



UNIVERSIDADE FEDERAL DO MARANHÃO
CENTRO DE CIÊNCIAS EXATAS E TECNOLOGIA - CCET
ENGENHARIA DA COMPUTAÇÃO

Renan Sousa Freire

**Otimização de Cobertura de Áreas por Drones:
Uma Análise Comparativa de Algoritmos de
Enxame**

São Luís - MA

2026

Renan Sousa Freire

Otimização de Cobertura de Áreas por Drones: Uma Análise Comparativa de Algoritmos de Enxame

Trabalho de Conclusão de Curso (TCC)
apresentado como requisito para a obtenção
do grau de Bacharel em Engenharia da
Computação pela Universidade Federal do
Maranhão – UFMA

Orientador: Prof. Dr. Pedro Baptista Fernandes

São Luís - MA

2026

RENAN SOUSA FREIRE

Otimização de Cobertura de Áreas por Drones: Uma Análise Comparativa de Algoritmos de Enxame

Trabalho de Conclusão de Curso (TCC)
apresentado como requisito para a obtenção
do grau de Bacharel em Engenharia da
Computação pela Universidade Federal do
Maranhão – UFMA

Orientador: Prof. Dr. Pedro Baptista
Fernandes

Aprovado em 17 de julho de 2026.

BANCA EXAMINADORA

Prof. Dr. Pedro Baptista Fernandes - UFMA
(Orientador)

Prof. Dr. Dênis Fabrício Sousa de Sá - UFMA

Prof. Me. Márcio Mendes Cerqueira - UFMA

São Luís - MA

2026

Ficha gerada por meio do SIGAA/Biblioteca com dados fornecidos pelo(a) autor(a).
Diretoria Integrada de Bibliotecas/UFMA

Freire, Renan Sousa.

Otimização de cobertura de áreas por drones : uma
análise comparativa de algoritmos de enxame / Renan Sousa
Freire. - 2026.
67 f.

Orientador(a): Pedro Baptista Fernandes.
Curso de Engenharia da Computação, Universidade Federal
do Maranhão, São Luís - Ma, 2026.

1. Algoritmos de Enxame. 2. Cobertura de Área. 3.
Vants. 4. Otimização Por Meta-heurísticas. 5.
Posicionamento de Sensores. I. Fernandes, Pedro Baptista.
II. Título.

Agradecimentos

A Deus, pela sabedoria e força que sustentaram cada etapa deste trabalho.

Ao meu orientador, Prof. Dr. Pedro Baptista Fernandes, pela paciência, pela atenção dedicada e pelas contribuições que tornaram este trabalho possível.

Aos meus pais, por todo apoio ao longo da minha vida, sobretudo durante minha trajetória acadêmica. Agradeço por todos os ensinamentos que me ajudaram a me tornar quem eu sou hoje e a conseguir chegar até aqui.

À Giovanna, minha namorada, por sempre se fazer presente em todos os momentos, por me incentivar quando eu pensava que não seria capaz, por me ajudar com os estudos mesmo sem entender nada do que eu falava, e por tantas outras formas de apoio ao longo desta trajetória.

À Maria Carolina, minha irmã, por sempre se preocupar comigo e me apoiar até quando eu não queria.

Aos meus outros familiares e amigos que sempre me apoiaram, oraram por mim e me incentivaram a dar o melhor de mim.

À Universidade Federal do Maranhão e a todos os professores que contribuíram para minha formação e aperfeiçoamento pessoal.

A todos que, de alguma forma, colaboraram para a realização deste trabalho, o meu sincero agradecimento.

"Você é o protagonista dessa história — só não esquece de salvar o jogo de vez em quando."

— Autor desconhecido, provavelmente um NPC sábio do tutorial

Resumo

O uso de Veículos Aéreos Não Tripulados (VANTs) para a cobertura de áreas extensas é uma aplicação crescente em diversos setores da engenharia, mas a coordenação eficiente de múltiplos drones para esta tarefa constitui um problema de otimização complexo, e há uma lacuna na literatura quanto a estudos que comparem sistematicamente as abordagens de inteligência de enxame sob este problema específico. Este trabalho realizou um estudo comparativo do desempenho de cinco meta-heurísticas de enxame — *Particle Swarm Optimization* (PSO), *Grey Wolf Optimizer* (GWO), *Whale Optimization Algorithm* (WOA), *Salp Swarm Algorithm* (SSA) e *Cuckoo Search* (CS) — aplicadas ao posicionamento estático de VANTs para cobertura de área, avaliadas por uma função custo multi-objetivo normalizada que agrega cobertura, sobreposição e conectividade de rede. Os algoritmos foram testados em nove cenários (três escalas de área e três densidades de drones), sob três perfis de peso representando prioridades operacionais distintas, com vinte execuções independentes por condição (3.240 execuções no total), e as diferenças de desempenho foram validadas pelo teste de Wilcoxon com correção de Holm-Bonferroni. Os resultados indicam que o PSO foi o algoritmo mais eficiente e robusto, obtendo o menor custo médio em 25 das 27 combinações de cenário e perfil (ranking médio de 1,07) e superioridade estatisticamente significativa sobre WOA, SSA, CS e o *baseline* aleatório em todas as combinações; o GWO consolidou-se como segunda opção, estatisticamente equivalente ao PSO no regime de drones redundantes. Constatou-se ainda que a densidade de drones, e não a escala física da área, é a variável que efetivamente define o comportamento dos algoritmos, e que a sobreposição entre coberturas muda de natureza conforme o regime: evitável quando a capacidade do enxame é inferior à área, geometricamente inevitável quando a excede.

Palavras-chave: Algoritmos de Enxame; Cobertura de Área; VANTs; Otimização por Meta-heurísticas; Posicionamento de Sensores.

Abstract

The use of Unmanned Aerial Vehicles (UAVs) for large-scale area coverage is a growing application in several engineering sectors, yet the efficient coordination of multiple drones for this task constitutes a complex optimization problem, and there is a gap in the literature regarding studies that systematically compare swarm intelligence approaches on this specific problem. This work carried out a comparative study of five swarm metaheuristics —Particle Swarm Optimization (PSO), Grey Wolf Optimizer (GWO), Whale Optimization Algorithm (WOA), Salp Swarm Algorithm (SSA), and Cuckoo Search (CS) — applied to the static positioning of UAVs for area coverage, evaluated through a normalized multi-objective cost function aggregating coverage, overlap, and network connectivity. The algorithms were tested on nine scenarios (three area scales and three drone densities), under three weight profiles representing distinct operational priorities, with twenty independent runs per condition (3,240 runs in total), and performance differences were validated by the Wilcoxon signed-rank test with Holm-Bonferroni correction. Results indicate that PSO was the most efficient and robust algorithm, achieving the lowest mean cost in 25 of the 27 scenario-profile combinations (mean ranking of 1.07) and statistically significant superiority over WOA, SSA, CS, and the random baseline in all combinations; GWO ranked second, being statistically equivalent to PSO in the redundant-drone regime. It was further found that drone density, rather than the physical scale of the area, is the variable that effectively governs algorithm behavior, and that coverage overlap changes in nature across regimes: avoidable when the swarm’s capacity is below the area, geometrically unavoidable when it exceeds it.

Keywords: Swarm Algorithms; Area Coverage; UAVs; Metaheuristic Optimization; Sensor Placement.

Lista de ilustrações

Figura 1 – Fluxograma do procedimento experimental.	41
Figura 2 – Curvas de convergência do custo médio — cenário 100×100 suficiente, Perfil A.	48
Figura 3 – Posicionamento final dos drones — cenário 100×100 escasso, Perfil A.	48
Figura 4 – Posicionamento final dos drones — cenário 100×100 redundante, Perfil A.	49
Figura 5 – Posicionamento final dos drones — cenário 100×100 redundante, Perfil C (rede prioritária).	49

Lista de Códigos

B.1	Função objetivo normalizada e vetorizada.	65
B.2	Identificação do maior componente conexo via <i>Union-Find</i>	66
B.3	Execução de uma <i>run</i> independente com semente pareada.	67

Lista de tabelas

Tabela 1 – Parâmetros ópticos por escala de área	37
Tabela 2 – Perfis de peso da função custo	39
Tabela 3 – Caracterização dos cenários experimentais: número de drones e capacidade nominal de cobertura	45
Tabela 4 – Custo médio (desvio padrão) por algoritmo e densidade — escala 100×100 m, Perfil A	46
Tabela 5 – Cobertura média (%) por algoritmo e densidade — escala 100×100 m, Perfil A	46
Tabela 6 – Ranking médio por perfil de peso, ranking médio global e vitórias isoladas (27 combinações)	50
Tabela 7 – Síntese dos testes de Wilcoxon com correção de Holm-Bonferroni: número de combinações com diferença significativa ($\alpha = 0,05$), por densidade (máximo de 9 por densidade; 27 no total)	51
Tabela 8 – Ganho médio de cobertura sobre o <i>baseline</i> aleatório (pontos percentuais), por densidade — Perfil A, média das três escalas	52
Tabela 9 – Número médio de pares de drones em sobreposição, por densidade — Perfil A, média das três escalas	53
Tabela 10 – Custo médio (desvio padrão) por algoritmo, cenário e densidade — escalas 100×100 e 250×250 m, Perfil A (cobertura prioritária)	62
Tabela 11 – Custo médio (desvio padrão) por algoritmo, cenário e densidade — escalas 100×100 e 250×250 m, Perfil B (transição)	62
Tabela 12 – Custo médio (desvio padrão) por algoritmo, cenário e densidade — escalas 100×100 e 250×250 m, Perfil C (rede prioritária)	63
Tabela 13 – P-valores do teste de Wilcoxon para os confrontos PSO $\times$ GWO, GWO $\times$ WOA e SSA $\times$ CS — Perfil A (cobertura prioritária) (* indica diferença significativa após correção de Holm-Bonferroni, $\alpha = 0,05$)	63
Tabela 14 – P-valores do teste de Wilcoxon para os confrontos PSO $\times$ GWO, GWO $\times$ WOA e SSA $\times$ CS — Perfil B (transição) (* indica diferença significativa após correção de Holm-Bonferroni, $\alpha = 0,05$)	64
Tabela 15 – P-valores do teste de Wilcoxon para os confrontos PSO $\times$ GWO, GWO $\times$ WOA e SSA $\times$ CS — Perfil C (rede prioritária) (* indica diferença significativa após correção de Holm-Bonferroni, $\alpha = 0,05$)	64

Lista de Quadros

1	Hiperparâmetros adotados na implementação dos algoritmos	32
2	Comparação dos Mecanismos de Exploração-Explotação	32

Lista de abreviaturas e siglas

ACO	Ant Colony Optimization
CS	Cuckoo Search
DSU	Disjoint Set Union (Union-Find)
FOV	Field of View (Campo de Visão)
GWO	Grey Wolf Optimizer
HFOV	Horizontal Field of View (Campo de Visão Horizontal)
LIDAR	Light Detection and Ranging
MDE	Modelo Digital de Elevação
MDT	Modelo Digital de Terreno
MSE	Mean Squared Error (Erro Quadrático Médio)
NDVI	Normalized Difference Vegetation Index
NFL	No Free Lunch (teorema)
NIR	Near Infrared (Infravermelho Próximo)
PSO	Particle Swarm Optimization
RGB	Red, Green, Blue
SI	Swarm Intelligence (Inteligência de Enxame)
SSA	Salp Swarm Algorithm
ToF	Time-of-Flight
VANT	Veículo Aéreo Não Tripulado
VTOL	Vertical Takeoff and Landing
WOA	Whale Optimization Algorithm

