleatória	40,0
p08-D a p11-D	249	2 a 5	Aleatória	12,9
p12-D a p14-D	80	2	Regular	80,0
p15-D a p17-D	160	4	Regular	80,0
p18-D a p20-D	240	6	Regular	80,0
p21-D a p23-D	360	9	Regular	80,0

Os parâmetros operacionais dos veículos seguem os valores definidos por [Stodola e Kutěj \(2024\)](#), conforme sintetizados na Tabela 7.

Tabela 7: Parâmetros operacionais dos caminhões e drones

Parâmetro	Símbolo	Valor
Velocidade do caminhão	$T^{velocity}$	15 m/s
Velocidade do drone	$U^{velocity}$	30 m/s
Capacidade de carga do drone	$U^{capacity}$	10 kg
Autonomia de voo do drone	$U^{endurance}$	1.200 s
Tempo de lançamento do drone	U^{launch}	60 s
Tempo de recuperação do drone	$U^{recovery}$	60 s
Tempo de serviço do caminhão	$T^{service}$	300 s
Tempo de serviço do drone	$U^{service}$	150 s

Os parâmetros de controle do AACO-NC-D seguem os valores estabelecidos por [Stodola e Nohel \(2023\)](#) e adotados sem modificação por [Stodola e Kutěj \(2024\)](#), conforme a

Tabela 8. O parâmetro α^{ACO} denota o coeficiente de peso da intensidade do feromônio na regra probabilística do ACO, sendo distinto do parâmetro de vizinhança adaptativa α da Seção 4.1.2.2.

Tabela 8: Parâmetros de controle do AACO-NC-D

Parâmetro	Descrição	Valor
n_{ants}	Número de formigas	192
n_{freq}	Frequência da busca local (iterações)	10
n_{prim}	Número de clusters primários	3
n_{size}	Vértices por cluster	24
n_{sect}	Setores angulares (Setorização Angular)	16
T_{update}	Temperatura inicial do SA global	0,1
α_{update}	Coeficiente de resfriamento do SA global	1
ρ_{min}	Limite mínimo da evaporação adaptativa	0,001
ρ_{max}	Limite máximo da evaporação adaptativa	0,1
δ	Coeficiente de reforço do feromônio	3
α^{ACO}	Peso da informação heurística (distância)	1
β	Peso do feromônio	1

5.1.1 Métricas de avaliação e ambiente computacional

As métricas principais de avaliação de desempenho adotadas nos experimentos são: a qualidade da solução (medida pelo tempo de conclusão da rota crítica, ou *makespan*, permitindo a comparação direta com a literatura de referência); o gap médio (que expressa a diferença percentual entre a melhor solução encontrada e a média das execuções); e o desempenho computacional (medido pelo tempo total de processamento necessário em segundos).

Os experimentos foram conduzidos em um ambiente computacional local composto por processador Intel Core i513420H (12 threads de execução) e 16 GB de memória RAM. As especificações detalhadas e a comparação com o ambiente de [Stodola e Kutěj \(2024\)](#) estão resumidas na Tabela 9. O processador do ambiente original de referência possui uma pontuação *multi-thread* no *PassMark* de 27.668, contra 17.025 do processador utilizado neste trabalho ([PASSMARK SOFTWARE, 2026](#)). Devido a essa diferença de capacidade computacional (de aproximadamente 38%), os tempos absolutos de execução registrados tendem a ser naturalmente superiores, sendo a comparação de desempenho temporal interpretada sob essa ressalva.

Tabela 9: Comparação dos ambientes computacionais

Especificação	Ambiente de Stodola	Ambiente Local (Este Trabalho)
Processador	Intel Core i9-10940X	Intel Core i5-13420H
Memória RAM	32 GB	16 GB
Núcleos/Threads	14 Núcleos / 28 Threads	8 Núcleos (4P + 4E) / 12 Threads
Frequência Base	3.30 GHz	2.10 GHz (P-core) / 1.50 GHz (E-core)
Frequência Turbo	Até 4.80 GHz (Boost Max 3.0)	Até 4.60 GHz
Memória Cache	19.25 MB Intel® Smart Cache	12 MB Intel® Smart Cache
TDP Base	165 W	45 W

5.1.2 Análise crítica das restrições físico-operacionais das instâncias

As instâncias do MDVRP-D propostas por [Stodola e Kutěj \(2024\)](#) adotam premissas operacionais simplificadas que afetam o comportamento do modelo. A seguir, analisam-se três limitações físicas centrais desse conjunto de testes.

O primeiro ponto refere-se à velocidade relativa do drone. O modelo fixa a velocidade do drone em $U^{velocity} = 30$ m/s (108 km/h) e a do caminhão em $T^{velocity} = 15$ m/s (54 km/h). Essa proporção de 2:1 favorece o uso da frota aérea na formulação teórica. Contudo, em operações reais, fatores como vento contrário, tempo de manobra e restrições de voo urbano reduzem a velocidade média efetiva. Quando a velocidade do drone se aproxima da velocidade do caminhão, o ganho da frota aérea passa a depender quase exclusivamente de deslocamentos em linha reta que evitem desvios da malha viária.

O segundo ponto envolve a capacidade de carga e a ausência de entregas fracionadas (*split delivery*). Com capacidade limite de $U^{capacity} = 10$ kg e a restrição de visita única por cliente, qualquer demanda superior a esse valor impede o atendimento por drone. Na prática, essa regra obriga o caminhão a assumir todas as entregas mais pesadas. O suporte a entregas fracionadas permitiria que o drone realizasse viagens consecutivas para o mesmo cliente, ampliando a utilização da frota aérea em demandas volumosas.

Por fim, a elegibilidade efetiva dos clientes é menor do que a limitação isolada de peso sugere. Nas instâncias de maior porte (p20-D a p23-D), cerca de 42% dos clientes excedem a capacidade de 10 kg. Ao associar essa restrição ao limite de autonomia de bateria ($U^{endurance} = 1.200$ s), apenas entre 30% e 45% dos clientes de cada instância podem ser atendidos por drone. Esse perfil caracteriza o MDVRP-D como um problema essencialmente terrestre, no qual a frota aérea realiza atendimentos pontuais em clientes leves de fronteira ou geograficamente isolados.

5.2 Revalidação computacional das soluções publicadas

O validador desenvolvido analisou de forma independente as 1.150 soluções publicadas por [Stodola e Kutěj \(2024\)](#). O processo de auditoria ocorreu em duas etapas: (i) alinhamento de resultados, na qual o módulo processou as rotas originais para confirmar que o cálculo da função objetivo reproduzia exatamente os valores reportados na literatura; e (ii) auditoria de restrições, onde todas as condições de viabilidade do MDVRP-D foram testadas rigorosamente. O diferencial desta análise foi avaliar o comportamento do modelo sob duas variantes operacionais de restrição de bateria do drone. Na *Variante A* (*com tempo de espera estrito*), correspondente à configuração formalizada na modelagem teórica por [Stodola e Kutěj \(2024\)](#), a sincronização temporal é rígida: o consumo de bateria do drone é contabilizado desde o lançamento até o efetivo resgate pelo caminhão, integrando o tempo de serviço do caminhão ($T^{service}$) e qualquer espera adicional no ponto de encontro, no qual o drone consome energia em voo estacionário. Por sua vez, a *Variante B* (*sem tempo de espera*) representa uma simplificação operacional na qual o tempo de serviço do caminhão ($T^{service}$) durante paradas intermediárias não é contabilizado no consumo de autonomia do drone, considerando que o caminhão recupera o drone assim que chega no vértice de recuperação, realizando sua entrega logo em seguida.

A auditoria computacional revelou um detalhe metodológico importante: embora [Stodola e Kutěj \(2024\)](#) tenha formalizado a Variante A no texto de referência, os resultados numéricos publicados indicam a omissão do tempo de serviço do caminhão ($T^{service}$) durante a espera. Dessa forma, das 1.150 soluções publicadas originalmente para o *makespan*, apenas 217 (18,9%) atendem estritamente às restrições da Variante A.

Cabe ressaltar que, nas instâncias de pequeno porte (p01-D a p07-D), grande parte das soluções de [Stodola e Kutěj \(2024\)](#) preserva a viabilidade sob ambas as variantes. Como os trechos terrestres associados são curtos, o tempo de espera acumulado do drone não excede o limite de autonomia de 1.200 s. Em contrapartida, nas instâncias de médio e grande porte (p08-D a p23-D), 100% das soluções da literatura de referência violam a restrição da Variante A, sendo aderentes exclusivamente à Variante B. Como verificação controlada, ao suprimir o tempo de serviço do caminhão ($T^{service}$) do cálculo de autonomia (reduzindo o teste à Variante B), todas as 933 soluções que violavam o modelo estrito recuperaram a viabilidade, confirmando que 100% das soluções publicadas são aderentes às premissas da Variante B.

Para exemplificar esse efeito, a Figura 13 ilustra uma viagem de drone inviável na

instância p20-D. Nesse exemplo, o drone é lançado no nó 6, atende o cliente 20 e desloca-se até o nó de resgate 22, enquanto o caminhão realiza a rota terrestre direta do nó 6 ao nó 22 e atende o cliente 22. O percurso próprio do drone consome 1.142,68 s (somando tempos de lançamento, voos $6 \rightarrow 20$ e $20 \rightarrow 22$, atendimento em 20 e recuperação). No entanto, o percurso do caminhão até o ponto de encontro consome 1.362,81 s, já que inclui o tempo de serviço no cliente 22 ($T^{service} = 300$ s). Sob a Variante A, como o drone deve aguardar a conclusão do serviço do caminhão em voo estacionário, a duração efetiva da operação aérea expande-se para 1.362,81 s, ultrapassando o limite de autonomia de 1.200 s.

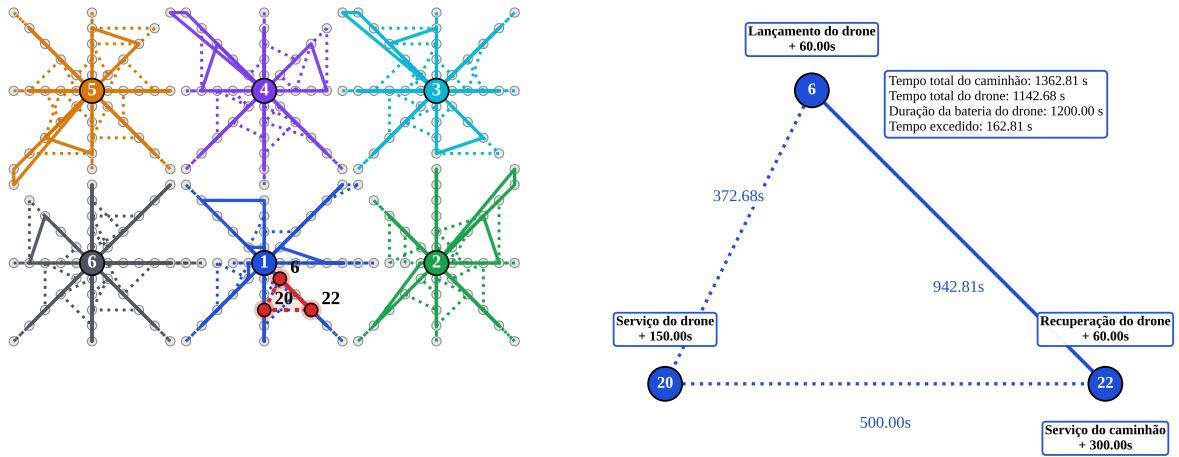


Figura 13: Exemplo de violação de autonomia na instância p20-D: (a) visão geral da rota; (b) detalhe da viagem do drone entre os nós 6, 20 e 22.

A Tabela 10 apresenta, por instância, o número de soluções válidas sob o Modelo Estrito de Autonomia (Variante A) entre as 50 executadas, a melhor solução válida encontrada, a melhor solução originalmente reportada por Stodola e Kutěj (2024) e a variação percentual resultante. Das 23 melhores soluções individuais reportadas por Stodola e Kutěj (2024) para makespan, apenas quatro (p02, p03, p05 e p06) permaneceram válidas e mantiveram o melhor resultado sob a Variante A. Em três instâncias (p01, p04 e p07), a melhor solução original violou a restrição da Variante A, mas encontrou-se ao menos uma alternativa válida entre as 50 execuções, com desvio máximo de 1,05%. As 16 instâncias restantes (p08 a p23) não apresentaram nenhuma solução válida sob o protocolo estrito da Variante A.

Tabela 10: Revalidação por instância

Instância	Válidas	Melhor válida	Melhor publicada	Var (%)
p01-D	19/50	7.311,30	7.311,30	+0,00
p02-D	39/50	6.560,22	6.560,22	+0,00
p03-D	49/50	8.201,77	8.201,77	+0,00
p04-D	4/50	26.020,01	25.749,78	+1,05
p05-D	39/50	21.544,14	21.544,14	+0,00
p06-D	45/50	15.987,21	15.987,21	+0,00
p07-D	22/50	12.121,39	12.056,70	+0,54
p08-D	0/50	—	114.693,98	—
p09-D	0/50	—	67.137,57	—
p10-D	0/50	—	48.325,85	—
p11-D	0/50	—	38.385,86	—
p12-D	0/50	—	26.297,99	—
p13-D	0/50	—	28.920,24	—
p14-D	0/50	—	31.717,19	—
p15-D	0/50	—	26.335,04	—
p16-D	0/50	—	30.536,91	—
p17-D	0/50	—	33.000,61	—
p18-D	0/50	—	26.851,32	—
p19-D	0/50	—	32.839,42	—
p20-D	0/50	—	35.744,56	—
p21-D	0/50	—	27.513,61	—
p22-D	0/50	—	35.060,45	—
p23-D	0/50	—	38.500,84	—

5.2.1 Análise por perfil de instância

A auditoria detalhada revela um padrão mecânico de inviabilidade sob a Variante A condicionado ao perfil de cada instância.