Sumário

1	INTRODUÇÃO	15
1.1	Objetivos	16
1.1.1	Geral	17
1.1.2	Específicos	17
2	FUNDAMENTAÇÃO TEÓRICA	18
2.1	Veículos Aéreos Não Tripulados (VANTs)	18
2.1.1	Arquiteturas e Dinâmica de Voo	18
2.1.2	Sensores para Cobertura de Área	19
2.2	O Problema de Cobertura de Área	20
2.2.1	Modelagem Matemática (Disco Binário)	20
2.2.2	Restrições Operacionais	21
2.2.3	Função Objetivo	22
2.3	Otimização por Meta-heurísticas	22
2.3.1	Inteligência de Enxame	23
2.4	Algoritmos de Enxame Selecionados	24
2.4.1	Particle Swarm Optimization (PSO)	24
2.4.2	Grey Wolf Optimizer (GWO)	25
2.4.3	Whale Optimization Algorithm (WOA)	27
2.4.4	Salp Swarm Algorithm (SSA)	28
2.4.5	Cuckoo Search (CS)	29
3	METODOLOGIA	33
3.1	Abordagem da Pesquisa	33
3.2	Modelo do Problema e Ambiente de Simulação	33
3.3	Função Custo e Normalização	34
3.3.1	Componente de Cobertura	35
3.3.2	Componente de Sobreposição	35
3.3.3	Componente de Conectividade	35
3.4	Delineamento Experimental	36
3.4.1	Parâmetros Fixos	36
3.4.2	Parâmetros de Cenário (Variáveis Independentes)	38
3.4.3	Perfis de Peso e Análise de Robustez	38
3.4.4	Calibração do Raio de Comunicação	40
3.4.5	Procedimento Experimental	40
3.5	Análise Estatística e Critérios de Comparação	42

4	RESULTADOS E DISCUSSÃO	44
4.1	Apresentação Geral dos Resultados	44
4.2	Análise por Regime de Densidade	45
4.2.1	Regime Escasso	46
4.2.2	Regime Suficiente	47
4.2.3	Regime Redundante	47
4.2.4	Figuras Seleccionadas	47
4.3	Robustez entre Perfis de Peso	49
4.4	Significância Estatística	50
4.5	Comparação com o <i>Baseline</i>	51
4.6	Discussão dos Achados Geométricos	52
4.6.1	Invariância de Escala do Modelo Normalizado	52
4.6.2	Sobreposição e Cobertura por Regime de Densidade	53
5	CONCLUSÃO	55
5.1	Síntese dos Resultados e Resposta aos Objetivos	55
5.2	Limitações	56
5.3	Trabalhos Futuros	57
	REFERÊNCIAS	58
	APÊNDICES	61
	APÊNDICE A – TABELAS COMPLETAS DE RESULTADOS	62
A.1	Custo Médio por Perfil de Peso	62
A.2	P-valores dos Confrontos Disputados	63
	APÊNDICE B – TRECHOS-CHAVE DO CÓDIGO DE SIMULAÇÃO	65
B.1	Função Objetivo Normalizada e Vetorizada	65
B.2	Maior Componente Conexo via Union-Find	66
B.3	Execução Independente com Semente Pareada	67

1 Introdução

Inicialmente projetados como ferramentas para combate militar e missões de reconhecimento em ambientes hostis, os Veículos Aéreos Não Tripulados (VANTs), popularmente conhecidos como drones, passaram por notáveis transformações em seu design nas últimas décadas para servir em diferentes áreas (NWAOGU et al., 2023). Hoje, os VANTs representam uma tecnologia disruptiva com um vasto leque de aplicações civis. Essa expansão foi impulsionada por constantes avanços em suas tecnologias embarcadas, como sensores multiespectrais e sistemas LIDAR (ALSAMHI et al., 2022), que viabilizaram seu emprego em áreas como a inspeção de grandes infraestruturas (NWAOGU et al., 2023) e o monitoramento de precisão na agricultura (ZHANG; KOVACS, 2012). Dentre essas diversas aplicações, uma das mais relevantes e desafiadoras consiste na utilização de múltiplos drones para a cobertura otimizada de área, uma tarefa fundamental para missões de mapeamento topográfico, monitoramento de safras agrícolas e vigilância ambiental.

O emprego de VANTs em setores estratégicos como a engenharia civil e a agricultura de precisão oferece vantagens significativas sobre métodos tradicionais, como a rápida aquisição de dados em larga escala e a notável redução de custos operacionais (FRIHA et al., 2021). O avanço de tecnologias como o sensoriamento remoto e a inteligência artificial tem ampliado ainda mais o potencial dessas aplicações (TSOUROS; BIBI; SARIGIANNIDIS, 2019; ELIJAH et al., 2018). Contudo, a efetividade das missões com drones depende de fatores críticos, que vão desde as características do VANT, como autonomia e estabilidade, até as especificações dos sensores embarcados, por exemplo, resolução e campo de visão (SANKARAN et al., 2015). Essa complexidade é amplificada ao se utilizar múltiplos drones em conjunto para cobrir grandes áreas, criando um desafio de coordenação fundamental: como orquestrar o esquadrão para maximizar a área coberta, minimizando a sobreposição e mantendo a conectividade da rede? A resolução deste problema de otimização é crucial para aumentar a eficiência e a confiabilidade da tecnologia, tornando-a mais acessível e robusta.

Para solucionar problemas de otimização complexos como este, a ciência da computação oferece uma classe de técnicas poderosas conhecidas como meta-heurísticas (BLUM; ROLI, 2003). Dentre elas, destacam-se os algoritmos de enxame, uma vertente da Inteligência Artificial que se inspira no comportamento coletivo e auto-organizado de sistemas naturais, como revoadas de pássaros ou colônias de formigas (BONABEAU; DORIGO; THERAULAZ, 1999). Essa inspiração deu origem a alguns dos algoritmos de otimização mais conhecidos, como o Particle Swarm Optimization (PSO), proposto em (KENNEDY; EBERHART, 1995), que demonstram grande eficácia em encontrar soluções de alta qualidade em um tempo computacional viável. Subsequentemente, a área

expandiu-se com novos modelos inspirados em diferentes comportamentos da natureza, como é o caso do Grey Wolf Optimizer (GWO), proposto em (MIRJALILI; MIRJALILI; LEWIS, 2014), e o Whale Optimization Algorithm (WOA), apresentado em (MIRJALILI; LEWIS, 2016). Desse modo, a aplicação desses algoritmos se mostra particularmente promissora neste trabalho, uma vez que a dinâmica de um enxame de drones autônomos possui uma analogia direta com os sistemas biológicos que inspiraram essas técnicas, tornando-os candidatos ideais para orquestrar o posicionamento e a cobertura de área de forma eficiente e descentralizada.

Para a presente análise comparativa, foi selecionado um conjunto diversificado de cinco meta-heurísticas. A seleção inclui algoritmos já consagrados e amplamente utilizados como *benchmarks* na literatura, como o PSO, o GWO e o WOA, cujo desempenho em diversos problemas de otimização é frequentemente documentado e comparado (DOKEROGLU; DENIZ; KIZILOZ, 2019). Para ampliar o escopo da comparação e testar diferentes paradigmas de busca, foram incluídos também algoritmos com mecanismos distintos: o Salp Swarm Algorithm (SSA), que simula o comportamento de cadeias de salpas marinhas (MIRJALILI et al., 2017) e tem apresentado promissor avanço na resolução de problemas de engenharia com restrições (TUBA; ALIHODZIC; TUBA, 2018), e o Cuckoo Search (CS), inspirado no parasitismo de ninhada e caracterizado pelo uso de Voos de Lévy para buscas globais eficientes (YANG; DEB, 2009). Essa seleção heterogênea é proposital e visa uma avaliação robusta, pois, conforme os teoremas "*No Free Lunch*", não existe um único algoritmo universalmente superior para todos os problemas de otimização, reforçando a necessidade de estudos comparativos focados em domínios específicos (WOLPERT; MACREADY, 1997).

Apesar da promessa teórica e da vasta gama de algoritmos de enxame disponíveis na literatura — como o PSO, GWO, WOA, SSA e CS —, a escolha da abordagem mais eficaz não é direta, uma vez que seu desempenho pode variar drasticamente conforme a natureza e as restrições do problema de otimização a ser aplicado. Atualmente, existe uma lacuna na literatura no que tange a estudos comparativos sistemáticos que avaliem o desempenho desses cinco algoritmos sob o problema específico de cobertura de área por múltiplos VANTs com posicionamento estático. Portanto, este trabalho se justifica pela necessidade de desenvolver e estruturar uma análise comparativa robusta, a fim de prover uma avaliação aprofundada dos pontos fortes e fracos de cada algoritmo e identificar as estratégias mais promissoras para esta aplicação.

1.1 Objetivos

Diante do que foi exposto, este trabalho tem como alicerce os seguintes objetivos geral e específicos:

1.1.1 Geral

Realizar um estudo comparativo do desempenho de meta-heurísticas de enxame (PSO, GWO, WOA, SSA e CS) na otimização do posicionamento estático de múltiplos VANTs para cobertura de área, visando identificar as abordagens mais eficientes quanto à qualidade da solução e custo computacional.

1.1.2 Específicos

Para garantir o alcance do objetivo geral definido acima, foram traçados os seguintes objetivos específicos:

- Realizar uma revisão bibliográfica aprofundada sobre Veículos Aéreos Não Tripulados (VANTs), o problema de cobertura de área e os cinco algoritmos de otimização selecionados.
- Modelar matematicamente o problema de cobertura, estabelecendo uma função objetivo que considere a maximização da área monitorada, a minimização da sobreposição e a manutenção da conectividade de rede entre os VANTs.
- Implementar computacionalmente os algoritmos PSO, GWO, WOA, SSA e CS aplicados ao modelo desenvolvido, utilizando a linguagem Python para simulação dos cenários.
- Definir uma metodologia experimental sistemática, estabelecendo métricas de avaliação e cenários de teste que permitam a validação estatística dos resultados.
- Executar as simulações e realizar uma análise comparativa dos dados obtidos, discutindo o desempenho de cada algoritmo em termos de taxa de cobertura final, velocidade de convergência e estabilidade.

2 Fundamentação Teórica

Este capítulo estabelece o arcabouço teórico necessário para a compreensão e desenvolvimento da pesquisa. A discussão inicia-se com as características técnicas dos Veículos Aéreos Não Tripulados (VANTs), fundamentais para a aplicação proposta. Em seguida, aborda-se a teoria da otimização por meta-heurísticas, detalhando o conceito de Inteligência de Enxame. Por fim, o problema de cobertura de área é formalmente definido com suas restrições matemáticas e os algoritmos selecionados são descritos com suas respectivas formulações completas e mecanismos de controle.

2.1 Veículos Aéreos Não Tripulados (VANTs)

Os Veículos Aéreos Não Tripulados (VANTs), comumente conhecidos como drones, são aeronaves projetadas para operar sem um piloto a bordo, controladas remotamente ou por meio de sistemas autônomos. Originalmente desenvolvidos para fins militares, os VANTs passaram por uma rápida evolução tecnológica que permitiu sua ampla adoção em diversas aplicações civis, transformando setores como a engenharia, agricultura e monitoramento ambiental ([ALSAMHI et al., 2022](#)). A sua capacidade de embarcar sensores avançados tornou-os ferramentas poderosas para a aquisição de dados georreferenciados de alta resolução, reduzindo significativamente custos operacionais em relação às técnicas tradicionais de levantamento topográfico ([TSOUROS; BIBI; SARIGIANNIDIS, 2019](#); [ZHANG; KOVACS, 2012](#)). Esta versatilidade faz dos VANTs candidatos ideais para aplicações que requerem cobertura espacial eficiente, sendo particularmente valiosos em cenários onde o acesso terrestre é limitado ou perigoso.

2.1.1 Arquiteturas e Dinâmica de Voo

Para a operacionalização de missões de cobertura, é fundamental distinguir as plataformas aéreas disponíveis, cada uma com características e limitações específicas. A literatura classifica os VANTs em duas arquiteturas predominantes:

- **VANTs de Asa Fixa:** Utilizam sustentação aerodinâmica gerada pelo fluxo de ar nas asas. Essa eficiência aerodinâmica permite autonomias elevadas, variando de 1,5 a mais de 20 horas de voo contínuo, tornando-os ideais para varredura de grandes latifúndios e levantamentos em áreas extensas (acima de 100 hectares em um único voo). Entretanto, requerem espaços adequados para decolagem e pouso, necessitam manter velocidade constante para sustentação e não possuem capacidade de pairar (*hovering*) sobre pontos específicos ([ALSAMHI et al., 2022](#)). Essa limitação reduz

sua aplicabilidade em missões que requerem posicionamento estático para captura detalhada de áreas reduzidas.

- **VANTs Multi-rotor:** Sustentados pela propulsão direta de múltiplas hélices (ex: quadricópteros com 4 rotores, hexacópteros com 6 rotores), possuem capacidade de decolagem e pouso vertical (VTOL — *Vertical Takeoff and Landing*) e voo pairado estático em três dimensões. Esta característica permite manter uma posição fixa no espaço tridimensional (x, y, z) sem deslocamento horizontal, sendo essencial para a captura de imagens nadir (câmera apontada verticalmente para baixo, perpendicular ao solo) com composição consistente. Embora a autonomia seja limitada (tipicamente entre 20 a 40 minutos para quadricópteros comerciais), a capacidade de posicionamento estático e a manobrabilidade superior os tornam a plataforma ideal para o problema de posicionamento estático de sensores abordado neste trabalho (CABREIRA; BRISOLARA; FERREIRA, 2019).

2.1.2 Sensores para Cobertura de Área

A eficácia da missão é intrinsecamente ligada à carga útil (*payload*). Para o problema de cobertura, selecionam-se sensores baseados na natureza dos dados requeridos e nas características operacionais. Os principais sensores utilizados são:

- **Câmeras RGB:** Sensores passivos que capturam a radiação eletromagnética no espectro visível nas três bandas espectrais fundamentais: Vermelho (R, $\sim 630\text{--}700$ nm), Verde (G, $\sim 525\text{--}600$ nm) e Azul (B, $\sim 450\text{--}520$ nm). São a base para a fotogrametria digital, permitindo a geração de ortomosaicos e modelos tridimensionais texturizados através de técnicas de reconstrução estéreo. Suas principais vantagens incluem baixo custo, alta resolução espacial (frequentemente inferior a 2 cm/pixel em altitudes operacionais típicas) e facilidade de integração. As limitações primárias são a dependência de iluminação solar adequada, incapacidade de penetrar vegetação densa para mapear o solo, e sensibilidade a sombras e variações de iluminação (TSOUROS; BIBI; SARIGIANNIDIS, 2019; LUHMANN et al., 2013).
- **Sensores LIDAR:** O *Light Detection and Ranging* é um sensor ativo que emite pulsos laser de alta frequência (tipicamente na faixa de 850–1550 nm no infravermelho próximo) para medir distâncias via princípio *Time-of-Flight* (ToF). O mecanismo funciona pela medição do tempo decorrido entre a emissão do pulso e o retorno do sinal refletido. Diferente das câmeras RGB, o LIDAR é capaz de penetrar nos interstícios da copa das árvores, registrando múltiplos retornos do sinal (ecos sucessivos) conforme diferentes camadas de vegetação refletem a radiação laser. Esta propriedade permite filtrar a vegetação para gerar Modelos Digitais de Terreno (MDT) e Modelos Digitais de Elevação (MDE) com precisão métrica, sendo indispensável em topografia sob

florestas e ambientes com cobertura vegetal densa (ALSAMHI et al., 2022). A resolução horizontal típica varia entre 5 e 30 cm, dependendo da altitude e da especificação do equipamento.

- **Câmeras Multiespectrais:** Equipadas com filtros ópticos seletivos que capturam bandas espectrais específicas além do visível, notadamente o Infravermelho Próximo (NIR, $\sim 800\text{--}1000\text{ nm}$) e o *Red Edge* ($\sim 700\text{--}750\text{ nm}$). Esses dados possibilitam o cálculo de índices de vegetação, sendo o NDVI (*Normalized Difference Vegetation Index*) o mais utilizado, definido como $\text{NDVI} = \frac{\text{NIR} - \text{Red}}{\text{NIR} + \text{Red}}$. O NDVI permite diagnosticar a saúde fotossintética de culturas agrícolas, detectar deficiências nutricionais, avaliar estresse hídrico e mapear ciclos fenológicos antes que sintomas visuais sejam aparentes. Esta capacidade de detecção precoce torna as câmeras multiespectrais ferramentas valiosas em agricultura de precisão (CAVALCANTI et al., 2025; SANKARAN et al., 2015).

2.2 O Problema de Cobertura de Área

O Problema de Cobertura de Área (*Area Coverage Problem*) visa garantir que uma região de interesse seja monitorada ou sensoriada conforme requisitos operacionais. Este trabalho foca no cenário específico do **Posicionamento Estático de Sensores** (*Static Sensor Placement*), em que as posições dos sensores (drones) são determinadas e permanecem fixas durante a operação, diferindo de problemas de cobertura dinâmica ou *path planning* nos quais os sensores se movimentam (CABREIRA; BRISOLARA; FERREIRA, 2019).

2.2.1 Modelagem Matemática (Disco Binário)

O modelo adotado é o **Disco Binário**, um modelo determinístico que assume que um sensor cobre completamente qualquer ponto dentro de um raio r_s e não cobre nenhum ponto fora deste raio. A escolha deste modelo determinístico em detrimento de modelos probabilísticos (por exemplo, baseados em padrão de ruído e propagação de sinal) justifica-se pela natureza dos sensores ópticos (câmeras) focados no nadir (visada vertical), onde o Campo de Visão (FOV — *Field of View*) geométrico define uma área de cobertura efetiva clara e bem delimitada. O raio r_s é calculado a partir dos parâmetros ópticos da câmera (distância focal, tamanho do sensor) e da altitude de operação (TOSSA et al., 2022).

A função de cobertura $f(s_i, p)$ que indica se um sensor s_i cobre um ponto alvo p é formulada como:

$$f(s_i, p) = \begin{cases} 1 & \text{se } \|s_i - p\| \leq r_s \\ 0 & \text{caso contrário} \end{cases} \quad (2.1)$$

Onde:

- s_i é a posição do i -ésimo sensor no plano horizontal (projeção em 2D na altitude h).
- p é a posição de um ponto alvo na região de interesse.
- r_s é o raio de sensoriamento, determinado pela geometria óptica da câmera e altitude.
- $\|s_i - p\|$ denota a distância euclidiana entre sensor e ponto.

2.2.2 Restrições Operacionais

Para garantir a viabilidade operacional da solução, o problema de otimização é sujeito a restrições rígidas que refletem limitações práticas do sistema:

1. **Limites Geográficos:** Todos os drones devem permanecer dentro de uma área de operação permitida $A \subset \mathbb{R}^2$, definida por limites legais, restrições de terreno ou objetivos da missão. Esta restrição garante que nenhum veículo opera fora da jurisdição autorizada.

$$(x_i, y_i) \in A, \quad \forall i \in \{1, 2, \dots, N\} \quad (2.2)$$

Onde (x_i, y_i) são as coordenadas planares do drone i e N é o número total de VANTs.

2. **Sobreposição (*Overlap*):** Embora certa sobreposição seja necessária para a costura de imagens (fotogrametria necessita tipicamente 60–80% de *overlap* entre imagens adjacentes para reconstrução estéreo precisa), a sobreposição excessiva desperdiça recursos computacionais e área de cobertura útil. Define-se uma penalidade quando a distância euclidiana entre dois drones d_{ij} fica abaixo de um limiar de eficiência $d_{\min}$:

$$P_{\text{overlap}} = \sum_{i=1}^N \sum_{j=i+1}^N \max(0, d_{\min} - \|s_i - s_j\|) \quad (2.3)$$

Onde $d_{\min}$ é a distância mínima desejável entre drones (em função da sobreposição aceitável e do raio de cobertura).

3. **Conectividade de Rede:** Além de cobrir a área de interesse, o enxame precisa permanecer capaz de retransmitir os dados coletados até uma estação base, requisito recorrente em missões de monitoramento por múltiplos VANTs (TOSSA et al., 2022). Essa exigência é modelada como uma restrição de conectividade sobre um grafo

$G = (V, E)$, no qual cada drone corresponde a um vértice e uma aresta conecta dois drones i e j sempre que a distância euclidiana entre eles não exceder um raio de comunicação R_{comm} , tipicamente maior que o raio de sensoriamento r_s :

$$\text{conectado}(i, j) = \begin{cases} 1 & \text{se } \|s_i - s_j\| \leq R_{comm} \\ 0 & \text{caso contrário} \end{cases} \quad (2.4)$$

A conectividade indireta — em que dois drones fora do alcance direto um do outro ainda pertencem à mesma rede por meio de drones intermediários que retransmitem o sinal — é igualmente relevante, caracterizando uma rede de comunicação em múltiplos saltos (*multi-hop*). A viabilidade da missão está associada ao tamanho do maior subconjunto de drones mutuamente alcançáveis, direta ou indiretamente, dentro desse grafo: quanto menor esse subconjunto em relação ao total de drones, maior a fração de dados que a missão não conseguiria efetivamente recuperar.

2.2.3 Função Objetivo

O objetivo geral do problema de posicionamento consiste em determinar a configuração espacial do enxame de drones que maximize a cobertura efetiva da área de interesse, respeitando simultaneamente as restrições de sobreposição mínima e de conectividade de rede descritas na seção anterior. Esses três fatores (cobertura, sobreposição e conectividade) frequentemente competem entre si: maximizar a cobertura pode exigir posicionamentos que comprometam a conectividade em cenários com poucos drones disponíveis, assim como minimizar a sobreposição pode ser incompatível com a cobertura total em cenários com excesso de drones. O problema caracteriza-se, portanto, como uma otimização multi-objetivo, na qual nenhum dos três fatores pode ser tratado isoladamente sem considerar seu efeito sobre os demais.

A formulação matemática precisa da função objetivo, incluindo a normalização de cada componente ao intervalo $[0, 1]$ e sua agregação ponderada, é detalhada na metodologia deste trabalho (Seção 3.3), onde também é discutida a sensibilidade dessa formulação ao vetor de pesos escolhido.

2.3 Otimização por Meta-heurísticas

A otimização de cobertura de área por múltiplos agentes é um problema de natureza combinatória e de alta complexidade computacional. A maioria desses problemas pertence à classe de problemas NP-difícil, o que estabelece que não existe algoritmo determinístico capaz de encontrar a solução ótima em tempo polinomial (assumindo que $P \neq NP$) (GAREY; JOHNSON, 1979). Para contornar essa complexidade intratável, recorre-se às meta-heurísticas, que são estratégias de otimização de alto nível projetadas para explorar

o espaço de soluções de forma eficiente, encontrando soluções de alta qualidade em um tempo computacional razoável, ainda que sem garantia de otimalidade global (BLUM; ROLI, 2003).

O sucesso de uma meta-heurística reside em seu balanço entre dois mecanismos complementares: a **exploração** (do inglês *exploration*), que corresponde à busca em novas regiões do espaço de soluções para evitar convergência prematura a ótimos locais, e a **exploração** (do inglês *exploitation*), que refina a busca em regiões já identificadas como promissoras, concentrando esforços computacionais em vizinhanças de boas soluções. Este equilíbrio é crítico: excessiva exploração resulta em convergência lenta e consumo de recursos computacionais, enquanto excessiva exploração aumenta o risco de aprisionamento em ótimos locais (HUSSAIN et al., 2019).