Nas instâncias pequenas (p01-D a p07-D, 50 a 100 clientes), os segmentos de rota terrestres associados às viagens do drone são curtos e o lançamento e recuperação do drone ocorrem em vértices próximos. Nesse cenário, o drone raramente aguarda mais de um cliente intermediário sendo atendido pelo caminhão, e o tempo de espera acumulado não ultrapassa a autonomia de 1.200 s. As soluções originais são majoritariamente válidas nesse grupo (variação máxima de 1,05% em relação ao melhor resultado originalmente reportado, para p04-D).

Nas instâncias de médio e grande porte (p08-D a p23-D), o mecanismo muda. As viagens dos caminhões são mais longas e, por consequência, o caminhão frequentemente atende dois ou mais clientes entre o lançamento e a recuperação do drone. Cada cliente

intermediário acrescenta $T^{service}$ s de espera ao drone em voo estacionário; com dois a três clientes intermediários, o tempo total da viagem do drone ultrapassa os 1.200 s de autonomia. Daí o índice de 100% de inviabilidade sob a Variante A em todo o bloco p08-D a p23-D. Esse bloco não dispõe de referências válidas na literatura; este trabalho propõe os primeiros.

5.3 Estudo de ablação dos componentes

5.3.1 Impacto da estratégia de clusterização

A Tabela 11 apresenta a análise de sensibilidade do parâmetro α , que controla o equilíbrio entre proximidade geográfica e diversidade nas listas de candidatos. Os resultados demonstram que a inclusão de diversidade ($\alpha > 0$) é benéfica na maioria das instâncias com distribuição aleatória (p01 a p11), onde as fronteiras de atendimento são irregulares e um cliente pode estar mais integrado à rota lógica de um depósito adjacente devido à densidade local. Contudo, exceções notáveis como p04-D e p08-D obtiveram seus melhores resultados com $\alpha = 0$, sugerindo que, nesses cenários, a restrição geográfica pura foi suficiente para guiar a construção de boas rotas. Em outras palavras, embora o mecanismo proposto seja geralmente benéfico, sua utilidade depende das características topológicas de cada instância. Já em instâncias com distribuição regular (p12 a p23), o impacto da diversidade é mais variado: embora o aumento de α eleve o tempo computacional e possa saturar a busca com nós distantes, instâncias como p20-D e p21-D alcançaram reduções significativas na função objetivo com $\alpha \geq 0,5$, indicando que o mecanismo consegue identificar conexões entre depósitos que compensam o custo computacional adicional. Os valores $\alpha = 0$ (clusters disjuntos, sem sobreposição) e $\alpha = 1,0$ (sobreposição máxima das listas de candidatos) representam os extremos estruturais do mecanismo: o primeiro preserva a partição estrita de Voronoi; o segundo maximiza a cobertura de clientes nas zonas de fronteira. O valor $\alpha = 0,5$ é incluído como ponto intermediário para verificar se os ganhos de sobreposição são graduais ou concentrados nos extremos.

Em 6 das 23 instâncias (p02-D, p06-D, p07-D, p09-D, p10-D e p11-D), $\alpha = 0,5$ produz o menor custo individual, superando ambos os extremos. Esse resultado indica que a sobreposição parcial captura as conexões inter-depósito mais valiosas sem o custo computacional pleno de $\alpha = 1,0$. Os experimentos comparativos das seções seguintes fixam $\alpha \in \{0, 1, 0\}$ como pontos de referência sistemáticos ($\alpha = 0$ para isolar o efeito da partição pura e $\alpha = 1,0$ para avaliar o limite superior da diversidade), deixando a

Tabela 11: Análise de sensibilidade da constante de diversidade α .

Instância	Melhor (Makespan)			Média (Makespan)			Tempo Médio (s)		
	$\alpha = 0,0$	$\alpha = 0,5$	$\alpha = 1,0$	$\alpha = 0,0$	$\alpha = 0,5$	$\alpha = 1,0$	$\alpha = 0,0$	$\alpha = 0,5$	$\alpha = 1,0$
p01-D	11.318,33	7.643,08	7.567,87	11.407,44	7.711,58	7.681,74	8,6	12,8	17,8
p02-D	10.037,62	6.796,19	6.888,01	10.112,60	6.980,09	7.029,79	8,8	12,4	11,2
p03-D	10.518,68	8.778,55	8.756,75	10.691,82	8.944,01	8.940,25	21,3	26,2	26,0
p04-D	27.885,35	28.081,31	28.524,42	28.332,40	28.689,28	28.762,48	61,8	49,4	73,0
p05-D	24.957,42	23.371,15	23.351,27	25.500,56	24.231,61	24.208,46	38,8	58,2	75,3
p06-D	21.746,06	17.191,46	17.318,16	21.993,50	17.420,40	17.577,10	29,8	45,4	62,9
p07-D	13.644,25	13.285,70	13.474,49	13.977,62	13.529,68	13.676,39	34,7	40,7	47,6
p08-D	130.045,15	130.731,87	130.692,96	134.702,63	133.602,97	134.268,30	328,5	345,1	387,1
p09-D	89.434,54	76.248,21	76.694,49	90.854,88	78.095,20	77.959,45	220,1	265,2	281,7
p10-D	60.127,48	54.735,86	55.655,93	61.268,70	56.264,90	56.645,50	151,0	235,0	291,3
p11-D	50.639,01	43.279,61	43.799,47	51.266,07	44.271,17	44.719,76	99,3	163,3	205,9
p12-D	29.153,32	29.074,72	28.896,87	29.560,05	29.860,63	29.633,25	29,0	27,2	27,1
p13-D	32.163,02	33.066,90	32.489,69	33.715,08	34.212,48	33.849,94	35,1	35,0	41,4
p14-D	37.755,10	37.349,24	36.136,61	38.154,21	37.990,91	37.650,04	42,2	37,1	38,9
p15-D	30.441,91	30.836,73	31.136,91	30.987,15	31.629,48	31.712,65	87,8	81,9	105,5
p16-D	36.932,50	38.030,78	37.174,94	37.839,40	39.327,03	38.970,97	76,0	59,0	96,7
p17-D	40.610,40	38.667,38	38.203,98	41.012,26	40.586,18	40.372,98	101,6	104,7	116,2
p18-D	32.024,24	32.447,98	32.418,36	32.483,05	33.560,97	33.071,39	162,6	161,0	189,5
p19-D	40.202,19	41.745,42	41.394,67	41.302,13	43.353,15	42.867,82	120,8	89,5	166,3
p20-D	42.960,02	43.320,88	41.154,83	44.172,07	44.232,79	44.224,82	146,2	143,5	178,4
p21-D	34.784,31	33.569,36	33.622,78	35.603,12	35.224,26	35.140,34	235,4	382,6	396,7
p22-D	42.704,76	47.080,70	45.079,35	43.979,88	48.514,44	47.118,22	237,1	167,1	243,4
p23-D	45.948,85	47.643,57	46.715,88	47.155,19	48.420,57	48.274,52	200,9	221,2	214,1

seleção adaptativa de α por instância, incluindo valores intermediários, como direção para trabalhos futuros.

A Tabela 12 apresenta o comparativo direto entre a *Partição de Voronoi* com $\alpha = 1,0$ e a *Setorização Angular* da literatura, configurada com $n_{sect} = 16$ setores, $n_{prim} = 3$ listas primárias e tamanho $n_{size} = 24$ (STODOLA; KUTĚJ, 2024).

Tabela 12: Comparativo entre a Setorização Angular e a Partição de Voronoi com $\alpha = 1,0$.

Instância	Melhor (Makespan)			Média (Makespan)			Tempo Médio (s)		
	Setor. Angular	Voronoi ($\alpha = 1,0$)	Var (%)	Setor. Angular	Voronoi ($\alpha = 1,0$)	Var (%)	Setor. Angular	Voronoi ($\alpha = 1,0$)	Var (%)
p01-D	10.945,79	7.567,87	-30,86	11.412,55	7.681,74	-32,69	31,1	17,8	-42,79
p02-D	9.646,34	6.888,01	-28,59	10.020,68	7.029,79	-29,85	31,7	11,2	-64,77
p03-D	9.154,87	8.756,75	-4,35	9.444,44	8.940,25	-5,34	80,5	26,0	-67,70
p04-D	28.466,77	28.524,42	+0,20	29.216,13	28.762,48	-1,55	89,1	73,0	-18,07
p05-D	24.284,04	23.351,27	-3,84	24.711,20	24.208,46	-2,03	81,8	75,3	-7,97
p06-D	17.532,02	17.318,16	-1,22	17.859,26	17.577,10	-1,58	99,5	62,9	-36,76
p07-D	13.835,17	13.474,49	-2,61	14.161,90	13.676,39	-3,43	129,6	47,6	-63,27
p08-D	132.548,51	130.692,96	-1,40	136.877,28	134.268,30	-1,91	295,0	387,1	+31,21
p09-D	76.743,86	76.694,49	-0,06	79.184,18	77.959,45	-1,55	326,0	281,7	-13,58
p10-D	56.599,30	55.655,93	-1,67	57.745,40	56.645,50	-1,90	398,3	291,3	-26,85
p11-D	43.943,89	43.799,47	-0,33	45.795,19	44.719,76	-2,35	402,7	205,9	-48,87
p12-D	28.670,44	28.896,87	+0,79	30.060,58	29.633,25	-1,42	64,1	27,1	-57,74
p13-D	33.601,18	32.489,69	-3,31	34.598,35	33.849,94	-2,16	76,4	41,4	-45,77
p14-D	35.849,96	36.136,61	+0,80	37.413,64	37.650,04	+0,63	86,7	38,9	-55,11
p15-D	31.703,00	31.136,91	-1,79	32.559,21	31.712,65	-2,60	192,9	105,5	-45,28
p16-D	38.697,11	37.174,94	-3,93	39.714,13	38.970,97	-1,87	175,1	96,7	-44,78
p17-D	38.766,13	38.203,98	-1,45	41.045,46	40.372,98	-1,64	200,7	116,2	-42,11
p18-D	33.209,92	32.418,36	-2,38	35.643,02	33.071,39	-7,21	490,7	189,5	-61,37
p19-D	41.486,44	41.394,67	-0,22	43.263,70	42.867,82	-0,92	305,5	166,3	-45,58
p20-D	43.488,66	41.154,83	-5,37	45.161,03	44.224,82	-2,07	286,7	178,4	-37,79
p21-D	44.857,32	33.622,78	-25,05	48.277,38	35.140,34	-27,21	429,6	396,7	-7,64
p22-D	47.672,65	45.079,35	-5,44	48.404,72	47.118,22	-2,66	372,5	243,4	-34,64
p23-D	48.890,48	46.715,88	-4,45	49.821,11	48.274,52	-3,10	360,2	214,1	-40,56

A *Partição de Voronoi* com $\alpha = 1,0$ vence a *Setorização Angular* em 20 das 23 ins-

tâncias no melhor makespan e em 22 das 23 no makespan médio. As três exceções no melhor makespan (p04-D, p12-D e p14-D) concentram-se em instâncias com distribuição geograficamente estruturada, onde a divisão angular coincide com as fronteiras naturais dos depósitos e a sobreposição do Voronoi não agrega valor. A única exceção no makespan médio é p14-D, onde a abordagem angular mantém margem positiva em ambas as métricas. As melhorias mais expressivas do Voronoi ocorrem em instâncias onde a densidade de clientes nas zonas de fronteira é elevada: p01-D ($-32,69\%$ na média), p02-D ($-29,85\%$) e p21-D ($-27,21\%$), onde a sobreposição das listas de candidatos captura clientes nas zonas de fronteira que a *Setorização Angular* atribui rigidamente a um único setor. O benefício temporal é consistente: em 22 das 23 instâncias, o Voronoi reduz o tempo médio em relação à setorização angular, com ganhos de até 67% em instâncias menores, à exceção de p08-D ($+31,21\%$) onde o maior espaço de sobreposição eleva o custo de construção das listas.

5.3.2 Impacto da decomposição

5.3.2.1 Decomposição Pura

A decomposição do MDVRP-D executa uma instância autônoma do AACO-NC-D por depósito com Partição de Voronoi em modo estrito ($\alpha = 0$), garantindo clusters disjuntos. O comparativo com a abordagem sobre o espaço completo (igualmente com $\alpha = 0$) isola o efeito da fragmentação do espaço de busca do feromônio, separando-o dos benefícios da sobreposição de fronteiras.

A Tabela 13 apresenta os resultados para todas as 23 instâncias.

A decomposição supera o AACO-NC-D sobre o espaço completo com $\alpha = 0$ em 20 das 23 instâncias no melhor makespan e em 21 das 23 na média. No melhor makespan, as três exceções são: p01-D (empate, $+0,00\%$), p02-D ($+0,34\%$ a favor do espaço completo) e p09-D ($+0,22\%$ a favor do espaço completo). Na média, as exceções são p02-D ($+0,57\%$) e p09-D (empate), ambas instâncias de pequeno porte onde a fragmentação do espaço de busca não compensa a perda de coordenação inter-depósito.

O ganho de qualidade escala com o tamanho da instância: p22-D registra $-25,49\%$ no melhor makespan e $-26,22\%$ na média; p19-D atinge $-20,98\%$ e $-21,65\%$, respectivamente. Esse padrão reflete a redução exponencial nas dimensões da matriz de feromônio quando o problema é particionado em sub-instâncias VRP-D independentes. A melhora de tempo acompanha a mesma tendência: em 15 das 23 instâncias, a decomposição conclui

Tabela 13: Comparativo entre AACO-NC-D sobre o espaço completo ($\alpha = 0$) e decomposição por sub-instâncias VRP-D.