A necessidade de comparar diferentes abordagens é fundamentada pelo teorema "No Free Lunch" (NFL), que postula que nenhum algoritmo de otimização é inerentemente superior para todos os tipos de problemas quando considerados em distribuição uniforme sobre todo o espaço de funções objetivo possíveis (WOLPERT; MACREADY, 1997). Esta constatação fundamental justifica o estudo comparativo de múltiplas meta-heurísticas para um problema específico, pois cada algoritmo pode apresentar desempenho superior em subconjuntos particulares do espaço de problemas.

2.3.1 Inteligência de Enxame

A Inteligência de Enxame (*Swarm Intelligence*, SI) é um paradigma da Inteligência Artificial inspirado no comportamento coletivo de sistemas descentralizados e auto-organizados observados na natureza. A premissa central é que comportamentos globais complexos e inteligentes podem emergir da população de agentes simples que seguem regras locais e determinísticas, sem a necessidade de um controle centralizado ou conhecimento global (BONABEAU; DORIGO; THERAULAZ, 1999).

Os princípios fundamentais da Inteligência de Enxame são:

- **Descentralização:** Não existe uma autoridade central coordenando as ações. Cada agente toma decisões baseado exclusivamente em informações locais.
- **Auto-organização:** A estrutura global emerge sem plano pré-definido. A ordem emerge dos padrões locais de interação.
- **Comportamento Emergente:** Dinâmicas coletivas complexas surgem de regras simples, permitindo adaptação a mudanças ambientais.
- **Robustez:** A perda de indivíduos não compromete o funcionamento do sistema como um todo.

Exemplos naturais abundam: revoadas de pássaros que realizam manobras coordenadas sem liderança, colônias de formigas que encontram caminhos ótimos para alimento através da comunicação química (feromônios), e cardumes de peixes que coordenam movimento evasivo contra predadores. Estes sistemas inspiram algoritmos como *Particle Swarm Optimization* (PSO), baseado em revoadas de aves, *Ant Colony Optimization* (ACO), baseado em feromônios de formigas, e muitos outros.

2.4 Algoritmos de Enxame Seleccionados

A seguir, são descritos em detalhe os cinco algoritmos seleccionados para este estudo comparativo, detalhando suas inspirações biológicas, formulações matemáticas completas, definições de parâmetros com valores recomendados da literatura, e explicitando os mecanismos pelos quais cada algoritmo controla o balanço entre exploração e exploração.

2.4.1 Particle Swarm Optimization (PSO)

Inspirado no voo coordenado de revoadas de pássaros e no comportamento social de cardumes de peixes, o PSO modela um enxame de partículas que se deslocam no espaço de soluções, tendo como guias a melhor posição pessoal visitada e a melhor posição global do enxame.

Formulação Matemática: A atualização de velocidade e posição no tempo t segue a formulação do PSO com peso de inércia, variante amplamente adotada do algoritmo original (KENNEDY; EBERHART, 1995):

$$\vec{v}_i^{t+1} = w\vec{v}_i^t + c_1r_1^t (\vec{pbest}_i - \vec{x}_i^t) + c_2r_2^t (\vec{gbest} - \vec{x}_i^t) \quad (2.5)$$

$$\vec{x}_i^{t+1} = \vec{x}_i^t + \vec{v}_i^{t+1} \quad (2.6)$$

Definição de Parâmetros:

- **w (Peso de Inércia):** Controla o impacto da velocidade anterior sobre o deslocamento atual. Um w elevado (próximo a 1) privilegia a inércia, favorecendo exploração e movimento em maior escala. Um w baixo (próximo a 0) reduz a influência da velocidade anterior, favorecendo atração para as melhores posições. Valores típicos variam de 0.9 (início, exploração) decrescendo para 0.4 (fim, exploração). A estratégia mais comum é o decrescimento linear: $w(t) = w_{\max} - \frac{w_{\max} - w_{\min}}{T_{\max}} \cdot t$.
- **c_1 (Coeficiente Cognitivo):** Pondera a atração da partícula para sua melhor posição pessoal ($\vec{pbest}_i$). Representa a confiança da partícula em sua própria experiência.

- **c_2 (Coeficiente Social):** Pondera a atração da partícula para a melhor posição global ($g\vec{best}$). Representa a confiança da partícula na experiência coletiva do enxame.
- **Valores Adotados:** Neste trabalho, adotou-se o esquema de peso de inércia com decrescimento linear de $w = 0.9$ (início, exploração) para $w = 0.4$ (fim, exploração), conforme originalmente proposto por (SHI; EBERHART, 1998), com coeficientes $c_1 = c_2 = 1.5$. Trata-se de uma configuração empírica consolidada na literatura, aplicada sem qualquer ajuste específico ao problema de cobertura, de modo a preservar a equidade da comparação entre os algoritmos. Configurações alternativas com garantias de convergência matemática, como $c_1 = c_2 = 1.496172$ e $w = 0.72984$ (CHIARADONNA; BONDAVALLI; GIANDOMENICO, 2020), também são reportadas na literatura.
- **$\mathbf{r}_1^t, \mathbf{r}_2^t$:** Vetores de números aleatórios uniformemente distribuídos em $[0, 1]$, regenerados a cada iteração. Introduzem estocasticidade, garantindo que a busca não seja determinística.

Mecanismo de Exploração vs. Exploração: O controle do balanço é realizado principalmente pelo parâmetro w . Quando w é grande, a influência da velocidade anterior domina, permitindo que partículas continuem deslocando-se em direções já exploradas (exploração global). Quando w diminui, a atração para $p\vec{best}_i$ e $g\vec{best}$ torna-se dominante, concentrando a busca em regiões promissoras (exploração local). Além disso, c_1 elevado aumenta a divergência (cada partícula segue seu próprio caminho), favorecendo exploração; c_2 elevado aumenta a convergência (partículas puxadas para $g\vec{best}$), favorecendo exploração.

2.4.2 Grey Wolf Optimizer (GWO)

O GWO simula a dinâmica hierárquica de uma alcateia de lobos, onde existe uma estrutura social bem definida: um líder alfa (α), segundo-em-comando beta (β), e terceira hierarquia delta (δ), sendo os demais lobos denominados ômega (ω). Apenas o líder alfa toma a maioria das decisões, enquanto os demais seguem e aprendem.

Formulação Matemática Completa:

Os vetores de coeficientes $\vec{A}$ e $\vec{C}$ controlam o comportamento de cerco (*encirclement*) da presa:

$$\vec{A} = 2\vec{a} \circ \vec{r}_1 - \vec{a} \quad (2.7)$$

$$\vec{C} = 2\vec{r}_2 \quad (2.8)$$

Onde $\vec{a}$ é um vetor que decresce linearmente de 2 a 0 ao longo das iterações, $\vec{r}_1$ e $\vec{r}_2$ são vetores aleatórios em $[0, 1]$, e $\circ$ denota multiplicação elemento-a-elemento.

A distância entre o lobo e a presa (melhor solução) é calculada como:

$$\vec{D} = |\vec{C} \circ \vec{X}_p(t) - \vec{X}(t)| \quad (2.9)$$

A posição é então atualizada como:

$$\vec{X}(t+1) = \vec{X}_p(t) - \vec{A} \circ \vec{D} \quad (2.10)$$

Onde $\vec{X}_p(t)$ é a posição da presa (neste caso, atualizada em relação a três candidatos: α, β, δ).

Atualização com Três Líderes:

Na prática, a posição de cada lobo ω é atualizada como a média ponderada das posições sugeridas pelos três líderes:

$$\vec{X}_\alpha(t+1) = \vec{X}_\alpha(t) - \vec{A}_1 \circ (\vec{D}_\alpha) \quad (2.11)$$

$$\vec{X}_\beta(t+1) = \vec{X}_\beta(t) - \vec{A}_2 \circ (\vec{D}_\beta) \quad (2.12)$$

$$\vec{X}_\delta(t+1) = \vec{X}_\delta(t) - \vec{A}_3 \circ (\vec{D}_\delta) \quad (2.13)$$

$$\vec{X}(t+1) = \frac{\vec{X}_\alpha(t+1) + \vec{X}_\beta(t+1) + \vec{X}_\delta(t+1)}{3} \quad (2.14)$$

Definição de Parâmetros:

- **a (Parâmetro de Decrescimento):** Decresce linearmente de 2.0 a 0.0 ao longo das iterações: $\vec{a} = 2 - \frac{2t}{T_{\max}}$. Este parâmetro controla diretamente o tamanho da região de busca.
- **Vetores $\vec{A}$ e $\vec{C}$:** Determinam como os lobos se movem. O vetor $\vec{A}$ é crucial para o balanço exploração-exploração.

Mecanismo de Exploração vs. Exploração:

O controle do balanço é realizado primariamente pelo vetor $\vec{A}$:

- Quando $|\vec{A}| > 1$: O lobo é forçado a se afastar da posição atual em uma direção aleatória, facilitando **exploração** de novas regiões.
- Quando $|\vec{A}| < 1$: O lobo é forçado a se aproximar da presa de forma contínua, favorecendo **exploração** de regiões promissoras.

Além disso, como $\vec{a}$ decresce de 2 a 0, os valores de $|\vec{A}|$ também decrescem probabilisticamente ao longo das iterações, implementando uma estratégia de exploração inicial seguida de exploração final (MIRJALILI; MIRJALILI; LEWIS, 2014).

2.4.3 Whale Optimization Algorithm (WOA)

O WOA simula o comportamento de caça da baleia-jubarte (*Humpback Whale*, *Megaptera novaeangliae*), que emprega uma estratégia conhecida como *bubble-net feeding* (alimentação em rede de bolhas). A baleia circunda o cardume de presas criando uma rede de bolhas em espiral logarítmica, confinando a presa, antes de avançar verticalmente através do centro para alimentar-se.

Formulação Matemática Completa:

A atualização de posição alterna entre dois mecanismos: cerco direto e movimento espiral logarítmico:

$$\vec{X}(t+1) = \begin{cases} \vec{X}^*(t) - \vec{A} \circ \vec{D} & \text{se } p < 0.5 \\ \vec{D}' \cdot e^{bl} \cdot \cos(2\pi l) + \vec{X}^*(t) & \text{se } p \geq 0.5 \end{cases} \quad (2.15)$$

Onde:

- $\vec{X}^*(t)$ é a melhor posição encontrada até o momento (localização da presa).
- $\vec{D} = |\vec{C} \circ \vec{X}^*(t) - \vec{X}(t)|$ é a distância até a presa (para o caso de cerco direto).
- $\vec{D}' = |\vec{X}^*(t) - \vec{X}(t)|$ é a distância para o movimento espiral.
- $\vec{A} = 2\vec{a} \circ \vec{r} - \vec{a}$, onde $\vec{a}$ decresce de 2 a 0.
- $\vec{C} = 2\vec{r}$, onde $\vec{r}$ é aleatório em $[0, 1]$.
- p é um número aleatório em $[0, 1]$ que determina qual mecanismo usar.
- $b = 1$ é uma constante que define o formato logarítmico da espiral.
- l é um número aleatório em $[-1, 1]$ que controla a amplitude da espiral.

Mecanismo de Exploração vs. Exploração:

- **Quando $p < 0.5$** (probabilidade 50%): O algoritmo executa **cerco direto** (*shrinking encircling*). Conforme $\vec{a}$ decresce, $|\vec{A}|$ também decresce, reduzindo o tamanho do passo. Isto implementa progressivamente uma **exploração refinada**.
- **Quando $p \geq 0.5$** (probabilidade 50%): O algoritmo executa **movimento espiral** (*spiral updating position*). O termo e^{bl} cria uma trajetória logarítmica, permitindo que a baleia se aproxime da presa de forma não-linear. Este mecanismo também favorece **exploração**, mas com um padrão de movimento diferente que pode escapar de ótimos locais rasos.

(MIRJALILI; LEWIS, 2016)

2.4.4 Salp Swarm Algorithm (SSA)

O SSA baseia-se na observação do comportamento de salpas, pequenos animais marinhos que se deslocam em grupos coordenados em uma estrutura de cadeia hierárquica. A salpa líder navega em direção à fonte de alimento, enquanto as demais (seguidoras) seguem mantendo uma distância relativa, implementando um padrão de comunicação simples mas efetivo.

Formulação Matemática Completa:

A atualização da posição da salpa líder (índice 1) em cada dimensão j é:

$$x_j^1(t+1) = \begin{cases} F_j + c_1 [(UB_j - LB_j)c_2 + LB_j] & \text{se } c_3 \geq 0.5 \\ F_j - c_1 [(UB_j - LB_j)c_2 + LB_j] & \text{se } c_3 < 0.5 \end{cases} \quad (2.16)$$

As salps seguidoras ($i \geq 2$) atualizam sua posição seguindo a Lei de Newton (aceleração em direção à salpa anterior):

$$x_j^i(t+1) = \frac{1}{2} (x_j^i(t) + x_j^{i-1}(t+1)) \quad (2.17)$$

Definição de Parâmetros:

- **F (Fonte de Alimento):** Representa a melhor solução encontrada até o momento (análogo ao $gbest$ do PSO).
- **c_1 (Parâmetro de Decrescimento):** É o parâmetro mais crítico do SSA. Define-se como:

$$c_1(t) = 2 \cdot e^{-\left(\frac{4t}{T_{\max}}\right)^2} \quad (2.18)$$

Decresce exponencialmente de 2 (no início) para próximo de 0 (no fim), balanceando exploração inicial com exploração final.

- **c_2 (Parâmetro Aleatório):** Número aleatório uniforme em $[0, 1]$ regenerado a cada iteração. Adiciona estocasticidade à busca.
- **c_3 (Parâmetro Probabilístico):** Número aleatório em $[0, 1]$ que determina a direção da atualização (positiva ou negativa com probabilidade 50%).
- **UB e LB (Limites Superior e Inferior):** Definem os limites do espaço de busca para cada dimensão.

Mecanismo de Exploração vs. Exploração:

- No **início** (quando $c_1 \approx 2$): O termo $c_1 [(UB_j - LB_j)c_2 + LB_j]$ é grande, permitindo grandes saltos no espaço de soluções. Isto favorece **exploração global**.
- No **fim** (quando $c_1 \approx 0$): Os deslocamentos tornam-se pequenos, concentrando a busca em regiões já exploradas. Isto favorece **exploração local**.
- O mecanismo de *chain following* das seguidoras amplifica a convergência: uma vez que a líder se aproxima de uma boa solução, todas as seguidoras também se aproximam, implementando um comportamento de reforço coletivo.

(MIRJALILI et al., 2017)

2.4.5 Cuckoo Search (CS)

O CS baseia-se no parasitismo de ninhada de cucos, pássaros que depositam seus ovos em ninhos de outras espécies. Um cuco bem-sucedido é aquele cujo ovo é semelhante aos ovos do hospedeiro, evitando detecção. O algoritmo modela este comportamento através de: (1) descoberta de ninhos (exploração), (2) uso de voos de Lévy para busca global eficiente, e (3) descarte seletivo de piores soluções.

Formulação Matemática Completa:

A geração de novas soluções utiliza o mecanismo de *Lévy flight*:

$$\vec{x}_i^{(t+1)} = \vec{x}_i^{(t)} + \alpha \oplus L(\lambda) \quad (2.19)$$

Onde:

- α é o tamanho do passo (*step size*), que escalona os deslocamentos à dimensão do espaço de busca; neste trabalho adotou-se $\alpha = 0.01$.

- $L(\lambda)$ é a distribuição de voo de Lévy com parâmetro λ . A distribuição de Lévy possui cauda pesada (*heavy tail*), permitindo ocasionalmente saltos muito grandes, alternados com movimentos pequenos.
- $\oplus$ denota multiplicação elemento-a-elemento (operador de Hadamard).

Implementação Prática: Algoritmo de Mantegna

A distribuição de Lévy é implementada na prática através do algoritmo de Mantegna, que gera passos $Lévy(\lambda)$ como:

$$Lévy(\lambda) = \frac{u}{|v|^{1/\beta}} \quad (2.20)$$

Onde u e v são distribuições normais padrão:

$$u \sim \mathcal{N}(0, \sigma_u^2), \quad v \sim \mathcal{N}(0, 1) \quad (2.21)$$

E:

$$\sigma_u = \left[\frac{\Gamma(1 + \beta) \sin(\pi\beta/2)}{\Gamma((1 + \beta)/2) \beta 2^{(\beta-1)/2}} \right]^{1/\beta} \quad (2.22)$$

Onde Γ é a função Gamma e β é tipicamente 1.5. Este procedimento gera uma distribuição com cauda pesada que aproxima a dinâmica dos voos de Lévy biológicos.

Mecanismo de Descoberta e Descarte:

Em cada iteração, para cada ninho i :

1. Gera-se uma nova solução $\vec{x}_i^{\text{novo}}$ usando o voo de Lévy.
2. Com probabilidade P_a (probabilidade de descoberta, típico 0.25), o ninho i é descoberto e abandonado, sendo substituído por uma nova solução aleatória.
3. O ninho i é mantido se $\vec{x}_i^{\text{novo}}$ for melhor que a solução anterior. Caso contrário, mantém-se a solução anterior.

Formalmente:

$$\vec{x}_i^{(t+1)} = \begin{cases} \vec{x}_i^{(t)} + \alpha \cdot L(\lambda) & \text{com probabilidade } (1 - P_a) \\ \text{aleatório em } [LB, UB] & \text{com probabilidade } P_a \end{cases} \quad (2.23)$$

Definição de Parâmetros:

- **α (Tamanho do Passo):** Controla a amplitude dos deslocamentos gerados pelos voos de Lévy. Adotou-se $\alpha = 0.01$, valor que escalona o passo à dimensão do espaço de busca do problema; trata-se de uma calibração de escala, e não de um ajuste de desempenho específico ao problema. Um α maior permite exploração mais agressiva; um α menor concentra a busca localmente.
- **P_a (Probabilidade de Descoberta):** Fração de ninhos que são abandonados e substituídos em cada iteração. Valor típico é $P_a = 0.25$, significando que 25% dos piores ninhos são descartados. Este parâmetro controla a diversidade da população.
- **β (Parâmetro de Lévy):** Tipicamente $\beta = 1.5$. Valores menores (próximo a 1) produzem caudas mais pesadas (maiores saltos ocasionais); valores maiores (próximo a 2) produzem distribuição mais próxima do Gaussiano.

Mecanismo de Exploração vs. Exploração:

- **Lévy Flights (Exploração Global):** O mecanismo de voos de Lévy permite ocasionalmente saltos muito grandes, permitindo escape de ótimos locais e exploração global eficiente. A cauda pesada garante que a busca não fica presa em uma região.
- **Descoberta P_a (Renovação de Diversidade):** A probabilidade de descoberta garante que uma fração fixa da população é renovada a cada iteração, evitando convergência prematura e mantendo diversidade.
- **Balanco Dinâmico:** Diferente do PSO e GWO que reduzem exploração ao longo das iterações, o CS mantém um balanço relativo entre exploração (Lévy flights + descoberta) e exploração (seleção dos melhores ninhos) ao longo de todo o processo.

([YANG; DEB, 2009](#))

Para consolidar a configuração experimental, o Quadro 1 reúne os valores de todos os hiperparâmetros efetivamente adotados na implementação dos cinco algoritmos. Todos correspondem a configurações consolidadas na literatura, mantidas sem ajuste individual a cada algoritmo, de modo a assegurar a equidade da comparação.

Resumo Comparativo dos Algoritmos

Os cinco algoritmos analisados apresentam estratégias distintas para balancear exploração e exploração. O Quadro 2 sintetiza as características principais:

Algoritmo	Parâmetro	Valor
PSO	w	$0,9 \rightarrow 0,4$ (linear)
PSO	c_1	1,5
PSO	c_2	1,5
GWO	a	$2 \rightarrow 0$ (linear)
WOA	a	$2 \rightarrow 0$ (linear)
WOA	b	1
SSA	$c_1(t)$	$2e^{-(4t/T_{\max})^2}$
CS	α	0,01
CS	P_a	0,25
CS	β	1,5
Todos	Agentes / Iterações	30 / 100

Fonte: Elaborado pelo autor (2026).

Quadro 1 – Hiperparâmetros adotados na implementação dos algoritmos

Algoritmo	Inspiração	Parâmetro Chave	Estratégia
PSO	Bando de pássaros	Inércia w	Decrescente (explor. $\rightarrow$ explo.)
GWO	Alcateia de lobos	Vetor $\vec{A}$	Decrescente (explor. $\rightarrow$ explo.)
WOA	Baleia-jubarte	Parâmetro a	Misto: cerco + espiral
SSA	Cadeia de salpas	Decrescimento c_1	Decrescente (explor. $\rightarrow$ explo.)
CS	Parasitismo de cuco	Descoberta P_a	Constante (voos Lévy)

Fonte: Elaborado pelo autor (2026).

Quadro 2 – Comparação dos Mecanismos de Exploração-Exploração

3 Metodologia

Este capítulo detalha a metodologia empregada na condução do estudo comparativo proposto. A abordagem foi estruturada a fim de garantir a validade e a reprodutibilidade da pesquisa, abrangendo a definição formal do problema, o desenvolvimento do ambiente de simulação e o delineamento experimental. São detalhados a seguir a modelagem matemática e a função custo normalizada, os parâmetros fixos e variáveis que compõem os cenários de teste, a estratégia de avaliação sob múltiplos perfis de prioridade operacional e os critérios estatísticos utilizados para a análise de desempenho dos algoritmos.

3.1 Abordagem da Pesquisa

Este trabalho caracteriza-se como uma pesquisa quantitativa, de natureza aplicada, que utilizou a simulação computacional como método de procedimento para a realização de um estudo comparativo de algoritmos de otimização. Os algoritmos avaliados foram o *Particle Swarm Optimization* (PSO), o *Grey Wolf Optimizer* (GWO), o *Whale Optimization Algorithm* (WOA), o *Salp Swarm Algorithm* (SSA) e o *Cuckoo Search* (CS), comparados entre si e contra um método de posicionamento aleatório, adotado como linha de base (*baseline*) para quantificar o ganho efetivo proporcionado pela otimização.

3.2 Modelo do Problema e Ambiente de Simulação

O problema de pesquisa foi definido como o **Problema de Posicionamento de Sensores** (*Sensor Placement Problem*), categoria estabelecida na literatura para problemas nos quais sensores devem ser posicionados de forma estática, sem consideração de trajetória, para maximizar a cobertura de uma região (AKBARZADEH et al., 2014). Essa categorização é o que delimita formalmente o escopo do trabalho: os drones são tratados como plataformas de sensoriamento fixas a partir do momento em que atingem sua posição final, e não como veículos cuja trajetória de deslocamento precisa ser otimizada. É essa mesma delimitação que fundamenta a exclusão do consumo de bateria da função custo (Seção 3.3) e a distinção do trabalho frente a problemas de planejamento de rotas (*path planning*), nos quais o deslocamento em si é o objeto de otimização.

O ambiente de simulação foi desenvolvido na linguagem de programação Python (versão 3.12), com o suporte das bibliotecas NumPy (operações vetorizadas e álgebra matricial), Matplotlib (geração das visualizações), Pandas (organização e exportação tabular dos resultados), SciPy (teste estatístico de Wilcoxon e função Gamma empregada no Cuckoo Search) e Joblib (paralelização das execuções independentes). O posicionamento

de cada drone foi representado por suas coordenadas planares (x, y) , de modo que uma solução candidata completa corresponde ao vetor de $2N$ variáveis que descreve as posições de todos os N drones simultaneamente.