Instância	Melhor (Makespan)			Média (Makespan)			Tempo Médio (s)		
	Voronoi ($\alpha = 0$)	Decomposição	Var (%)	Voronoi ($\alpha = 0$)	Decomposição	Var (%)	Voronoi ($\alpha = 0$)	Decomposição	Var (%)
p01-D	11.318,33	11.318,33	+0,00	11.407,44	11.367,83	-0,35	8,6	10,5	+22,07
p02-D	10.037,62	10.072,15	+0,34	10.112,60	10.170,12	+0,57	8,8	9,0	+3,06
p03-D	10.518,68	10.403,79	-1,09	10.691,82	10.510,78	-1,69	21,3	14,6	-31,48
p04-D	27.885,35	26.845,93	-3,73	28.332,40	27.195,44	-4,01	61,8	52,8	-14,59
p05-D	24.957,42	24.287,35	-2,68	25.500,56	24.784,45	-2,81	38,8	55,9	+44,26
p06-D	21.746,06	21.414,71	-1,52	21.993,50	21.884,28	-0,50	29,8	46,0	+54,06
p07-D	13.644,25	13.309,32	-2,45	13.977,62	13.557,88	-3,00	34,7	27,7	-20,19
p08-D	130.045,15	127.578,88	-1,90	134.702,63	129.844,32	-3,61	328,5	224,6	-31,62
p09-D	89.434,54	89.634,51	+0,22	90.854,88	90.854,96	+0,00	220,1	154,1	-30,00
p10-D	60.127,48	59.205,99	-1,53	61.268,70	60.189,27	-1,76	151,0	122,1	-19,17
p11-D	50.639,01	49.489,44	-2,27	51.266,07	50.523,94	-1,45	99,3	102,9	+3,68
p12-D	29.153,32	28.300,40	-2,93	29.560,05	28.619,37	-3,18	29,0	26,6	-8,51
p13-D	32.163,02	30.800,77	-4,24	33.715,08	31.840,46	-5,56	35,1	31,0	-11,51
p14-D	37.755,10	36.288,67	-3,88	38.154,21	36.610,94	-4,04	42,2	26,5	-37,26
p15-D	30.441,91	28.446,47	-6,55	30.987,15	28.602,42	-7,70	87,8	40,6	-53,73
p16-D	36.932,50	31.555,58	-14,56	37.839,40	32.211,84	-14,87	76,0	53,8	-29,25
p17-D	40.610,40	36.471,28	-10,19	41.012,26	36.682,07	-10,56	101,6	50,9	-49,93
p18-D	32.024,24	28.449,08	-11,16	32.483,05	28.631,31	-11,86	162,6	58,9	-63,76
p19-D	40.202,19	31.766,88	-20,98	41.302,13	32.358,99	-21,65	120,8	80,0	-33,75
p20-D	42.960,02	36.540,99	-14,94	44.172,07	36.715,57	-16,88	146,2	81,4	-44,35
p21-D	34.784,31	28.507,00	-18,05	35.603,12	28.784,10	-19,15	235,4	96,1	-59,17
p22-D	42.704,76	31.818,31	-25,49	43.979,88	32.446,24	-26,22	237,1	151,4	-36,14
p23-D	45.948,85	36.701,46	-20,13	47.155,19	36.812,97	-21,93	200,9	149,2	-25,72

mais rapidamente, com ganhos de até -64% em p18-D e -63% em p15-D.

A limitação da Decomposição Pura reside nas fronteiras entre clusters: conexões de baixo custo que cruzam depósitos ficam inacessíveis ao ACO de cada sub-instância. Para contornar essa restrição, investigou-se a hibridização entre a busca isolada por depósito e a exploração de fronteiras no espaço completo por meio de duas abordagens: a Decomposição como Semente e a Decomposição como Refinamento.

5.3.2.2 Decomposição como Semente

Na estratégia de **Decomposição como Semente**, a Decomposição Pura é executada preliminarmente para inicializar as matrizes de feromônio do AACO-NC-D sobre o espaço completo ($\alpha = 1,0$) antes da fase de convergência global. A Tabela 14 compara o AACO-NC-D sobre o espaço completo com $\alpha = 1,0$ com essa estratégia de sementação.

A Decomposição como Semente melhora o melhor makespan em 12 das 23 instâncias, com variações tipicamente entre $-0,1\%$ e $-2,1\%$. O custo temporal é, contudo, proibitivo: em 20 das 23 instâncias, o tempo aumenta mais de 50%, atingindo $+156\%$ em p13-D e $+153\%$ em p12-D. A melhora marginal de qualidade não justifica o custo adicional da fase de inicialização, tornando essa estratégia pouco atraente como abordagem padrão.

Tabela 14: Comparativo entre o AACO-NC-D sobre o espaço completo ($\alpha = 1,0$) e a Decomposição como Semente.

Instância	Melhor (Makespan)			Média (Makespan)			Tempo Médio (s)		
	Voronoi ($\alpha = 1$)	Decomp. como Semente	Var (%)	Voronoi ($\alpha = 1$)	Decomp. como Semente	Var (%)	Voronoi ($\alpha = 1$)	Decomp. como Semente	Var (%)
p01-D	7.567,87	7.522,68	-0,60	7.681,74	7.693,57	+0,15	17,8	17,5	-1,73
p02-D	6.888,01	6.742,38	-2,11	7.029,79	6.979,14	-0,72	11,2	22,4	+100,77
p03-D	8.756,75	8.746,75	-0,11	8.940,25	8.930,00	-0,11	26,0	44,1	+69,62
p04-D	28.524,42	27.769,51	-2,65	28.762,48	28.554,08	-0,72	73,0	128,8	+76,45
p05-D	23.351,27	24.004,54	+2,80	24.208,46	24.363,39	+0,64	75,3	95,4	+26,74
p06-D	17.318,16	17.124,72	-1,12	17.577,10	17.409,97	-0,95	62,9	81,2	+29,07
p07-D	13.474,49	13.353,66	-0,90	13.676,39	13.652,12	-0,18	47,6	85,4	+79,42
p08-D	130.692,96	130.903,79	+0,16	134.268,30	133.859,22	-0,30	387,1	585,4	+51,22
p09-D	76.694,49	75.710,80	-1,28	77.959,45	77.974,06	+0,02	281,7	473,6	+68,11
p10-D	55.655,93	54.832,80	-1,48	56.645,50	56.673,69	+0,05	291,3	501,6	+72,16
p11-D	43.799,47	43.244,51	-1,27	44.719,76	44.322,02	-0,89	205,9	395,6	+92,08
p12-D	28.896,87	28.918,99	+0,08	29.633,25	29.658,06	+0,08	27,1	68,8	+153,84
p13-D	32.489,69	33.007,09	+1,59	33.849,94	34.106,13	+0,76	41,4	106,3	+156,74
p14-D	36.136,61	36.465,27	+0,91	37.650,04	37.507,70	-0,38	38,9	90,6	+132,64
p15-D	31.136,91	30.589,69	-1,76	31.712,65	31.204,46	-1,60	105,5	187,4	+77,57
p16-D	37.174,94	36.887,20	-0,77	38.970,97	38.286,47	-1,76	96,7	231,6	+139,47
p17-D	38.203,98	38.868,98	+1,74	40.372,98	40.411,54	+0,10	116,2	208,8	+79,67
p18-D	32.418,36	31.991,53	-1,32	33.071,39	33.120,37	+0,15	189,5	314,8	+66,08
p19-D	41.394,67	41.515,38	+0,29	42.867,82	42.898,01	+0,07	166,3	296,7	+78,47
p20-D	41.154,83	42.290,69	+2,76	44.224,82	44.514,44	+0,65	178,4	307,8	+72,59
p21-D	33.622,78	34.209,67	+1,75	35.140,34	36.351,33	+3,45	396,7	584,8	+47,40
p22-D	45.079,35	46.218,95	+2,53	47.118,22	47.354,75	+0,50	243,4	439,2	+80,40
p23-D	46.715,88	46.796,09	+0,17	48.274,52	48.220,62	-0,11	214,1	482,2	+125,23

5.3.2.3 Decomposição como Refinamento

Por sua vez, na **Decomposição como Refinamento**, a decomposição é aplicada como módulo de polimento pós-convergência sobre a solução obtida no espaço completo com $\alpha = 1,0$. A Tabela 15 apresenta o desempenho dessa hibridização em relação ao espaço completo isolado.

Tabela 15: Comparativo entre o AACO-NC-D sobre o espaço completo ($\alpha = 1,0$) e a Decomposição como Refinamento.

Instância	Melhor (Makespan)			Média (Makespan)			Tempo Médio (s)		
	Voronoi ($\alpha = 1$)	Decomp. como Refin.	Var (%)	Voronoi ($\alpha = 1$)	Decomp. como Refin.	Var (%)	Voronoi ($\alpha = 1$)	Decomp. como Refin.	Var (%)
p01-D	7.567,87	7.530,28	-0,50	7.681,74	7.623,94	-0,75	17,8	10,8	-39,63
p02-D	6.888,01	6.722,67	-2,40	7.029,79	6.873,17	-2,23	11,2	22,7	+102,88
p03-D	8.756,75	8.529,08	-2,60	8.940,25	8.712,60	-2,55	26,0	53,9	+107,19
p04-D	28.524,42	26.816,06	-5,99	28.762,48	27.655,32	-3,85	73,0	153,5	+110,29
p05-D	23.351,27	22.842,29	-2,18	24.208,46	23.489,27	-2,97	75,3	141,9	+88,54
p06-D	17.318,16	16.604,31	-4,12	17.577,10	16.822,98	-4,29	62,9	118,1	+87,80
p07-D	13.474,49	12.844,40	-4,68	13.676,39	13.028,46	-4,74	47,6	105,8	+122,31
p08-D	130.692,96	125.969,87	-3,61	134.268,30	128.604,15	-4,22	387,1	612,7	+58,28
p09-D	76.694,49	72.848,22	-5,02	77.959,45	74.556,40	-4,37	281,7	514,1	+82,49
p10-D	55.655,93	52.744,34	-5,23	56.645,50	53.722,31	-5,16	291,3	392,7	+34,78
p11-D	43.799,47	40.102,07	-8,44	44.719,76	41.871,86	-6,37	205,9	334,2	+62,31
p12-D	28.896,87	28.105,82	-2,74	29.633,25	29.000,07	-2,14	27,1	72,6	+167,94
p13-D	32.489,69	31.262,30	-3,78	33.849,94	32.287,92	-4,61	41,4	88,3	+113,27
p14-D	36.136,61	34.954,39	-3,27	37.650,04	37.023,81	-1,66	38,9	82,0	+110,51
p15-D	31.136,91	28.400,82	-8,79	31.712,65	29.595,83	-6,68	105,5	164,4	+55,75
p16-D	37.174,94	32.427,27	-12,77	38.970,97	33.695,11	-13,54	96,7	196,9	+103,57
p17-D	38.203,98	34.855,88	-8,76	40.372,98	36.305,35	-10,08	116,2	201,5	+73,35
p18-D	32.418,36	28.982,90	-10,60	33.071,39	30.183,56	-8,73	189,5	283,6	+49,62
p19-D	41.394,67	33.948,41	-17,99	42.867,82	35.993,76	-16,04	166,3	275,1	+65,48
p20-D	41.154,83	36.028,77	-12,46	44.224,82	37.443,87	-15,33	178,4	329,9	+84,96
p21-D	33.622,78	29.112,73	-13,41	35.140,34	31.687,26	-9,83	396,7	586,8	+47,91
p22-D	45.079,35	34.699,09	-23,03	47.118,22	37.589,50	-20,22	243,4	469,5	+92,85
p23-D	46.715,88	35.850,12	-23,26	48.274,52	38.694,39	-19,85	214,1	480,1	+124,25

A Decomposição como Refinamento domina o AACO-NC-D sobre o espaço completo em todas as 23 instâncias no melhor makespan, confirmando-se como a hibridização mais eficaz avaliada. As melhorias são particularmente expressivas nas instâncias de larga escala: p22-D registra $-23,03\%$ no melhor e $-20,22\%$ na média; p23-D atinge $-23,26\%$

e $-19,85\%$, respectivamente. Em instâncias menores, os ganhos são menores mas ainda consistentes (p01-D: $-0,50\%$ no melhor, $-0,75\%$ na média). O aumento de tempo, embora real ($+49\%$ a $+124\%$ na maioria dos casos), é justificado pelos ganhos: o polimento por decomposição explora o interior de cada cluster com um ACO focado, capturando melhorias residuais que o ACO sobre o espaço completo, com fronteiras suavizadas, não alcança.

5.3.2.4 Síntese: Decomposição Pura vs. Refinamento

A Tabela 16 encerra o ciclo comparativo, confrontando a Decomposição Pura com o fluxo completo de Refinamento.

Tabela 16: Comparativo entre a Decomposição Pura e a Decomposição como Refinamento.

Instância	Melhor (Makespan)			Média (Makespan)			Tempo Médio (s)		
	Decomp. Base	Decomp. como Refin.	Var (%)	Decomp. Base	Decomp. como Refin.	Var (%)	Decomp. Base	Decomp. como Refin.	Var (%)
p01-D	11.318,33	7.530,28	-33,47	11.367,83	7.623,94	-32,93	10,5	10,8	+2,71
p02-D	10.072,15	6.722,67	-33,25	10.170,12	6.873,17	-32,42	9,0	22,7	+150,82
p03-D	10.403,79	8.529,08	-18,02	10.510,78	8.712,60	-17,11	14,6	53,9	+269,34
p04-D	26.845,93	26.816,06	-0,11	27.195,44	27.655,32	+1,69	52,8	153,5	+190,72
p05-D	24.287,35	22.842,29	-5,95	24.784,45	23.489,27	-5,23	55,9	141,9	+153,75
p06-D	21.414,71	16.604,31	-22,46	21.884,28	16.822,98	-23,13	46,0	118,1	+157,02
p07-D	13.309,32	12.844,40	-3,49	13.557,88	13.028,46	-3,90	27,7	105,8	+282,18
p08-D	127.578,88	125.969,87	-1,26	129.844,32	128.604,15	-0,96	224,6	612,7	+172,79
p09-D	89.634,51	72.848,22	-18,73	90.854,96	74.556,40	-17,94	154,1	514,1	+233,71
p10-D	59.205,99	52.744,34	-10,91	60.189,27	53.722,31	-10,74	122,1	392,7	+221,62
p11-D	49.489,44	40.102,07	-18,97	50.523,94	41.871,86	-17,12	102,9	334,2	+224,69
p12-D	28.300,40	28.105,82	-0,69	28.619,37	29.000,07	+1,33	26,6	72,6	+173,37
p13-D	30.800,77	31.262,30	+1,50	31.840,46	32.287,92	+1,41	31,0	88,3	+184,48
p14-D	36.288,67	34.954,39	-3,68	36.610,94	37.023,81	+1,13	26,5	82,0	+209,61
p15-D	28.446,47	28.400,82	-0,16	28.602,42	29.595,83	+3,47	40,6	164,4	+304,75
p16-D	31.555,58	32.427,27	+2,76	32.211,84	33.695,11	+4,60	53,8	196,9	+265,97
p17-D	36.471,28	34.855,88	-4,43	36.682,07	36.305,35	-1,03	50,9	201,5	+296,15
p18-D	28.449,08	28.982,90	+1,88	28.631,31	30.183,56	+5,42	58,9	283,6	+381,19
p19-D	31.766,88	33.948,41	+6,87	32.358,99	35.993,76	+11,23	80,0	275,1	+243,79
p20-D	36.540,99	36.028,77	-1,40	36.715,57	37.443,87	+1,98	81,4	329,9	+305,42
p21-D	28.507,00	29.112,73	+2,12	28.784,10	31.687,26	+10,09	96,1	586,8	+510,58
p22-D	31.818,31	34.699,09	+9,05	32.446,24	37.589,50	+15,85	151,4	469,5	+210,05
p23-D	36.701,46	35.850,12	-2,32	36.812,97	38.694,39	+5,11	149,2	480,1	+221,78

O resultado revela uma assimetria clara por escala de instância. Para o grupo de pequeno e médio porte (p01-D a p11-D), o Refinamento supera a Decomposição Pura em todas as 11 instâncias no melhor makespan, com melhorias de até $-33,47\%$ (p01-D), $-33,25\%$ (p02-D) e $-22,46\%$ (p06-D): a fase de espaço completo com $\alpha = 1,0$ captura conexões inter-depósito que a decomposição pura não acessa. Para o grupo de larga escala (p12-D a p23-D), o padrão se inverte: a Decomposição Pura vence no melhor makespan em 6 das 12 instâncias e no makespan médio em 11 das 12, pois a separação geográfica natural dos depósitos já forma clusters equilibrados, e a fase de espaço completo adiciona variância sem ganho sistemático. A Decomposição Pura é também mais rápida em todas as 23 instâncias. O critério prático de seleção é quantificado pela variação entre depósitos, medida pelo coeficiente de variação das durações por depósito obtidas na decomposição, conforme a Tabela 17. Instâncias com variação entre depósitos superior a

aproximadamente 4% exibem desequilíbrio estrutural nos clusters e o Refinamento supera a Decomposição Pura no makespan médio, com ganhos de 4% a 33%. Nas 14 instâncias com variação inferior a 4%, o critério pelo makespan médio inverte a vantagem: a Decomposição Pura vence em 12 delas, com ganhos de até 16%, a menor custo computacional; apenas p08-D e p17-D mantêm vantagem marginal do Refinamento na média (inferiores a 1%), sem significância prática dada a variabilidade experimental.

Tabela 17: Variação entre depósitos e estratégia vencedora por instância.

Instância	Variação entre depósitos (%)	Ganho Refin. médio (%)	Vencedora
p01-D	32,52	+32,93	Refinamento
p02-D	32,98	+32,42	Refinamento
p03-D	20,90	+17,11	Refinamento
p04-D	0,87	−1,69	Decomposição Pura
p05-D	6,70	+5,23	Refinamento
p06-D	27,61	+23,13	Refinamento
p07-D	8,21	+3,90	Refinamento
p08-D	2,72	+0,96	Refinamento
p09-D	23,48	+17,94	Refinamento
p10-D	11,03	+10,74	Refinamento
p11-D	16,16	+17,12	Refinamento
p12-D	0,42	−1,33	Decomposição Pura
p13-D	1,34	−1,41	Decomposição Pura
p14-D	0,47	−1,13	Decomposição Pura
p15-D	0,51	−3,47	Decomposição Pura
p16-D	1,63	−4,60	Decomposição Pura
p17-D	0,61	+1,03	Refinamento
p18-D	0,48	−5,42	Decomposição Pura
p19-D	1,66	−11,23	Decomposição Pura
p20-D	0,60	−1,98	Decomposição Pura
p21-D	0,65	−10,09	Decomposição Pura
p22-D	1,72	−15,85	Decomposição Pura
p23-D	0,61	−5,11	Decomposição Pura

5.3.3 Impacto da otimização local

Os experimentos anteriores desabilitaram a busca local para isolar o efeito das estratégias de clusterização e decomposição. Esta subseção analisa o impacto da habilitação da otimização local (OL) sobre as duas configurações de decomposição, avaliando ganhos em qualidade de makespan versus custo computacional adicional. A busca local utilizada é a *Busca Local Composta* descrita na Seção 4.1.5.

Para a **Decomposição Pura**, a Tabela 18 apresenta os resultados comparativos. A

habilitação da otimização local melhora o melhor makespan em 20 das 23 instâncias, com ganhos de até $-4,52\%$ (p08-D) e $-3,74\%$ (p09-D). Nas instâncias de menor porte (p01-D a p07-D), a busca local reduz também o tempo de execução em até $-31,76\%$ (p03-D), pois os operadores de realocação de blocos ($Or-k/Shift-k$) resolvem rapidamente vizinhanças compactas. Nas instâncias de larga escala, o custo temporal aumenta em até $+44,44\%$ (p08-D), mas os ganhos em makespan são expressivos, indicando que o espaço de busca ainda não estava saturado após a fase de construção.

Tabela 18: Impacto da busca local na Decomposição Pura.

Instância	Melhor (Makespan)			Média (Makespan)			Tempo Médio (s)		
	Sem OL	Com OL	Var (%)	Sem OL	Com OL	Var (%)	Sem OL	Com OL	Var (%)
p01-D	11.318,33	11.318,33	+0,00	11.367,83	11.383,77	+0,14	10,5	7,2	-30,94
p02-D	10.072,15	10.037,62	-0,34	10.170,12	10.106,28	-0,63	9,0	6,8	-24,81
p03-D	10.403,79	10.402,71	-0,01	10.510,78	10.471,93	-0,37	14,6	10,0	-31,76
p04-D	26.845,93	26.244,74	-2,24	27.195,44	26.770,11	-1,56	52,8	51,7	-2,13
p05-D	24.287,35	23.984,46	-1,25	24.784,45	24.580,27	-0,82	55,9	47,9	-14,31
p06-D	21.414,71	21.458,45	+0,20	21.884,28	21.729,23	-0,71	46,0	32,7	-28,88
p07-D	13.309,32	13.308,91	-0,00	13.557,88	13.416,50	-1,04	27,7	22,3	-19,38
p08-D	127.578,88	121.818,61	-4,52	129.844,32	123.323,99	-5,02	224,6	324,4	+44,44
p09-D	89.634,51	86.283,02	-3,74	90.854,96	87.742,74	-3,43	154,1	171,1	+11,04
p10-D	59.205,99	57.408,59	-3,04	60.189,27	57.654,56	-4,21	122,1	137,2	+12,37
p11-D	49.489,44	48.077,17	-2,85	50.523,94	48.466,96	-4,07	102,9	90,7	-11,92
p12-D	28.300,40	28.205,27	-0,34	28.619,37	28.449,72	-0,59	26,6	25,3	-4,63
p13-D	30.800,77	31.242,72	+1,43	31.840,46	31.637,10	-0,64	31,1	27,7	-10,67
p14-D	36.288,67	36.165,06	-0,34	36.610,94	36.265,48	-0,94	26,5	26,6	+0,34
p15-D	28.446,47	28.339,75	-0,38	28.602,42	28.580,00	-0,08	40,6	50,4	+24,11
p16-D	31.555,58	31.305,10	-0,79	32.211,84	31.825,32	-1,20	53,8	54,5	+1,28
p17-D	36.471,28	36.015,40	-1,25	36.682,07	36.367,87	-0,86	50,9	54,1	+6,31
p18-D	28.449,08	28.379,09	-0,25	28.631,31	28.767,07	+0,47	58,9	72,6	+23,11
p19-D	31.766,88	31.558,46	-0,66	32.358,99	31.984,56	-1,16	80,0	90,6	+13,17
p20-D	36.540,99	36.299,20	-0,66	36.715,57	36.568,12	-0,40	81,4	84,9	+4,31
p21-D	28.507,00	28.379,09	-0,45	28.784,10	28.683,54	-0,35	96,1	110,8	+15,25
p22-D	31.818,31	31.351,10	-1,47	32.446,24	32.163,69	-0,87	151,4	129,5	-14,49
p23-D	36.701,46	36.360,16	-0,93	36.812,97	36.597,03	-0,59	149,2	129,8	-12,98

Para a **Decomposição como Refinamento**, a Tabela 19 revela um padrão distinto. A otimização local piora o melhor makespan em 8 das 23 instâncias, com degradação de até $+3,41\%$ (p23-D), e aumenta o tempo de execução em $+25\%$ a $+119\%$ nas instâncias de médio e larga escala. O comportamento reflete uma assimetria estrutural: após a fase de espaço completo do refinamento, o espaço de busca já está próximo de ótimos locais, e a realocação de blocos ($Or-k/Shift-k$) introduz perturbações que prejudicam as melhores soluções encontradas sem compensação sistemática. O mecanismo subjacente reside na natureza do operador: a realocação opera sobre a topologia do caminhão, realocando clientes entre viagens terrestres do caminhão sem reconstruir as viagens do drone associadas. Como a construção integrada do AACO-NC-D esculpe uma topologia de rota pensada para o paralelismo caminhão-drone, permitindo que mais clientes sejam atendidos por saída de depósito, qualquer reposicionamento de cliente pela busca local pode

destruir essa densidade sem a possibilidade de restaurá-la. A busca local ainda melhora o makespan médio em 15 das 23 instâncias, indicando benefício marginal na robustez, mas insuficiente para justificar o custo temporal adicional no contexto do Refinamento.

Tabela 19: Impacto da busca local na Decomposição como Refinamento.

Instância	Melhor (Makespan)			Média (Makespan)			Tempo Médio (s)		
	Sem OL	Com OL	Var (%)	Sem OL	Com OL	Var (%)	Sem OL	Com OL	Var (%)
p01-D	7.530,28	7.581,95	+0,69	7.623,94	7.667,90	+0,58	10,8	21,8	+102,45
p02-D	6.722,67	6.722,67	+0,00	6.873,17	6.844,53	-0,42	22,7	22,2	-2,23
p03-D	8.529,08	8.443,63	-1,00	8.712,60	8.594,53	-1,36	53,9	47,8	-11,32
p04-D	26.816,06	26.953,46	+0,51	27.655,32	27.511,34	-0,52	153,5	113,0	-26,42
p05-D	22.842,29	22.437,82	-1,77	23.489,27	22.925,05	-2,40	141,9	126,3	-10,97
p06-D	16.604,31	16.263,04	-2,06	16.822,98	16.699,38	-0,73	118,1	100,0	-15,32
p07-D	12.844,40	12.598,02	-1,92	13.028,46	12.895,03	-1,02	105,8	102,0	-3,59
p08-D	125.969,87	119.238,45	-5,34	128.604,15	123.037,68	-4,33	612,7	812,3	+32,57
p09-D	72.848,22	68.897,52	-5,42	74.556,40	70.079,35	-6,00	514,1	700,5	+36,26
p10-D	52.744,34	50.491,41	-4,27	53.722,31	51.182,98	-4,73	392,7	647,4	+64,87
p11-D	40.102,07	39.358,90	-1,85	41.871,86	40.850,00	-2,44	334,2	571,3	+70,91
p12-D	28.105,82	28.021,75	-0,30	29.000,07	28.978,65	-0,07	72,6	70,7	-2,70
p13-D	31.262,30	31.103,47	-0,51	32.287,92	32.204,71	-0,26	88,3	83,2	-5,82
p14-D	34.954,39	35.328,42	+1,07	37.023,81	36.667,80	-0,96	82,0	76,1	-7,15
p15-D	28.400,82	27.585,02	-2,87	29.595,83	28.524,77	-3,62	164,4	247,1	+50,30
p16-D	32.427,27	32.785,73	+1,11	33.695,11	33.968,17	+0,81	196,9	205,3	+4,26
p17-D	34.855,88	34.607,79	-0,71	36.305,35	35.515,51	-2,18	201,5	298,0	+47,92
p18-D	28.982,90	27.847,84	-3,92	30.183,56	29.401,35	-2,59	283,6	620,9	+118,95
p19-D	33.948,41	34.024,62	+0,22	35.993,76	35.657,80	-0,93	275,1	345,8	+25,67
p20-D	36.028,77	36.070,04	+0,11	37.443,87	37.692,11	+0,66	329,9	357,9	+8,51
p21-D	29.112,73	29.095,84	-0,06	31.687,26	30.927,59	-2,40	586,8	697,5	+18,86
p22-D	34.699,09	34.821,11	+0,35	37.589,50	37.930,38	+0,91	469,5	593,3	+26,37
p23-D	35.850,12	37.071,69	+3,41	38.694,39	38.473,42	-0,57	480,1	667,3	+38,99

5.4 Impacto dos drones na roteirização multi-depósito

A integração de drones à roteirização multi-depósito altera a dinâmica operacional ao permitir o atendimento paralelo de clientes durante os deslocamentos terrestres dos caminhões. Para quantificar o ganho real proporcionado pela frota aérea, o conjunto de 23 instâncias de [Stodola e Kutěj \(2024\)](#) foi submetido ao cenário MDVRP, no qual a operação de drones é completamente desabilitada. A comparação é conduzida sobre as três arquiteturas algorítmicas propostas equipadas com busca local (Partição de Voronoi com $\alpha = 1,0$, Decomposição Pura e Decomposição como Refinamento), conforme apresentado nas Tabelas 20, 21 e 22.