A altitude de operação não foi tratada como variável de otimização, e sim como parâmetro de entrada definido por cenário, mantendo o problema estritamente bidimensional. Essa decisão é coerente com a natureza do problema de cobertura por sensores ópticos em visada nadir, no qual a altitude determina o raio de cobertura projetado no solo (Seção 3.4.1), mas não constitui, em si, um grau de liberdade da busca.

3.3 Função Custo e Normalização

A qualidade das soluções foi avaliada por uma função custo multi-objetivo, formulada para minimizar simultaneamente fatores conflitantes. A função agregou três componentes: (1) a área não coberta pelo enxame, (2) a sobreposição (*overlap*) entre as áreas de cobertura dos drones e (3) a fragmentação da rede de comunicação entre os veículos.

O terceiro componente (conectividade de rede) substituiu o modelo de consumo de bateria considerado na concepção inicial do trabalho. A substituição justifica-se conceitualmente: o consumo de bateria está associado ao deslocamento dos veículos, métrica pertinente a problemas de planejamento de rotas, mas pouco significativa em um problema de posicionamento estático, no qual se avalia a configuração final e não a trajetória percorrida. A conectividade de rede, por outro lado, é uma restrição diretamente relevante ao posicionamento estático: os drones precisam permanecer mutuamente alcançáveis para que os dados coletados possam ser retransmitidos até a estação base, requisito recorrente em missões de monitoramento por enxames de VANTs (TOSSA et al., 2022).

Cada componente, por possuir natureza e escala distintas — uma fração de área, uma soma de distâncias e uma proporção de drones conectados —, foi normalizado para o intervalo $[0, 1]$ antes da agregação ponderada. Essa normalização é fundamental em otimização multi-objetivo para evitar que um único objetivo, em virtude de sua magnitude numérica, domine o processo de busca (DEB, 2014). A função custo total foi definida como a soma ponderada dos três componentes normalizados:

$$\text{Custo}(S) = w_c \cdot c_1(S) + w_o \cdot c_2(S) + w_r \cdot c_3(S) \quad (3.1)$$

onde c_1 , c_2 e c_3 são, respectivamente, os componentes normalizados de cobertura, sobreposição e conectividade, e w_c , w_o e w_r são os pesos associados a cada um, com $w_c + w_o + w_r = 1$.

3.3.1 Componente de Cobertura

O componente de cobertura foi definido como o complemento da taxa de cobertura, de modo que sua minimização equivale à maximização da área efetivamente coberta. Diferentemente dos dois componentes seguintes, este não exigiu uma etapa adicional de normalização: a taxa de cobertura $C(S)$ é, por construção, uma proporção — a razão entre a área coberta e a área total —, estando naturalmente contida no intervalo $[0, 1]$. Bastou, portanto, tomar seu complemento para obter um termo de custo (a minimizar) a partir de um termo de benefício (a maximizar):

$$c_1(S) = 1 - C(S) \quad (3.2)$$

onde $C(S)$ é a taxa de cobertura definida na Seção 2.2, obtida pela razão entre a área coberta pela união dos discos de sensoriamento e a área total da região de interesse. Para o cálculo computacional dessa razão, a área foi discretizada em uma grade regular de pontos, e a taxa de cobertura aproximada pela fração de pontos da grade situados dentro do raio de cobertura de ao menos um drone, conforme o modelo de Disco Binário (Equação 2.1).

3.3.2 Componente de Sobreposição

O componente de sobreposição penalizou pares de drones cuja distância euclidiana ficou abaixo de um limiar mínimo de eficiência $d_{\min}$, conforme a penalidade P_{overlap} definida na Equação 2.3. Para sua normalização ao intervalo $[0, 1]$, adotou-se como referência o pior caso teórico, no qual todos os N drones estariam sobrepostos em um mesmo ponto:

$$c_2(S) = \frac{P_{\text{overlap}}(S)}{P_{\text{overlap}}^{\max}}, \quad \text{com} \quad P_{\text{overlap}}^{\max} = \frac{N(N-1)}{2} \cdot d_{\min} \quad (3.3)$$

onde $\frac{N(N-1)}{2}$ é o número total de pares de drones e $d_{\min} = 2 \cdot r_s$ é a distância mínima desejável entre dois drones para que seus discos de cobertura não se sobreponham.

3.3.3 Componente de Conectividade

A conectividade entre os drones foi modelada como um grafo, no qual dois drones foram considerados diretamente conectados quando a distância euclidiana entre eles foi menor ou igual a um raio de comunicação R_{comm} . A partir desse grafo, identificou-se o maior componente conexo, isto é, o maior conjunto de drones mutuamente alcançáveis, direta ou indiretamente. A consideração de alcançabilidade indireta é relevante: dois drones fora do alcance direto um do outro podem ainda integrar a mesma rede caso exista um caminho de retransmissão através de drones intermediários, comportamento característico de redes

de comunicação em múltiplos saltos (*multi-hop*). A identificação do maior componente conexo foi realizada por meio da estrutura de dados *Union-Find* (*Disjoint Set Union*).

O componente de custo foi definido como a fração de drones situados fora do maior componente conexo:

$$c_3(S) = 1 - \frac{|G_{max}(S)|}{N} \quad (3.4)$$

onde $|G_{max}(S)|$ é o número de drones pertencentes ao maior grupo conectado. Assim, $c_3 = 0$ indica que todos os drones integram uma única rede, e valores maiores indicam fragmentação crescente, condição na qual parte dos dados coletados não poderia ser efetivamente retransmitida até a base.

Optou-se deliberadamente por um modelo binário de conectividade — um par de drones está conectado ou não está, sem gradação intermediária —, em detrimento de um modelo alternativo no qual a força do sinal decairia continuamente com a distância até um ponto de perda total de conexão. Embora esse segundo modelo represente de forma mais realista a física da comunicação sem fio, sua adoção reabriria um acoplamento entre os componentes de cobertura e conectividade que este trabalho buscou deliberadamente evitar: caso um drone “perdido” da rede deixasse de contribuir para a cobertura, os componentes c_1 e c_3 deixariam de ser independentes, comprometendo a interpretação isolada de cada um. A adoção do modelo graduado exigiria, ainda, a calibração de um parâmetro adicional (a taxa de decaimento do sinal), sem garantia de que essa complexidade adicional se traduzisse em ganho analítico proporcional. Por esses motivos, o modelo binário foi mantido para o escopo deste trabalho, e a extensão para um modelo de conectividade graduada é indicada como direção de trabalho futuro (Seção 5.3).

3.4 Delineamento Experimental

Para conduzir uma análise comparativa abrangente, foi definida uma matriz experimental estruturada a partir da variação de dois fatores (a escala da área e a densidade de drones), combinada com a avaliação dos algoritmos sob múltiplos perfis de prioridade operacional (Seção 3.4.3).

3.4.1 Parâmetros Fixos

Os seguintes parâmetros foram mantidos constantes em todas as simulações, para garantir a comparabilidade dos resultados:

- **Ponto de partida:** fixo na coordenada $[0, 0]$, utilizado como referência inicial comum a todos os algoritmos.

- **Câmera de referência:** adotou-se um campo de visão horizontal (HFOV) de 75° , compatível com câmeras de mapeamento embarcadas em VANTs comerciais. O raio de cobertura projetado no solo foi derivado da relação fotogramétrica $r_s = h \cdot \tan(\text{HFOV}/2)$, onde h é a altitude de operação (LUHMANN et al., 2013).
- **Raio de cobertura por escala de área:** diferentemente da concepção inicial, que adotava um raio fixo de 15 metros para todas as escalas, o raio de cobertura foi ajustado proporcionalmente à escala da área, por meio da altitude de operação. Essa decisão garante que a relação entre o alcance do sensor e a dimensão da área permaneça coerente entre os cenários — um raio fixo de 15 metros seria desproporcionalmente pequeno em uma área de 500×500 metros, inviabilizando cobertura significativa com um número operacionalmente razoável de drones. A alternativa de manter altitude fixa e compensar unicamente pelo aumento do número de drones foi descartada por exigir, nas áreas maiores, uma quantidade de veículos incompatível com missões reais de VANTs. Ressalta-se que essa decisão não simplifica artificialmente o problema de otimização: a dimensionalidade do espaço de busca ($2N$ variáveis) escala proporcionalmente ao número de drones em qualquer um dos cenários, de modo que a dificuldade combinatória enfrentada pelos algoritmos permanece comparável entre escalas. Os valores adotados constam na Tabela 1.
- **Raio de comunicação:** definido proporcionalmente ao raio de cobertura, segundo $R_{comm} = k \cdot r_s$, com $k = 2,5$. A calibração desse fator é detalhada na Seção 3.4.4.
- **Resolução da grade de discretização:** ajustada por escala de área (1 m, 2 m e 5 m, respectivamente para 100×100 , 250×250 e 500×500 metros), de modo a equilibrar a precisão da estimativa de cobertura com o custo computacional, que cresceria proibitivamente caso se mantivesse resolução de 1 metro nas áreas maiores.
- **Hiperparâmetros dos algoritmos:** adotaram-se os valores consagrados na literatura, conforme detalhado na Seção 2.4, a fim de garantir uma comparação justa entre as abordagens.
- **Número de agentes e iterações:** todos os algoritmos foram executados com 30 agentes e 100 iterações por execução.

Tabela 1 – Parâmetros ópticos por escala de área

Área (m)	Altitude (m)	Raio de cobertura r_s (m)
100×100	20	15
250×250	50	38
500×500	100	75

Fonte: Elaborado pelo autor (2026).

3.4.2 Parâmetros de Cenário (Variáveis Independentes)

Os experimentos foram estruturados em torno da variação de dois fatores, com o objetivo de avaliar a escalabilidade e a sensibilidade dos algoritmos:

- **Escala da área:** três escalas — 100×100 m, 250×250 m e 500×500 m.
- **Densidade de drones:** para cada escala, três densidades — escassa, suficiente e redundante.

A densidade suficiente foi estimada com base na cobertura mínima necessária para a área, em analogia ao problema clássico de empacotamento de círculos, cuja complexidade combinatória é da mesma natureza da discutida em (GAREY; JOHNSON, 1979). As demais densidades foram definidas proporcionalmente, com a categoria escassa representando recursos limitados e a redundante, recursos em excesso. A combinação das três escalas com as três densidades resultou em nove cenários experimentais distintos.

Cada densidade impõe um regime analítico distinto. No cenário escasso, a capacidade total de sensoriamento do enxame é inferior à área, de modo que a cobertura máxima alcançável é limitada pela quantidade de recursos. Avalia-se, portanto, a capacidade de cada algoritmo de extrair a maior cobertura possível com poucos drones. No cenário redundante, a capacidade total excede a área, e o fator determinante passa a ser a eficiência na disposição dos drones, isto é, a capacidade de minimizar sobreposições desnecessárias. O cenário suficiente representa o regime intermediário, no qual os três componentes da função custo exercem influência mais equilibrada.

3.4.3 Perfis de Peso e Análise de Robustez

A função custo definida na Equação 3.1 emprega o método da soma ponderada (*weighted sum method*) para agregar os três objetivos. É propriedade conhecida desse método que a solução ótima obtida depende estruturalmente do vetor de pesos escolhido: diferentes ponderações podem conduzir a soluções distintas e, conseqüentemente, a diferentes ordenações de desempenho entre os algoritmos (MARLER; ARORA, 2010). A adoção de um único vetor de pesos arbitrário tornaria, portanto, as conclusões do estudo dependentes dessa escolha específica, fragilizando sua generalidade.

Para contornar essa limitação de forma metodologicamente fundamentada, optou-se por não fixar um único conjunto de pesos, e sim avaliar os algoritmos sob três perfis de peso distintos, cada um representando uma prioridade operacional de missão plausível. Os três perfis foram mantidos fixos e aplicados de maneira idêntica a todos os algoritmos e a todos os cenários, de modo que a comparação entre algoritmos ocorresse sempre sob condições equivalentes. Os perfis adotados constam na Tabela 2.

Tabela 2 – Perfis de peso da função custo

Perfil	w_c	w_o	w_r	Interpretação operacional
A — Cobertura prioritária	0,7	0,2	0,1	Missões de mapeamento, nas quais a cobertura total é o objetivo dominante
B — Transição	0,5	0,2	0,3	Regime intermediário, com leve predominância da cobertura
C — Rede prioritária	0,3	0,2	0,5	Missões nas quais a manutenção da rede de comunicação é crítica

Fonte: Elaborado pelo autor (2026).

A construção dos perfis seguiu dois princípios. Primeiro, o peso da sobreposição foi mantido constante (0, 2) nos três perfis, por representar um critério de eficiência transversal a qualquer missão e não uma prioridade operacional concorrente. A sobreposição excessiva é indesejável independentemente do objetivo da missão; variá-la introduziria um grau de liberdade adicional que obscureceria o *trade-off* central em estudo, qual seja, o equilíbrio entre cobertura e conectividade. Assim, a variação concentrou-se exclusivamente entre os pesos de cobertura e conectividade, ajustados de forma simétrica em incrementos de 0, 2. Segundo, manteve-se em todos os perfis a hierarquia na qual o peso da cobertura nunca é inferior ao da sobreposição: mesmo no perfil de rede prioritária, a cobertura permanece como componente de peso não desprezível (0, 3), refletindo que a conectividade é tratada como restrição crítica de uma missão que continua sendo, em essência, de cobertura, e não como objetivo que a substitua.

Definida a avaliação sob múltiplos perfis, o critério de desempenho deixou de ser a identificação do algoritmo vencedor sob uma única ponderação, e passou a privilegiar a robustez, isto é, a consistência do desempenho de cada algoritmo frente à variação das prioridades operacionais. Adotou-se como métrica de robustez o **ranking médio**: para cada combinação de cenário e perfil, os algoritmos foram ordenados pelo custo médio obtido, atribuindo-se a posição 1 ao de menor custo; o ranking médio de cada algoritmo corresponde à média de suas posições ao longo de todas as combinações. Optou-se pela média das posições de ranking, e não pela média do custo bruto, porque a variação dos pesos entre perfis altera a própria escala numérica do custo, tornando os valores absolutos não diretamente comparáveis entre perfis distintos; a posição relativa, por sua vez, permanece comparável independentemente da escala. O algoritmo considerado mais robusto foi aquele que apresentou o menor ranking médio.

Cabe registrar duas considerações de natureza metodológica relativas à definição dos perfis. A decisão de manter os perfis compartilhados entre todos os algoritmos — em oposição a permitir que cada algoritmo fosse avaliado sob o vetor de pesos que lhe fosse

mais favorável — preserva a comparabilidade que fundamenta a pergunta de pesquisa: a comparação só é legítima quando todos os algoritmos resolvem o mesmo problema. De modo análogo, optou-se por não atribuir pesos distintos a cada cenário em função de sua densidade; embora tal associação pudesse parecer natural, ela incorreria em circularidade, ao embutir no desenho experimental conclusões que só poderiam ser obtidas após a execução. Eventuais relações entre densidade de cenário e sensibilidade aos pesos foram, portanto, tratadas como objeto de análise dos resultados, e não como premissa do delineamento.

3.4.4 Calibração do Raio de Comunicação

O fator k , que define o raio de comunicação como múltiplo do raio de cobertura ($R_{comm} = k \cdot r_s$), foi calibrado de modo a tornar a restrição de conectividade efetiva, isto é, capaz de diferenciar o desempenho dos algoritmos. Avaliaram-se três valores. Com $k = 3$, o raio de comunicação mostrou-se amplo o suficiente para que a conectividade total fosse alcançada em praticamente todas as configurações, tornando o componente incapaz de discriminar entre soluções. Com $k = 2$, o raio de comunicação coincidiu com a distância mínima de sobreposição ($d_{min} = 2r_s$), gerando conflito direto entre os componentes de conectividade e de sobreposição e produzindo configurações com sobreposição excessiva. O valor $k = 2,5$ foi adotado por estabelecer uma faixa de operação intermediária, na qual os drones podem se distanciar o suficiente para evitar sobreposição sem, com isso, perder a conectividade. Dessa forma, permitiu-se que a restrição de rede influenciasse efetivamente a busca sem dominar os demais objetivos.

3.4.5 Procedimento Experimental

Para cada combinação de cenário, perfil de peso e algoritmo, foram realizadas 20 execuções independentes, a fim de assegurar a validade estatística dos resultados. Cada execução foi inicializada de forma independente, sem memória das execuções anteriores. Para garantir a reprodutibilidade e isolar adequadamente os efeitos em análise, a semente do gerador de números aleatórios foi fixada em função do índice da execução, de modo que a n -ésima execução de um dado algoritmo partisse sempre da mesma configuração inicial aleatória nos três perfis de peso. Esse procedimento assegura que as diferenças observadas entre perfis decorram da variação dos pesos, e não da variabilidade estocástica entre execuções.

Em razão do custo computacional do experimento completo — nove cenários, três perfis de peso, seis condições (cinco algoritmos e o *baseline* aleatório) e vinte execuções por condição —, as execuções independentes foram distribuídas entre os múltiplos núcleos de processamento por meio da biblioteca Joblib. A paralelização não altera os resultados, uma vez que cada execução é independente e tem sua semente determinada individualmente; afeta apenas o tempo total de processamento.

Um mecanismo de tratamento de exceções foi incluído por robustez, atribuindo custo de pior caso a qualquer execução que falhasse por erro numérico inesperado, evitando que uma falha isolada interrompesse um experimento de horas. Nas 3.240 execuções realizadas, nenhuma falha dessa natureza foi registrada.

A Figura 1 ilustra a sequência de etapas para a avaliação de uma combinação (cenário, algoritmo) sob um único perfil de peso; o procedimento completo repete essa sequência de forma idêntica para os três perfis de peso definidos na Seção 3.4.3, e para todas as combinações da matriz experimental.

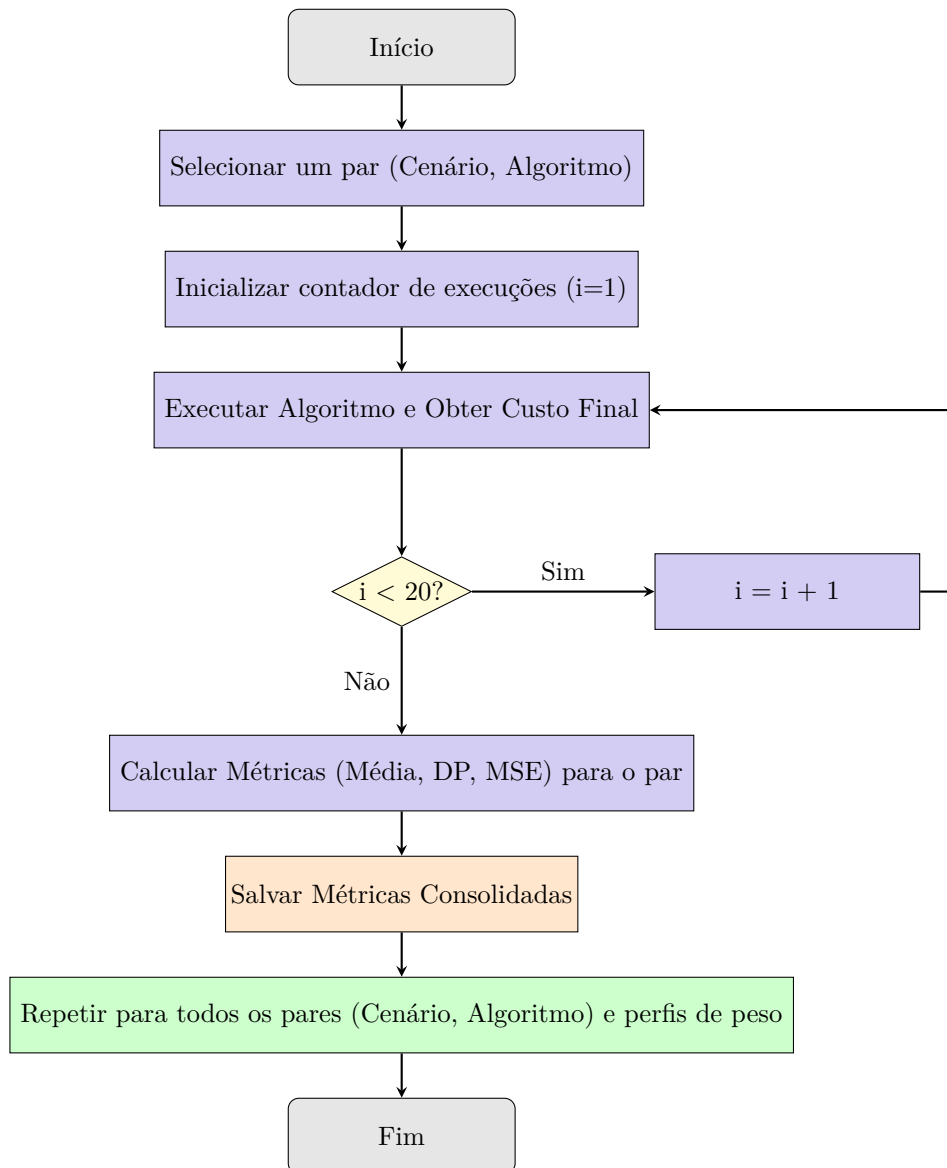


Figura 1 – Fluxograma do procedimento experimental.

3.5 Análise Estatística e Critérios de Comparação

Os dados coletados foram submetidos a análise estatística para assegurar a robustez das conclusões. Para cada combinação de cenário e perfil, registraram-se, a partir das 20 execuções de cada algoritmo, as seguintes métricas: a média do custo final, o desvio padrão do custo e o erro quadrático médio (MSE) em relação à melhor solução encontrada no respectivo cenário. A média indica a qualidade típica das soluções; o desvio padrão, a consistência do algoritmo entre execuções; e o MSE, a dispersão das soluções em torno do melhor resultado obtido. Os valores de MSE calculados para cada condição constam nos dados brutos do repositório público, não sendo tabulados no corpo do texto para não sobrecarregar as tabelas de resultados.

Para verificar se as diferenças de desempenho observadas entre os algoritmos eram estatisticamente significativas ou poderiam ser atribuídas à variabilidade aleatória, aplicou-se o teste de Wilcoxon *signed-rank* a cada par de algoritmos, adotando-se nível de significância $\alpha = 0,05$ (DERRAC et al., 2011). O teste de Wilcoxon foi escolhido por ser um teste não paramétrico, que não pressupõe distribuição normal dos dados — premissa frequentemente não satisfeita pelos resultados de algoritmos meta-heurísticos. O teste foi aplicado dentro de cada perfil de peso separadamente, comparando algoritmos avaliados sob a mesma ponderação; comparações entre perfis distintos não foram realizadas, uma vez que a alteração dos pesos modifica a própria definição do problema otimizado, tornando tal comparação destituída de sentido estatístico.