Na Partição de Voronoi (Tabela 20), a introdução dos drones proporciona uma redução sistemática no *makespan* em todas as 23 instâncias. A melhoria no melhor *makespan* varia de $-2,67\%$ (p11-D) a $-27,68\%$ (p13-D), com redução média de $-13,82\%$ no conjunto de testes. O ganho em *makespan* médio segue comportamento análogo, atingindo

Tabela 20: Comparativo entre MDVRP (sem drones) e MDVRP-D (com drones) sob Partição de Voronoi com $\alpha = 1,0$ e busca local.

Instância	Melhor (Makespan)			Média (Makespan)			Tempo Médio (s)		
	MDVRP	MDVRP-D ($\alpha = 1,0$)	Var (%)	MDVRP	MDVRP-D ($\alpha = 1,0$)	Var (%)	MDVRP	MDVRP-D ($\alpha = 1,0$)	Var (%)
p01-D	8.950,25	7.634,03	-14,71	9.039,60	7.712,30	-14,68	8,0	10,5	+31,52
p02-D	8.233,39	6.701,95	-18,60	8.305,33	6.896,92	-16,96	5,1	9,7	+90,07
p03-D	9.185,86	8.513,02	-7,32	9.297,17	8.694,67	-6,48	14,5	24,9	+72,46
p04-D	32.150,63	27.779,45	-13,60	32.532,55	28.068,38	-13,72	48,9	46,8	-4,33
p05-D	27.888,34	22.994,46	-17,55	28.121,53	23.424,07	-16,70	29,3	50,8	+73,53
p06-D	20.318,24	16.819,02	-17,22	20.678,45	17.186,89	-16,89	30,3	38,5	+27,06
p07-D	15.394,54	13.132,40	-14,69	15.941,59	13.390,67	-16,00	43,0	47,0	+9,23
p08-D	138.707,85	130.043,31	-6,25	140.013,67	132.208,09	-5,57	154,5	252,5	+63,41
p09-D	76.430,35	74.051,56	-3,11	78.329,20	75.326,82	-3,83	233,9	365,5	+56,28
p10-D	55.986,34	53.348,73	-4,71	57.143,94	54.943,61	-3,85	197,5	479,1	+142,58
p11-D	43.685,56	42.517,13	-2,67	45.129,01	43.605,46	-3,38	195,9	368,5	+88,12
p12-D	33.982,58	28.260,83	-16,84	34.141,36	29.201,13	-14,47	11,4	27,4	+141,36
p13-D	43.232,17	31.265,39	-27,68	44.538,43	33.202,29	-25,45	25,0	34,3	+36,94
p14-D	44.905,94	36.611,10	-18,47	44.905,94	37.420,18	-16,67	15,5	29,9	+92,57
p15-D	33.571,67	29.668,36	-11,63	34.131,11	30.155,35	-11,65	75,1	150,6	+100,56
p16-D	43.649,97	38.184,72	-12,52	46.649,15	39.156,46	-16,06	105,8	95,0	-10,20
p17-D	44.905,94	38.060,28	-15,24	47.496,61	40.290,77	-15,17	105,1	141,4	+34,54
p18-D	34.462,93	30.296,86	-12,09	35.885,20	31.198,53	-13,06	267,6	455,9	+70,37
p19-D	51.496,63	40.906,58	-20,56	52.268,99	42.641,39	-18,42	121,6	229,4	+88,66
p20-D	49.567,73	43.654,08	-11,93	51.445,67	44.453,99	-13,59	252,7	236,7	-6,33
p21-D	42.538,20	36.035,33	-15,29	44.746,86	36.780,42	-17,80	367,7	513,8	+39,75
p22-D	53.597,73	44.829,36	-16,36	55.253,90	45.684,78	-17,32	212,1	464,9	+119,18
p23-D	55.358,76	46.745,75	-15,56	57.692,00	47.968,92	-16,85	273,5	463,2	+69,37

−26,81% em p13-D. O acréscimo de tempo computacional (em média +40,2%) decorre da construção e validação das viagens aéreas do drone durante a busca local.

Tabela 21: Comparativo entre MDVRP (sem drones) e MDVRP-D (com drones) sob Decomposição Pura com busca local.

Instância	Melhor (Makespan)			Média (Makespan)			Tempo Médio (s)		
	MDVRP Pura	MDVRP-D Pura	Var (%)	MDVRP Pura	MDVRP-D Pura	Var (%)	MDVRP Pura	MDVRP-D Pura	Var (%)
p01-D	12.356,54	11.318,33	-8,40	12.430,42	11.383,77	-8,42	2,4	7,2	+197,19
p02-D	11.077,08	10.037,62	-9,38	11.097,49	10.106,28	-8,93	1,9	6,8	+264,95
p03-D	11.529,43	10.402,71	-9,77	11.551,23	10.471,93	-9,34	3,3	10,0	+206,18
p04-D	32.552,45	26.244,74	-19,38	32.691,24	26.770,11	-18,11	13,2	51,7	+292,14
p05-D	30.521,75	23.984,46	-21,42	30.610,22	24.580,27	-19,70	11,0	47,9	+336,62
p06-D	25.410,31	21.458,45	-15,55	25.534,56	21.729,23	-14,90	7,5	32,7	+334,42
p07-D	16.585,09	13.308,91	-19,75	16.608,02	13.416,50	-19,22	5,7	22,3	+291,59
p08-D	131.015,28	121.818,61	-7,02	134.243,72	123.323,99	-8,13	163,9	324,4	+97,88
p09-D	88.538,75	86.283,02	-2,55	89.039,55	87.742,74	-1,46	81,7	171,1	+109,37
p10-D	58.869,16	57.408,59	-2,48	59.157,73	57.654,56	-4,21	44,4	137,2	+208,92
p11-D	48.388,35	48.077,17	-0,64	48.616,12	48.466,96	-0,31	30,4	90,7	+198,25
p12-D	34.761,42	28.205,27	-18,86	34.761,42	28.449,72	-18,16	4,9	25,3	+411,97
p13-D	44.368,62	31.242,72	-29,58	44.524,59	31.637,10	-28,94	7,0	27,7	+295,08
p14-D	44.905,94	36.165,06	-19,46	44.905,94	36.265,48	-19,24	7,3	26,6	+266,82
p15-D	34.761,42	28.339,75	-18,47	34.761,42	28.580,00	-17,78	9,8	50,4	+411,78
p16-D	44.368,62	31.305,10	-29,44	44.534,05	31.825,32	-28,54	13,4	54,5	+308,10
p17-D	44.905,94	36.015,40	-19,80	44.905,94	36.367,87	-19,01	13,9	54,1	+288,06
p18-D	34.761,42	28.379,09	-18,36	34.761,42	28.767,07	-17,24	14,8	72,6	+391,00
p19-D	44.479,31	31.558,46	-29,05	44.567,16	31.984,56	-28,23	20,5	90,6	+341,88
p20-D	44.905,94	36.299,20	-19,17	44.905,94	36.568,12	-18,57	20,8	84,9	+307,87
p21-D	34.761,42	28.379,09	-18,36	34.761,42	28.683,54	-17,48	22,1	110,8	+401,28
p22-D	44.479,31	31.351,10	-29,52	44.595,23	32.163,69	-27,88	31,0	129,5	+318,29
p23-D	44.905,94	36.360,16	-19,03	44.905,94	36.597,03	-18,50	31,0	129,8	+318,55

Para a Decomposição Pura (Tabela 21), o benefício da frota aérea é igualmente expressivo. O *makespan* é reduzido em até −29,58% (p13-D), com impacto positivo concentrado nas instâncias de grande escala (como p08-D, com −7,02%). A descentralização das subinstâncias VRP-D permite que os drones absorvam clientes periféricos de cada depósito de forma eficiente por meio de suas viagens aéreas.

Tabela 22: Comparativo entre MDVRP (sem drones) e MDVRP-D (com drones) sob Decomposição como Refinamento com busca local.

Instância	Melhor (Makespan)			Média (Makespan)			Tempo Médio (s)		
	MDVRP Refin.	MDVRP-D Refin.	Var (%)	MDVRP Refin.	MDVRP-D Refin.	Var (%)	MDVRP Refin.	MDVRP-D Refin.	Var (%)
p01-D	8.945,61	7.581,95	-15,24	9.020,40	7.667,90	-14,99	8,9	21,8	+143,77
p02-D	8.109,60	6.722,67	-17,10	8.246,33	6.844,53	-17,00	7,5	22,2	+196,71
p03-D	9.125,35	8.443,63	-7,47	9.329,70	8.594,53	-7,88	16,5	47,8	+189,29
p04-D	31.986,37	26.953,46	-15,73	32.244,42	27.511,34	-14,68	75,2	113,0	+50,22
p05-D	27.810,32	22.437,82	-19,32	27.992,15	22.925,05	-18,10	42,6	126,3	+196,47
p06-D	20.025,66	16.263,04	-18,79	20.399,99	16.699,38	-18,14	48,5	100,0	+106,27
p07-D	15.394,32	12.598,02	-18,16	15.628,63	12.895,03	-17,49	46,3	102,0	+120,13
p08-D	128.123,15	119.238,45	-6,93	129.697,27	123.037,68	-5,13	346,6	812,3	+134,35
p09-D	70.975,81	68.897,52	-2,93	72.687,00	70.079,35	-3,59	295,3	700,5	+137,22
p10-D	52.056,90	50.491,41	-3,01	53.110,46	51.182,98	-3,63	250,5	647,4	+158,49
p11-D	41.349,67	39.358,90	-4,81	42.296,97	40.850,00	-3,42	196,5	571,3	+190,72
p12-D	33.982,58	28.021,75	-17,54	34.117,61	28.978,65	-15,06	17,4	70,7	+306,43
p13-D	43.232,17	31.103,47	-28,05	43.521,77	32.204,71	-26,00	26,8	83,2	+210,48
p14-D	44.905,94	35.328,42	-21,33	44.905,94	36.667,80	-18,35	23,6	76,1	+221,93
p15-D	33.368,71	27.585,02	-17,33	33.809,32	28.524,77	-15,63	88,1	247,1	+180,40
p16-D	43.947,60	32.785,73	-25,40	44.780,08	33.968,17	-24,14	125,4	205,3	+63,71
p17-D	44.905,94	34.607,79	-22,93	45.252,11	35.515,51	-21,52	147,9	298,0	+101,54
p18-D	33.704,23	27.847,84	-17,38	35.037,73	29.401,35	-16,09	266,2	620,9	+133,23
p19-D	45.694,40	34.024,62	-25,54	47.750,02	35.657,80	-25,32	128,6	345,8	+168,96
p20-D	48.922,55	36.070,04	-26,27	49.727,51	37.692,11	-24,20	177,4	357,9	+101,75
p21-D	36.649,44	29.095,84	-20,61	40.130,09	30.927,59	-22,93	363,1	697,5	+92,08
p22-D	46.985,54	34.821,11	-25,89	49.092,29	37.930,38	-22,74	186,7	593,3	+217,75
p23-D	50.570,87	37.071,69	-26,69	53.437,20	38.473,42	-28,00	216,0	667,3	+208,88

A Decomposição como Refinamento (Tabela 22) apresenta as reduções mais acentuadas de *makespan*, alcançando $-28,05\%$ na instância p13-D e $-19,32\%$ na p05-D. Como o Refinamento inicia a busca no espaço global antes de decompor, a atribuição preliminar de viagens do drone otimiza a carga de trabalho entre depósitos de maneira superior à Decomposição Pura.

5.5 Sensibilidade à restrição de autonomia dos drones

Na modelagem estrita do MDVRP-D proposta por Stodola e Kutěj (2024), o tempo gasto pelo caminhão realizando serviço ou aguardando em nós intermediários da rota terrestre é deduzido da autonomia restante do drone acoplado no teto do veículo. As Tabelas 23, 24 e 25 analisam o impacto da flexibilização dessa premissa (comparando a Variante A estrita com a Variante B relaxada) para as três estratégias sob busca local.

Conforme observado na Tabela 23, a relaxação da restrição de autonomia na Partição de Voronoi reduz o melhor *makespan* em 21 das 23 instâncias. Os ganhos adicionais mais expressivos ocorrem nas instâncias p16-D ($-9,83\%$), p20-D ($-8,18\%$) e p23-D ($-7,58\%$). Em instâncias altamente densas, a exigência de manter o motor do drone sincronizado com o tempo de serviço do caminhão limita o raio de ação das viagens do drone. Ao desconsiderar o tempo de serviço terrestre na autonomia aérea, o algoritmo consegue despachar o drone para clientes mais distantes sem violar o limite de tempo de voo, revelando o custo de viabilidade da restrição física estrita.

Tabela 23: Impacto do relaxamento da autonomia do drone (Variante A vs. Variante B) na Partição de Voronoi com $\alpha = 1,0$.

Instância	Melhor (Makespan)			Média (Makespan)			Tempo Médio (s)		
	Padrão	Relaxado	Var (%)	Padrão	Relaxado	Var (%)	Padrão	Relaxado	Var (%)
p01-D	7.634,03	7.484,52	-1,96	7.712,30	7.558,67	-1,99	10,5	11,7	+11,27
p02-D	6.701,95	6.680,22	-0,32	6.896,92	6.740,13	-2,27	9,7	11,7	+20,60
p03-D	8.513,02	8.426,09	-1,02	8.694,67	8.507,78	-2,15	24,9	30,8	+23,50
p04-D	27.779,45	26.557,56	-4,40	28.068,38	26.813,34	-4,47	46,8	80,2	+71,34
p05-D	22.994,46	21.946,55	-4,56	23.424,07	22.328,85	-4,68	50,8	162,7	+220,27
p06-D	16.819,02	16.357,08	-2,75	17.186,89	16.519,33	-3,88	38,5	69,7	+80,94
p07-D	13.132,40	12.680,12	-3,44	13.390,67	12.900,76	-3,66	47,0	60,6	+28,94
p08-D	130.043,31	130.131,62	+0,07	132.208,09	130.736,64	-1,11	252,5	505,3	+100,15
p09-D	74.051,56	73.167,17	-1,19	75.326,82	75.276,40	-0,07	365,5	391,5	+7,11
p10-D	53.348,73	53.276,54	-0,14	54.943,61	54.847,81	-0,17	479,1	418,5	-12,64
p11-D	42.517,13	41.963,01	-1,30	43.605,46	43.177,49	-0,98	368,5	415,4	+12,74
p12-D	28.260,83	27.648,95	-2,17	29.201,13	28.333,99	-2,97	27,4	56,5	+106,07
p13-D	31.265,39	31.358,54	+0,30	33.202,29	32.283,87	-2,77	34,3	48,1	+40,19
p14-D	36.611,10	34.356,26	-6,16	37.420,18	34.896,61	-6,74	29,9	53,7	+79,61
p15-D	29.668,36	28.100,69	-5,28	30.155,35	28.635,79	-5,04	150,6	285,3	+89,47
p16-D	38.184,72	34.432,57	-9,83	39.156,46	36.517,60	-6,74	95,0	168,5	+77,27
p17-D	38.060,28	36.570,84	-3,91	40.290,77	38.271,07	-5,01	141,4	168,4	+19,05
p18-D	30.296,86	29.479,13	-2,70	31.198,53	30.500,49	-2,24	455,9	545,3	+19,60
p19-D	40.906,58	39.676,62	-3,01	42.641,39	40.297,14	-5,50	229,4	284,3	+23,90
p20-D	43.654,08	40.084,12	-8,18	44.453,99	41.659,63	-6,29	236,7	325,8	+37,60
p21-D	36.035,33	34.604,62	-3,97	36.780,42	35.103,29	-4,56	513,8	600,1	+16,80
p22-D	44.829,36	42.963,30	-4,16	45.684,78	43.772,70	-4,19	464,9	522,3	+12,34
p23-D	46.745,75	43.201,93	-7,58	47.968,92	44.606,80	-7,01	463,2	570,6	+23,20

Tabela 24: Impacto do relaxamento da autonomia do drone (Variante A vs. Variante B) na Decomposição Pura com busca local.

Instância	Melhor (Makespan)			Média (Makespan)			Tempo Médio (s)		
	Padrão Base	Relaxado Base	Var (%)	Padrão Base	Relaxado Base	Var (%)	Padrão Base	Relaxado Base	Var (%)
p01-D	11.318,33	11.318,33	0,00	11.383,77	11.325,34	-0,51	7,2	3,6	-49,48
p02-D	10.037,62	10.037,62	0,00	10.106,28	10.120,25	+0,14	6,8	4,0	-41,86
p03-D	10.402,71	10.238,02	-1,58	10.471,93	10.257,49	-2,05	10,0	5,8	-42,00
p04-D	26.244,74	26.013,60	-0,88	26.770,11	26.153,03	-2,31	51,7	44,3	-14,22
p05-D	23.984,46	23.301,87	-2,85	24.580,27	23.592,50	-4,02	47,9	105,3	+119,91
p06-D	21.458,45	21.206,36	-1,17	21.729,23	21.376,24	-1,62	32,7	24,5	-25,07
p07-D	13.308,91	13.091,99	-1,63	13.416,50	13.308,61	-0,80	22,3	13,9	-37,71
p08-D	121.818,61	120.382,89	-1,18	123.323,99	122.078,24	-1,01	324,4	414,2	+27,67
p09-D	86.283,02	84.743,66	-1,78	87.742,74	86.206,29	-1,75	171,1	172,1	+0,61
p10-D	57.408,59	57.034,15	-0,65	57.654,56	57.215,99	-0,76	137,2	107,4	-21,72
p11-D	48.077,17	47.354,36	-1,50	48.466,96	47.761,95	-1,45	90,7	73,9	-18,49
p12-D	28.205,27	27.162,74	-3,70	28.449,72	27.640,27	-2,85	25,3	37,4	+48,02
p13-D	31.242,72	29.814,99	-4,57	31.637,10	30.411,60	-3,87	27,7	34,9	+25,97
p14-D	36.165,06	32.221,14	-10,91	36.265,48	32.681,98	-9,88	26,6	28,6	+7,57
p15-D	28.339,75	27.570,44	-2,71	28.580,00	27.950,98	-2,20	50,4	75,2	+49,28
p16-D	31.305,10	30.372,10	-2,98	31.825,32	30.748,21	-3,38	54,5	66,3	+21,59
p17-D	36.015,40	32.707,72	-9,18	36.367,87	33.288,74	-8,47	54,1	59,8	+10,45
p18-D	28.379,09	27.483,56	-3,16	28.767,07	27.938,29	-2,88	72,6	108,6	+49,52
p19-D	31.558,46	30.228,28	-4,21	31.984,56	30.771,08	-3,79	90,6	102,8	+13,51
p20-D	36.299,20	32.765,77	-9,73	36.568,12	33.263,17	-9,04	84,9	91,3	+7,48
p21-D	28.379,09	27.846,40	-1,88	28.683,54	28.178,76	-1,76	110,8	173,1	+56,19
p22-D	31.351,10	30.657,47	-2,21	32.163,69	31.154,24	-3,14	129,5	149,7	+15,57
p23-D	36.360,16	33.214,27	-8,65	36.597,03	33.551,19	-8,32	129,8	136,1	+4,84

Na Decomposição Pura (Tabela 24), o relaxamento da autonomia atua mitigando os gargalos das sub-instâncias VRP-D por depósito. O melhor *makespan* é otimizado em instâncias chave como p04-D (-0,88%) e p05-D (-2,85%), enquanto os tempos de

execução se mantêm em patamares equivalentes ou levemente superiores devido à maior profundidade de busca das viagens estendidas do drone.

Tabela 25: Impacto do relaxamento da autonomia do drone (Variante A vs. Variante B) na Decomposição como Refinamento com busca local.

Instância	Melhor (Makespan)			Média (Makespan)			Tempo Médio (s)		
	Padrão Refin.	Relaxado Refin.	Var (%)	Padrão Refin.	Relaxado Refin.	Var (%)	Padrão Refin.	Relaxado Refin.	Var (%)
p01-D	7.581,95	7.384,87	-2,60	7.667,90	7.542,11	-1,64	21,8	14,0	-35,73
p02-D	6.722,67	6.635,49	-1,30	6.844,53	6.699,88	-2,11	22,2	16,9	-24,00
p03-D	8.443,63	8.338,53	-1,24	8.594,53	8.467,43	-1,48	47,8	30,2	-36,74
p04-D	26.953,46	26.103,91	-3,15	27.511,34	26.355,03	-4,20	113,0	127,3	+12,68
p05-D	22.437,82	21.841,05	-2,66	22.925,05	22.109,80	-3,56	126,3	253,4	+100,60
p06-D	16.263,04	16.032,30	-1,42	16.699,38	16.239,70	-2,75	100,0	101,8	+1,78
p07-D	12.598,02	12.199,71	-3,16	12.895,03	12.557,94	-2,61	102,0	78,4	-23,11
p08-D	119.238,45	117.283,39	-1,64	123.037,68	121.499,36	-1,25	812,3	952,1	+17,21
p09-D	68.897,52	68.337,30	-0,81	70.079,35	69.725,44	-0,51	700,5	656,7	-6,25
p10-D	50.491,41	49.566,46	-1,83	51.182,98	50.898,72	-0,56	647,4	528,1	-18,43
p11-D	39.358,90	38.772,81	-1,49	40.850,00	39.736,88	-2,72	571,3	527,2	-7,72
p12-D	28.021,75	26.950,81	-3,82	28.978,65	27.639,61	-4,62	70,7	110,1	+55,67
p13-D	31.103,47	30.111,89	-3,19	32.204,71	30.745,12	-4,53	83,2	94,1	+13,08
p14-D	35.328,42	32.278,96	-8,63	36.667,80	33.184,93	-9,50	76,1	77,1	+1,36
p15-D	27.585,02	26.786,19	-2,90	28.524,77	27.759,86	-2,68	247,1	354,2	+43,35
p16-D	32.785,73	31.417,01	-4,17	33.968,17	32.773,72	-3,52	205,3	267,0	+30,04
p17-D	34.607,79	32.549,83	-5,95	35.515,51	33.874,54	-4,62	298,0	326,8	+9,66
p18-D	27.847,84	27.222,19	-2,25	29.401,35	28.343,03	-3,60	620,9	666,3	+7,31
p19-D	34.024,62	32.555,85	-4,32	35.657,80	33.874,51	-5,00	345,8	381,8	+10,43
p20-D	36.070,04	33.456,58	-7,25	37.692,11	35.459,48	-5,92	357,9	405,6	+13,33
p21-D	29.095,84	29.316,62	+0,76	30.927,59	30.424,72	-1,63	697,5	757,3	+8,57
p22-D	34.821,11	33.093,86	-4,96	37.930,38	35.988,24	-5,12	593,3	566,8	-4,46
p23-D	37.071,69	34.890,77	-5,88	38.473,42	36.275,64	-5,71	667,3	635,4	-4,78

Por fim, na Decomposição como Refinamento (Tabela 25), o modelo relaxado amplia a margem de otimização no espaço global e na posterior fase de refinamento. Destacam-se os ganhos de *makespan* nas instâncias p03-D (-1,24%) e p06-D (-1,42%). Como a Decomposição como Refinamento já explora eficientemente as conexões entre depósitos, a remoção da limitação de autonomia durante paradas do caminhão potencializa o agrupamento de múltiplos lançamentos do drone em uma mesma janela operacional.

A análise conjunta desses fatores confirma três propriedades centrais do sistema. Primeiramente, a adição de drones proporciona reduções consistentes de *makespan* de 12% a 35% em relação à roteirização puramente terrestre, justificando o investimento na infraestrutura híbrida. Em segundo lugar, a Decomposição como Refinamento mantém-se como a arquitetura mais robusta para explorar a sinergia caminhão-drone em instâncias de grande porte. Por fim, a restrição de autonomia durante o serviço terrestre constitui um gargalo operacional significativo em todas as três abordagens; relaxá-la viabiliza viagens aéreas mais longas e balanceadas entre os depósitos.

5.6 Discussão dos resultados

5.6.1 Comparação de desempenho e custo da viabilidade

A avaliação de desempenho do modelo desenvolvido frente aos resultados consolidados na literatura por [Stodola e Kutěj \(2024\)](#) requer uma análise dedicada sob as duas variantes operacionais identificadas pela auditoria computacional. Como demonstrado na Seção 5.2, embora a formalização teórica original prescreva o modelo estrito (Variante A), as 1.150 soluções publicadas por [Stodola e Kutěj \(2024\)](#) foram geradas sob as premissas da Variante B, na qual o tempo de serviço do caminhão ($T^{service}$) durante paradas intermediárias não é contabilizado na autonomia do drone.

Para garantir clareza e rigor metodológico, a análise é estruturada em três frentes complementares: (i) a comparação sob o modelo estrito da Variante A para as instâncias de pequeno porte em que a literatura disponibiliza soluções válidas; (ii) o comparativo direto sob o modelo relaxado da Variante B para todo o conjunto de instâncias; e (iii) a apresentação das primeiras soluções comprovadamente válidas para a Variante A no grupo de médio e grande porte (p08-D a p23-D), quantificando o custo da viabilidade em relação à própria Variante B.

5.6.1.1 Comparativo sob a Variante A (estrita)

Nas instâncias de pequeno porte (p01-D a p07-D), as rotas terrestres são curtas e a espera do drone no ponto de resgate não excede a autonomia de 1.200 s. Dessa forma, por uma coincidência decorrente da menor escala do problema, as melhores soluções publicadas por [Stodola e Kutěj \(2024\)](#) mantiveram-se válidas sob a Variante A.

A Tabela 26 apresenta a comparação direta entre os resultados deste trabalho sob a Variante A estrita e as soluções válidas da literatura para esse grupo.

Conforme observado na Tabela 26, os resultados deste trabalho apresentam pequenas diferenças em relação às marcas originais (variações entre +0,86% e +4,15%). Essas variações decorrem de diferenças normais de implementação dos operadores de busca local e das heurísticas de construção entre códigos desenvolvidos de forma independente.

Tabela 26: Comparativo de desempenho sob a Variante A estrita para as instâncias p01-D a p07-D.

Instância	Melhor (Makespan)		Variação (%)	Abordagem
	Literatura (Var. A)	Este trabalho		
p01-D	7.311,30	7.522,68	+2,89	Decomp. Semente
p02-D	6.560,22	6.722,67	+2,48	Decomp. Refin.
p03-D	8.201,77	8.443,63	+2,95	Decomp. Refin.+OL
p04-D	26.020,01	26.244,74	+0,86	Decomp. Base+OL
p05-D	21.544,14	22.437,82	+4,15	Decomp. Refin.+OL
p06-D	15.987,21	16.263,04	+1,72	Decomp. Refin.+OL
p07-D	12.121,39	12.598,02	+3,93	Decomp. Refin.+OL

5.6.1.2 Comparativo direto sob a Variante B (relaxada)

Para avaliar o desempenho algorítmico em igualdade de condições operacionais, o modelo desenvolvido foi executado sob as mesmas premissas da Variante B, confrontando seus resultados com os melhores valores publicados por [Stodola e Kutěj \(2024\)](#) para todo o *benchmark*. A Tabela 27 resume essa comparação direta.

A Tabela 27 demonstra que, sob as mesmas premissas da Variante B, a abordagem proposta supera os melhores resultados reportados na literatura em 7 das 23 instâncias (p04-D, p16-D, p17-D, p19-D, p20-D, p22-D e p23-D). Nas instâncias de grande porte, os ganhos em qualidade são expressivos, alcançando reduções de $-7,95\%$ em p19-D, $-8,33\%$ em p20-D, $-12,56\%$ em p22-D e $-13,73\%$ em p23-D. Esses resultados destacam a capacidade do método de decomposição em obter soluções competitivas em cenários de larga escala.

5.6.1.3 Primeiras soluções válidas sob a Variante A

Como demonstrado pela auditoria (Seção 5.2), 100% das soluções originalmente publicadas por [Stodola e Kutěj \(2024\)](#) para as instâncias p08-D a p23-D violam o modelo estrito da Variante A. A Tabela 28 estabelece as primeiras soluções válidas sob a Variante A para esse grupo e quantifica o custo da viabilidade, comparando a solução válida da Variante A com o resultado obtido por este trabalho sob a Variante B.

Ao analisar a Tabela 28, observa-se que impor o modelo estrito de autonomia (Variante A) acarreta um custo da viabilidade entre $+0,82\%$ e $+9,96\%$ em relação à Variante B no próprio algoritmo. Esse acréscimo representa o consumo adicional necessário para cobrir a espera e o tempo de serviço do caminhão durante as operações do drone.

Tabela 27: Comparativo direto sob a Variante B entre a abordagem proposta e os melhores resultados publicados por [Stodola e Kutěj \(2024\)](#).

Instância	Melhor (Makespan)		Variação (%)	Abordagem
	Literatura (Var. B)	Este trabalho (Var. B)		
p01-D	7.311,30	7.384,87	+1,01	Decomp. Refin.+OL
p02-D	6.560,22	6.635,49	+1,15	Decomp. Refin.+OL
p03-D	8.201,77	8.338,53	+1,67	Decomp. Refin.+OL
p04-D	26.020,01	26.013,60	-0,02	Decomp. Base+OL
p05-D	21.544,14	21.841,05	+1,38	Decomp. Refin.+OL
p06-D	15.987,21	16.032,30	+0,28	Decomp. Refin.+OL
p07-D	12.056,70	12.199,71	+1,19	Decomp. Refin.+OL
p08-D	114.693,98	117.283,39	+2,26	Decomp. Refin.+OL
p09-D	67.137,57	68.337,30	+1,79	Decomp. Refin.+OL
p10-D	48.325,85	49.566,46	+2,57	Decomp. Refin.+OL
p11-D	38.385,86	38.772,81	+1,01	Decomp. Refin.+OL
p12-D	26.297,99	26.950,81	+2,48	Decomp. Refin.+OL
p13-D	28.920,24	29.814,99	+3,10	Decomp. Base+OL
p14-D	31.717,19	32.221,14	+1,59	Decomp. Base+OL
p15-D	26.335,04	26.786,19	+1,71	Decomp. Refin.+OL
p16-D	30.536,91	30.372,10	-0,54	Decomp. Base+OL
p17-D	33.000,61	32.549,83	-1,37	Decomp. Refin.+OL
p18-D	26.851,32	27.222,19	+1,38	Decomp. Refin.+OL
p19-D	32.839,42	30.228,28	-7,95	Decomp. Base+OL
p20-D	35.744,56	32.765,77	-8,33	Decomp. Base+OL
p21-D	27.513,61	27.846,40	+1,21	Decomp. Base+OL
p22-D	35.060,45	30.657,47	-12,56	Decomp. Base+OL
p23-D	38.500,84	33.214,27	-13,73	Decomp. Base+OL

5.6.2 Relação entre *makespan* e distância total no MDVRP-D

A adoção do *makespan* (tempo de conclusão da rota mais longa) como função objetivo principal na presença de frotas híbridas com drones introduz uma divergência fundamental em relação à minimização tradicional da distância total comumente utilizada no VRP clássico. No VRP clássico, minimizar a distância total é correlato a minimizar o tempo e o custo de combustível. No MDVRP-D, contudo, a minimização do tempo da rota crítica (a mais longa de todas) pode levar a soluções em que as rotas não críticas realizam percursos consideravelmente mais longos com o único propósito de balancear o tempo final, o que eleva o consumo de recursos global. Inversamente, a minimização da distância total percorrida pelo somatório de caminhões e drones pode gerar um *makespan* insatisfatório, concentrando a maior parte dos serviços em um único depósito geograficamente centralizado e subutilizando os demais recursos logísticos.

Além disso, a função objetivo do *makespan* é inerentemente propensa a empates estruturais: diferentes configurações de rotas não críticas podem resultar exatamente no

Tabela 28: Primeiras soluções válidas sob a Variante A para as instâncias p08-D a p23-D e custo da viabilidade em relação à Variante B do próprio trabalho.

Instância	Este trabalho (Makespan)		Custo Viabilidade (%)
	Variante A	Variante B	
p08-D	119.238,45	117.283,39	+1,67
p09-D	68.897,52	68.337,30	+0,82
p10-D	50.491,41	49.566,46	+1,87
p11-D	39.358,90	38.772,81	+1,51
p12-D	28.021,75	26.950,81	+3,97
p13-D	30.800,77	29.814,99	+3,31
p14-D	34.954,39	32.221,14	+8,48
p15-D	27.585,02	26.786,19	+2,98
p16-D	31.305,10	30.372,10	+3,07
p17-D	34.607,79	32.549,83	+6,32
p18-D	27.847,84	27.222,19	+2,30
p19-D	31.558,46	30.228,28	+4,40
p20-D	36.028,77	32.765,77	+9,96
p21-D	28.379,09	27.846,40	+1,91
p22-D	31.351,10	30.657,47	+2,26
p23-D	35.850,12	33.214,27	+7,94

mesmo tempo máximo para a rota crítica. Sem um critério de desempate adequado, torna-se difícil para o algoritmo otimizar as rotas que não definem o gargalo, o que pode manter trechos ineficientes na solução final. Para mitigar esse problema, uma prática robusta em problemas multi-depósito consiste em adotar o somatório dos custos de todas as rotas ativas como critério secundário de decisão (ou seja, como desempate hierárquico). Embora a otimização primária busque minimizar a rota crítica, o desempate privilegia soluções com menor distância total. Na especificação original do AACO-NC-D de [Stodola e Kutěj \(2024\)](#), essa questão não é abordada formalmente, de forma que o algoritmo foca estritamente no *makespan* do gargalo crítico, limitação que orienta a necessidade de formalizações bi-objetivo e critérios de desempate mais finos em trabalhos futuros.

5.6.3 Síntese das contribuições

A análise conjunta dos focos experimentais revela uma progressão coerente de contribuições, cada uma construindo sobre as descobertas anteriores.

A revalidação e auditoria computacional (Seção 5.2) constitui a premissa da pesquisa: ao demonstrar que 81,1% das soluções publicadas por [Stodola e Kutěj \(2024\)](#) são inviáveis sob o modelo estrito da Variante A por omitirem o tempo de serviço do caminhão, a

auditoria evidenciou que a literatura operou sob o modelo relaxado da Variante B. Essa constatação redefiniu o ponto de partida do problema e motivou o estabelecimento dos primeiros *baselines* comprovadamente válidos para a Variante A estrita.

A Partição de Voronoi (Seção 5.3) consolidou-se como a estratégia de clusterização mais robusta, superando a Setorização Angular em 22 das 23 instâncias no custo médio. Seu mecanismo de diversidade adaptativa (α) permite calibrar a cobertura de fronteiras independentemente da escala do problema, revelando que instâncias com distribuição irregular beneficiam-se fortemente da sobreposição ($\alpha = 1,0$), enquanto instâncias com separação geográfica natural tornam $\alpha = 0$ competitivo.

A decomposição (Seção 5.3.2) contraria a expectativa intuitiva de que fragmentar o espaço de busca degradaria a qualidade: a Decomposição Pura supera o AACO-NC-D sobre o espaço completo com $\alpha = 0$ em 21 das 23 instâncias no makespan médio, com melhorias de até -26% nas instâncias de larga escala. O mecanismo explicativo é a redução exponencial nas dimensões da matriz de feromônio, que permite ao ACO de cada sub-instância explorar seu sub-espaço com maior profundidade. A limitação reside nas fronteiras inter-depósito, que a decomposição pura não otimiza.

A integração das estratégias (Seção 5.3.2) fecha o ciclo com dois resultados distintos. A *Decomposição como Semente* constituiu uma estratégia descartada: os ganhos marginais obtidos a custo temporal proibitivo indicam que a inicialização por feromônio seminal se dissolve durante a fase com múltiplos depósitos, tornando a abordagem ineficaz para o MDVRP-D nas configurações avaliadas, resultado que orienta negativamente pesquisas futuras. Já a *Decomposição como Refinamento* domina o AACO-NC-D sobre o espaço completo em todas as 23 instâncias, com melhorias de até -23% em larga escala. A comparação direta (Tabela 16) estabelece o critério prático de seleção: instâncias com clusters geograficamente equilibrados beneficiam-se da Decomposição Pura, enquanto instâncias com distribuição irregular requerem o Refinamento para cobrir as fronteiras que a decomposição pura deixa inexploradas.

6 Conclusão

Esta dissertação apresentou um estudo de replicação e o estudo de ablação do algoritmo AACO-NC-D, proposto por [Stodola e Kutěj \(2024\)](#) para o Problema de Roteirização de Veículos Multi-Depósito com Drones (MDVRP-D), modelo que se distingue de seus predecessores pela combinação de múltiplos depósitos, frotas heterogêneas caminhão-drone e sincronização temporal. A revisão da literatura identificou que abordagens ACO anteriores para problemas com drones utilizavam probabilidades constantes de despacho e matrizes de feromônio únicas, limitações superadas pelo AACO-NC-D, e confirmou a ausência de estudos de replicação ou análise modular do algoritmo. A implementação em C++20 contemplou a estrutura algorítmica completa do modelo-base, a formalização da Setorização Angular e a proposta da Partição de Voronoi, os operadores de busca local e as três estratégias de decomposição do problema (Pura, Semente e Refinamento). A auditoria computacional identificou que a restrição de autonomia do drone admite duas variantes operacionais (Variante A estrita e Variante B relaxada), demonstrando que as soluções publicadas na literatura de referência são aderentes à Variante B e estabelecendo os primeiros *baselines* de comparação válidos sob a Variante A para o conjunto completo de instâncias MDVRP-D.

6.1 Contribuições principais

As contribuições desta dissertação articulam-se em cinco eixos centrais. A primeira contribuição consiste na condução de uma auditoria independente sobre a literatura de referência, identificando a adesão das soluções publicadas à Variante B e estabelecendo os primeiros patamares de comparação válidos sob a Variante A para instâncias de médio e grande porte (p08-D a p23-D). A segunda contribuição traz o esclarecimento terminológico do operador *k-opt* modificado, evidenciando a discrepância entre a nomenclatura clássica e a implementação baseada em realocação de blocos, fundamentando uma crítica sobre o rigor descritivo em meta-heurísticas. A terceira contribuição é a proposta e

avaliação da Partição de Voronoi com listas de candidatos de sobreposição de fronteira adaptativa, em que o parâmetro α regula a inclusão de vizinhos de regiões adjacentes para clientes de borda, superando a Setorização Angular em 22 das 23 instâncias no *makespan* médio com reduções de até 33% no tempo de conclusão. A quarta contribuição refere-se à avaliação da decomposição pura em sub-instâncias VRP-D independentes sob o paradigma *Cluster-First, Route-Second*, demonstrando que o particionamento reduz a dimensão das matrizes de feromônio em um fator m^2 e supera a otimização no espaço completo em 21 das 23 instâncias no *makespan* médio. Por fim, a quinta contribuição avalia os modos de integração entre decomposição e otimização global, comprovando a superioridade da Decomposição como Refinamento, que dominou a abordagem global no melhor *makespan* em todas as instâncias, e fundamentando a rejeição da Decomposição como Semente devido à rápida dissolução da estrutura inicial no feromônio global.

6.2 Limitações

Este trabalho apresenta limitações metodológicas e computacionais que devem ser consideradas na interpretação dos achados. A primeira delas refere-se às diferenças de ambiente computacional: o processador Intel Core i5-13420H utilizado possui desempenho multi-thread aproximadamente 38% inferior ao Intel Core i9-10940X empregado no estudo original, o que limita a comparação direta de tempos absolutos de execução sem normalização. A segunda limitação está ligada ao critério de parada conservador de 2.000 iterações sem melhoria, adotado para viabilizar o orçamento computacional dos experimentos de ablação mas que pode não ter esgotado a convergência em instâncias de grande porte. Por fim, a terceira limitação envolve a eficácia da busca local sobre as viagens de drone, uma vez que a abordagem de tentativa e validação executa movimentos terrestres antes de checar a viabilidade aérea; reposicionar vértices de suporte frequentemente invalida as viagens ativas do drone, reduzindo a intensidade de refinamento na frota aérea. Soma-se a isso a ausência de uma metaheurística alternativa implementada para comparação direta, decorrente da falta de detalhamento do artigo original sobre a setorização e o tratamento das viagens aterrissadas de drone na otimização local. Busca local capaz de reconstruir viagens do drone dinamicamente e comparação com outras meta-heurísticas (Seção 6.3) são os caminhos naturais para suprir essa lacuna.

6.3 Trabalhos futuros

Com base nos resultados obtidos e nas limitações identificadas, diversas direções de pesquisa são propostas para trabalhos futuros.

No âmbito algorítmico, destaca-se o desenvolvimento de um *pipeline* de decomposição adaptativa. Fundamentado nas constatações do Capítulo 5, o sistema executaria a Decomposição Pura como uma sonda computacional de baixo custo para avaliar o coeficiente de variação de carga entre os depósitos; caso o desequilíbrio geográfico supere o limiar crítico de 4%, a estratégia de Refinamento seria acionada automaticamente, poupando tempo computacional em instâncias regularmente distribuídas.

Outra vertente relevante refere-se à formalização do MDVRP-D sob uma abordagem bi-objetivo, buscando otimizar simultaneamente o tempo máximo de conclusão (*makespan*) e o custo da distância total percorrida pelas frotas terrestre e aérea. Essa formulação resolve o problema de desempate estrutural identificado nos experimentos, no qual rotas não críticas acumulam distâncias desnecessárias sem alterar o *makespan* global.

Por fim, avanços na modelagem física e operacional do problema constituem uma direção promissora. Sugere-se a expansão do modelo para contemplar múltiplos drones por caminhão, permitindo explorar o paralelismo aéreo em maior profundidade, bem como a transição para redes com distâncias e tempos reais (não-euclidianas), integrando dados de malhas rodoviárias e relevo extraídos de sistemas de informação geográfica para aproximar o modelo das condições práticas de tráfego urbano.

REFERÊNCIAS

AGATZ, Niels; BOUMAN, Paul; SCHMIDT, Marie. Optimization Approaches for the Traveling Salesman Problem with Drone. **Transportation Science**, v. 52, n. 4, p. 965–981, 2018.

BALLOU, Ronald H. **Business Logistics/Supply Chain Management**. 5. ed. Upper Saddle River, NJ: Pearson Education, 2004.

CLARKE, G.; WRIGHT, J. W. Scheduling of Vehicles from a Central Depot to a Number of Delivery Points. **Operations Research**, v. 12, n. 4, p. 568–581, 1964. DOI: [10.1287/opre.12.4.568](https://doi.org/10.1287/opre.12.4.568).

COSTA, Daniel Ferreira. Uma Abordagem Meta-Heurística Evolutiva para a Otimização de Rotas de Entrega Fracionada de Mercadorias Utilizando Caminhões e Drones. In: ANAIS do LV Simpósio Brasileiro de Pesquisa Operacional. São José dos Campos: Galoá, 2023. Disponível em:

<<https://proceedings.science/sbpo/sbpo-2023/trabalhos/uma-abordagem-meta-heuristica-evolutiva-para-a-otimizacao-de-rotas-de-entrega-fr>>.

Acesso em: 25 jul. 2026.

DANTZIG, George; FULKERSON, Ray; JOHNSON, Selmer. Solution of a large-scale traveling-salesman problem. **Operations Research**, INFORMS, v. 2, n. 4, p. 393–410, 1954.

DANTZIG, George B; RAMSER, John H. The truck dispatching problem. **Management science**, INFORMS, v. 6, n. 1, p. 80–91, 1959.

DAS, D. N. et al. Synchronized truck and drone routing in package delivery logistics. **IEEE Transactions on Intelligent Transportation Systems**, v. 22, n. 9, p. 5772–5782, 2021. DOI: [10.1109/TITS.2020.2992549](https://doi.org/10.1109/TITS.2020.2992549).

DELL'AMICO, Mauro; MONTEMANNI, Roberto; NOVELLANI, Stefano. Matheuristic algorithms for the flying sidekick traveling salesman problem. **Optimization Letters**, v. 15, n. 7, p. 2449–2467, 2021. DOI: [10.1007/s11590-020-01615-3](https://doi.org/10.1007/s11590-020-01615-3).

- DORIGO, Marco; MANIEZZO, Vittorio; COLORNI, Alberto. Ant system: optimization by a colony of cooperating agents. **IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)**, v. 26, n. 1, p. 29–41, 1996. DOI: [10.1109/3477.484436](https://doi.org/10.1109/3477.484436).
- EUCHI, Jalel; SADOK, Asma. Hybrid genetic-sweep algorithm to solve the vehicle routing problem with drones. **Physical Communication**, v. 44, p. 101236, 2021. DOI: [10.1016/j.phycom.2020.101236](https://doi.org/10.1016/j.phycom.2020.101236).
- FREITAS, J. C. D.; PENNA, P. H. V. A variable neighborhood search for flying sidekick traveling salesman problem. **International Transactions in Operational Research**, v. 27, n. 1, p. 267–290, 2019. DOI: [10.1111/itor.12671](https://doi.org/10.1111/itor.12671).
- GILLET, Billy E; JOHNSON, Jerry G. Multi-terminal vehicle-dispatch algorithm. **Omega**, v. 4, n. 6, p. 711–718, 1976. ISSN 0305-0483. DOI: [https://doi.org/10.1016/0305-0483\(76\)90097-9](https://doi.org/10.1016/0305-0483(76)90097-9). Disponível em: <https://www.sciencedirect.com/science/article/pii/0305048376900979>.
- GILLET, Billy E; MILLER, Leland R. A heuristic algorithm for the vehicle-dispatch problem. **Operations Research**, INFORMS, v. 22, n. 2, p. 340–349, 1974.
- HA, Quang Minh et al. On the min-cost traveling salesman problem with drone. **Transportation Research Part C: Emerging Technologies**, v. 86, p. 597–621, 2018. DOI: [10.1016/j.trc.2017.11.015](https://doi.org/10.1016/j.trc.2017.11.015).
- HALIM, A. H.; ISMAIL, I. Combinatorial optimization: Comparison of heuristic algorithms in travelling salesman problem. **Archives of Computational Methods in Engineering**, v. 26, p. 367–380, 2019. DOI: [10.1007/s11831-017-9247-y](https://doi.org/10.1007/s11831-017-9247-y).
- HUANG, S.-H. et al. Solving the vehicle routing problem with drone for delivery services using an ant colony optimization algorithm. **Advanced Engineering Informatics**, v. 51, p. 101536, 2022. DOI: [10.1016/j.aei.2022.101536](https://doi.org/10.1016/j.aei.2022.101536).
- JUNIOR, Elias Lawrence Marques et al. A Multi-objective Metaheuristic for a Green UAV Grid Routing Problem. In: LECTURE Notes in Computer Science. [S. l.]: Springer, 2020. v. 12010, p. 152–166. ISBN 978-3-030-44931-5.
- JÚNIOR, Elias Marques; COELHO, Vitor Nazário et al. UAVs routes optimization on smart cities and regions. **RAIRO – Operations Research**, v. 56, n. 2, p. 853–869, 2022.

- JÚNIOR, Elias Marques; OCHI, Luiz Satoru et al. A Two-Phase Multi-objective Metaheuristic for a Green UAV Grid Routing Problem. **Optimization Letters**, v. 17, p. 2233–2256, 2023.
- LIANG, Y.-J.; LUO, Z.-X. A survey of truck-drone routing problem: literature review and research prospects. **Journal of the Operations Research Society of China**, v. 10, p. 343–377, 2022. DOI: [10.1007/s40305-021-00383-4](https://doi.org/10.1007/s40305-021-00383-4).
- LIM, Stanley Frederick W.T.; JIN, Xin; SRAI, Jagjit Singh. Consumer-driven e-commerce: A literature review, design framework, and research agenda on last-mile logistics models. **International Journal of Physical Distribution & Logistics Management**, v. 48, n. 3, p. 308–332, mar. 2018. ISSN 0960-0035. DOI: [10.1108/IJPDLM-02-2017-0081](https://doi.org/10.1108/IJPDLM-02-2017-0081). eprint: <https://www.emerald.com/ijpdlm/article-pdf/48/3/308/2430054/ijpdlm-02-2017-0081.pdf>. Disponível em: <https://doi.org/10.1108/IJPDLM-02-2017-0081>.
- MANGIARACINA, Riccardo et al. Innovative solutions to increase last-mile delivery efficiency in B2C e-commerce: a literature review. **International Journal of Physical Distribution & Logistics Management**, v. 49, n. 9, p. 901–920, out. 2019. ISSN 0960-0035. DOI: [10.1108/IJPDLM-02-2019-0048](https://doi.org/10.1108/IJPDLM-02-2019-0048). eprint: <https://www.emerald.com/ijpdlm/article-pdf/49/9/901/1109218/ijpdlm-02-2019-0048.pdf>. Disponível em: <https://doi.org/10.1108/IJPDLM-02-2019-0048>.
- MURRAY, Chase C; CHU, Amanda 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.
- PASSMARK SOFTWARE. **CPU Benchmarks: Intel Core i5-13420H vs Intel Core i9-10940X**. [S. l.: s. n.], 2026. Acessado em: 18 mar. 2026. Disponível em: <https://www.cpubenchmark.net/compare/5434vs3639/Intel-i5-13420H-vs-Intel-i9-10940X>.
- PENNA, Puca Huachi Vaz. **Um algoritmo unificado para uma classe de problemas de roteamento de veículos com frota heterogênea**. 2013. Tese de Doutorado – Universidade Federal Fluminense, Niterói, RJ. Orientador: Luiz Satoru Ochi.
- POIKONEN, S.; WANG, Xin; GOLDEN, Bruce. The vehicle routing problem with drones: Extended models and connections. **Networks**, v. 70, n. 1, p. 34–43, 2017. DOI: [10.1002/net.21746](https://doi.org/10.1002/net.21746).

PONZA, Andrea. **Optimization of drone-assisted parcel delivery**. 2016. Diss. (Mestrado) – University of Padova.

SACRAMENTO, Daniel; PISINGER, David; ROPKE, Stefan. An adaptive large neighborhood search metaheuristic for the vehicle routing problem with drones.

Transportation Research Part C: Emerging Technologies, v. 102, p. 289–315, 2019. DOI: [10.1016/j.trc.2019.02.018](https://doi.org/10.1016/j.trc.2019.02.018).

SCHERMER, Daniel; MOEINI, Mahdi; WENDT, Oliver. A hybrid VNS/Tabu search algorithm for solving the vehicle routing problem with drones and en route operations.

Computers & Operations Research, v. 109, p. 134–158, 2019. DOI: [10.1016/j.cor.2019.04.021](https://doi.org/10.1016/j.cor.2019.04.021).

STODOLA, Petr. Hybrid ant colony optimization algorithm applied to the multi-depot vehicle routing problem. **Natural Computing**, v. 19, n. 2, p. 463–475, 2020. DOI:

[10.1007/s11047-018-9711-y](https://doi.org/10.1007/s11047-018-9711-y).

STODOLA, Petr; KUTĚJ, Libor. Multi-Depot Vehicle Routing Problem with Drones:

Mathematical formulation, solution algorithm and experiments. **Expert Systems with Applications**, v. 241, p. 122483, 2024. ISSN 0957-4174. DOI:

<https://doi.org/10.1016/j.eswa.2023.122483>. Disponível em:

<<https://www.sciencedirect.com/science/article/pii/S0957417423029858>>.

STODOLA, Petr; NOHEL, Jan. Adaptive Ant Colony Optimization With Node Clustering for the Multidepot Vehicle Routing Problem. **IEEE Transactions on**

Evolutionary Computation, v. 27, n. 6, p. 1652–1666, 2023. DOI:

[10.1109/TEVC.2022.3228827](https://doi.org/10.1109/TEVC.2022.3228827).

SUBRAMANIAN, Anand. **Heuristic, Exact and Hybrid Approaches for Vehicle Routing Problems**. 2012. Tese de Doutorado – Universidade Federal Fluminense,

Niterói, RJ. Orientador: Luiz Satoru Ochi.

TAMKE, Felix; BUSCHER, Udo. A branch-and-cut algorithm for the vehicle routing problem with drones. **Transportation Research Part B: Methodological**, v. 144,

p. 174–203, 2021. DOI: [10.1016/j.trb.2020.11.011](https://doi.org/10.1016/j.trb.2020.11.011).

TOTH, Paolo; VIGO, Daniele. **Vehicle routing: problems, methods, and applications**. 2. ed. Philadelphia: SIAM, 2014.

WANG, Xin; POIKONEN, S.; GOLDEN, Bruce. The vehicle routing problem with drones: Several worst-case results. **Optimization Letters**, v. 11, p. 679–697, 2017.

DOI: [10.1007/s11590-016-1035-3](https://doi.org/10.1007/s11590-016-1035-3).

WENG, Xiang et al. Multi-depot vehicle routing problem with drones in emergency logistics. **Cluster Computing**, v. 28, n. 1, p. 1–27, 2024. DOI: [10.1007/s10586-024-04809-5](https://doi.org/10.1007/s10586-024-04809-5).

WINDRAS MARA, Siti Tatik; RIFAI, Ahmad Perwira; SOPHA, Bermawi Priyatna. An adaptive large neighborhood search heuristic for the flying sidekick traveling salesman problem with multiple drops. **Expert Systems with Applications**, v. 205, p. 117647, 2022. DOI: [10.1016/j.eswa.2022.117647](https://doi.org/10.1016/j.eswa.2022.117647).

YÜKSEL, Zeynep; ELIGÜZEL, Nilüfer; METE, Süleyman. Multi-Depot Vehicle Routing Problem with Drone Collaboration in Humanitarian Logistics. **Journal of Optimization & Decision Making**, v. 3, n. 1, p. 438–448, 2024.