Como cada combinação de cenário e perfil envolveu seis condições experimentais (cinco algoritmos e o *baseline* aleatório), o número de comparações par-a-par realizadas simultaneamente foi de $\binom{6}{2} = 15$. A realização de múltiplos testes de hipótese em conjunto infla a probabilidade de erro tipo I, isto é, a chance de se detectar uma diferença estatisticamente significativa por acaso, quando ela não existe de fato. Para controlar esse efeito, aplicou-se a correção de Holm-Bonferroni (HOLM, 1979) ao conjunto de p-valores obtido em cada combinação de cenário e perfil, ajustando sequencialmente o limiar de significância exigido conforme a ordenação dos p-valores. A correção de Holm-Bonferroni foi preferida à correção de Bonferroni simples por ser menos conservadora, mantendo controle rigoroso da taxa de erro *familywise* sem penalizar excessivamente o poder estatístico do teste.

A avaliação final de desempenho combinou três níveis de análise: a comparação direta das métricas (média, desvio padrão e MSE) dentro de cada combinação de cenário e perfil; a significância estatística dessas diferenças, aferida pelo teste de Wilcoxon com correção de Holm-Bonferroni; e a robustez de cada algoritmo entre os perfis de prioridade, medida pelo ranking médio definido na Seção 3.4.3. Considerou-se mais performático o algoritmo que aliou baixa média de custo, baixo desvio padrão e, sobretudo, consistência de desempenho ao longo dos diferentes perfis operacionais, indicando uma solução de alta

qualidade, alta confiabilidade e robusta frente a diferentes prioridades de missão.

4 Resultados e Discussão

Este capítulo apresenta e discute os resultados obtidos a partir da execução completa da matriz experimental definida no Capítulo 3: nove cenários, três perfis de peso e seis condições experimentais — os cinco algoritmos de enxame e o *baseline* aleatório —, com vinte execuções independentes por condição, totalizando 3.240 execuções. A análise está organizada em seis níveis complementares: a caracterização quantitativa dos cenários e a visão consolidada dos resultados; o comportamento dos algoritmos em cada regime de densidade; a robustez frente à variação dos perfis de peso; a significância estatística das diferenças observadas; a comparação com o *baseline*; e, por fim, a discussão dos achados de natureza geométrica que emergem dos dados.

4.1 Apresentação Geral dos Resultados

Antes da análise comparativa propriamente dita, é necessário caracterizar quantitativamente o que cada regime de densidade representa. Para tanto, define-se o **teto teórico de cobertura** de um cenário como a razão entre a capacidade nominal de sensoriamento do enxame (a soma das áreas dos N discos de cobertura) e a área da região de interesse, limitada superiormente a 100%:

$$C_{\max} = \min \left(1, \frac{N \cdot \pi \cdot r_s^2}{A} \right) \quad (4.1)$$

onde A é a área total da região. Esse teto é inatingível na prática quando a capacidade nominal se aproxima de 100% — discos não tessalam o plano sem sobreposição —, mas fornece a referência contra a qual a cobertura obtida por cada algoritmo deve ser lida: uma cobertura de 35% pode representar desempenho quase ótimo em um cenário e desempenho pobre em outro. A Tabela 3 consolida, para os nove cenários, o número de drones e a capacidade nominal correspondente.

A Tabela 3 evidencia uma propriedade estrutural do delineamento que condiciona toda a apresentação subsequente: os cenários de 100×100 e 500×500 metros compartilham o mesmo número de drones e a mesma razão $r_s/L = 0,15$ entre raio de cobertura e lado da área. Como todos os componentes da função custo são adimensionais — proporções de área, de pares e de drones — e as sementes de inicialização são fixadas por índice de execução (Seção 3.4.3), o problema de otimização normalizado é idêntico nas duas escalas, e os resultados numéricos coincidem. Os cenários de 250×250 metros, com razão ligeiramente distinta (0,152) e contingentes de 9 e 14 drones nos regimes suficiente e redundante, fornecem a variação real entre escalas, e essa variação mostrou-se pequena,

Tabela 3 – Caracterização dos cenários experimentais: número de drones e capacidade nominal de cobertura

Escala (m)	Densidade	N drones	r_s (m)	Capacidade nominal (%)
100×100	escasso	5	15	35,3
100×100	suficiente	10	15	70,7
100×100	redundante	15	15	106,0
250×250	escasso	5	38	36,3
250×250	suficiente	9	38	65,3
250×250	redundante	14	38	101,6
500×500	escasso	5	75	35,3
500×500	suficiente	10	75	70,7
500×500	redundante	15	75	106,0

Fonte: Elaborado pelo autor (2026).

confirmando que, no modelo adotado, **a densidade de drones é a variável efetiva do experimento, e não a escala física da área**. Essa constatação, discutida em profundidade na Seção 4.6, justifica a escolha da escala 100×100 como representativa nas tabelas e figuras do corpo do texto; os resultados da escala 250×250 , que complementam a análise, constam no Apêndice A.

Em visão consolidada, os resultados estabelecem uma hierarquia estável entre os algoritmos. O PSO obteve o menor custo médio em 25 das 27 combinações de cenário e perfil, com ranking médio de 1,07; o GWO ocupou a segunda posição (ranking médio 2,00), vencendo nas duas combinações restantes; o WOA consolidou-se como terceiro (2,93); SSA (4,22) e CS (4,78) formaram o grupo de menor desempenho entre os enxames; e o *baseline* aleatório ocupou a última posição em todas as 27 combinações (6,00). As seções seguintes decompõem essa visão agregada, mostrando que a distância entre as posições do ranking varia substancialmente conforme o regime de densidade, e que parte das diferenças de posição não é estatisticamente significativa.

4.2 Análise por Regime de Densidade

A Tabela 4 apresenta o custo médio e o desvio padrão das 20 execuções de cada condição, na escala representativa de 100×100 metros, sob o Perfil A (cobertura prioritária); as tabelas análogas para os Perfis B e C e para a escala 250×250 , de papel apenas verificatório, constam no Apêndice A.

A Tabela 5 complementa a anterior traduzindo o custo (métrica agregada e dependente dos pesos) para a taxa de cobertura, métrica operacional diretamente interpretável, lida contra o teto teórico da Equação 4.1. Essa tabela é crucial porque é ela que distingue “cobertura baixa por limitação de recursos” de “cobertura baixa por falha do algoritmo”. Sem essa distinção, o regime escasso seria mal interpretado.

Tabela 4 – Custo médio (desvio padrão) por algoritmo e densidade — escala 100×100 m, Perfil A

Algoritmo	Escasso	Suficiente	Redundante
PSO	0,4555 (0,0002)	0,2266 (0,0085)	0,0881 (0,0128)
GWO	0,4566 (0,0005)	0,2467 (0,0247)	0,1032 (0,0419)
WOA	0,4588 (0,0071)	0,2400 (0,0088)	0,1119 (0,0145)
SSA	0,4601 (0,0042)	0,2813 (0,0131)	0,1786 (0,0160)
CS	0,4643 (0,0045)	0,2909 (0,0124)	0,1748 (0,0126)
Aleatório	0,5560 (0,0333)	0,4153 (0,0393)	0,3042 (0,0472)

Fonte: Elaborado pelo autor (2026).

Tabela 5 – Cobertura média (%) por algoritmo e densidade — escala 100×100 m, Perfil A

Algoritmo	Escasso	Suficiente	Redundante
PSO	34,9	67,8	87,9
GWO	34,8	65,1	86,1
WOA	34,5	65,9	84,6
SSA	34,4	60,8	75,9
CS	34,0	59,4	76,2
Aleatório	26,7	46,2	60,7
<i>Teto teórico</i>	<i>35,3</i>	<i>70,7</i>	<i>100,0</i>

Fonte: Elaborado pelo autor (2026).

4.2.1 Regime Escasso

No regime escasso, a capacidade nominal do enxame (35,3%) é muito inferior à área, de modo que o problema se aproxima de sua forma mais simples: espalhar cinco discos sem sobreposição. Todos os algoritmos de enxame chegaram muito próximos do teto. O PSO atingiu 34,9% de cobertura, o equivalente a 98,9% do máximo teórico, com sobreposição média nula e conectividade integral. O desvio padrão do PSO nesse regime (0,0002) merece registro: as vinte execuções convergiram para soluções de custo praticamente idêntico, indicando que o algoritmo localiza consistentemente a mesma região ótima independentemente da inicialização.

As diferenças absolutas entre os enxames são pequenas (da ordem de 1 ponto percentual de cobertura entre o primeiro e o último colocados, PSO e CS), mas não são ruído: como se verá na Seção 4.4, a superioridade do PSO sobre todos os demais é estatisticamente significativa em todos os cenários escassos. Já entre GWO e WOA, nenhuma diferença sobreviveu à correção estatística nesse regime: os dois são indistinguíveis quando o problema é apenas espalhar recursos insuficientes.

4.2.2 Regime Suficiente

O regime suficiente é o que mais separa os algoritmos, e por razão previsível: com capacidade nominal próxima de 65–70%, os três componentes da função custo exercem pressão simultânea, e a qualidade da coordenação espacial passa a importar mais do que no regime escasso. O intervalo de cobertura entre o melhor e o pior enxame abre-se para 8,4 pontos percentuais (67,8% do PSO contra 59,4% do CS), e a estratificação em três grupos — PSO isolado; GWO e WOA próximos; SSA e CS atrás — torna-se nítida. O PSO alcançou 95,9% do teto teórico na escala 100×100 e 97,7% na escala 250×250 , sustentando a leitura de que seu desempenho relativo independe do cenário.

4.2.3 Regime Redundante

No regime redundante, a capacidade nominal excede a área (106,0%), e a sobreposição deixa de ser evitável: discos circulares não cobrem o plano sem se sobrepor, de modo que qualquer solução que se aproxime da cobertura total necessariamente incorre em *overlap*. O dado empírico confirma a previsão geométrica: mesmo o PSO, melhor colocado, registrou em média 21,8 pares de drones em sobreposição na escala 100×100 , contra zero no regime escasso. A distância entre os grupos amplia-se em termos de cobertura (PSO e GWO acima de 86%, WOA em 84,6%, SSA e CS em torno de 76%), indicando que organizar o excesso de recursos é a tarefa que mais discrimina a qualidade da busca.

Há, porém, uma inversão importante na leitura estatística: é justamente nesse regime que PSO e GWO deixam de ser distinguíveis: nenhuma das nove combinações redundantes apresentou diferença significativa entre os dois após a correção de Holm-Bonferroni. As duas únicas vitórias do GWO no experimento ocorreram exatamente aqui (cenários redundantes de 100×100 e 500×500 sob o Perfil C), por margens de milésimos (0,0432 contra 0,0438 de custo médio), dentro do que a própria análise estatística classifica como empate. A conclusão adequada não é “o GWO vence em rede prioritária redundante”, e sim “no regime redundante, PSO e GWO convergem para desempenho estatisticamente equivalente”.

4.2.4 Figuras Seleccionadas

Para manter a objetividade do texto, o capítulo apresenta apenas quatro figuras, focadas em demonstrar resultados que as tabelas não conseguem expressar. A Figura 2 apresenta as curvas de convergência no cenário 100×100 suficiente sob o Perfil A. É o regime e o perfil em que as diferenças entre algoritmos são maiores e estatisticamente robustas; a figura é necessária porque velocidade e estabilidade de convergência constam dos objetivos específicos deste trabalho e não são observáveis nas métricas finais. A Figura 3 exibe o mapa de posicionamento final no regime escasso, evidenciando visualmente a

disposição espaçada e sem sobreposição que explica a proximidade do teto teórico. A Figura 4 exibe o posicionamento no regime redundante, no qual a sobreposição inevitável — prevista geometricamente e confirmada nos dados — torna-se diretamente visível. Por fim, a Figura 5 repete o cenário redundante sob o Perfil C (rede prioritária), permitindo comparar o efeito do perfil de peso sobre a geometria da solução. É a única evidência visual do papel dos perfis, e corresponde à única condição em que a liderança do PSO foi alcançada pelo GWO.

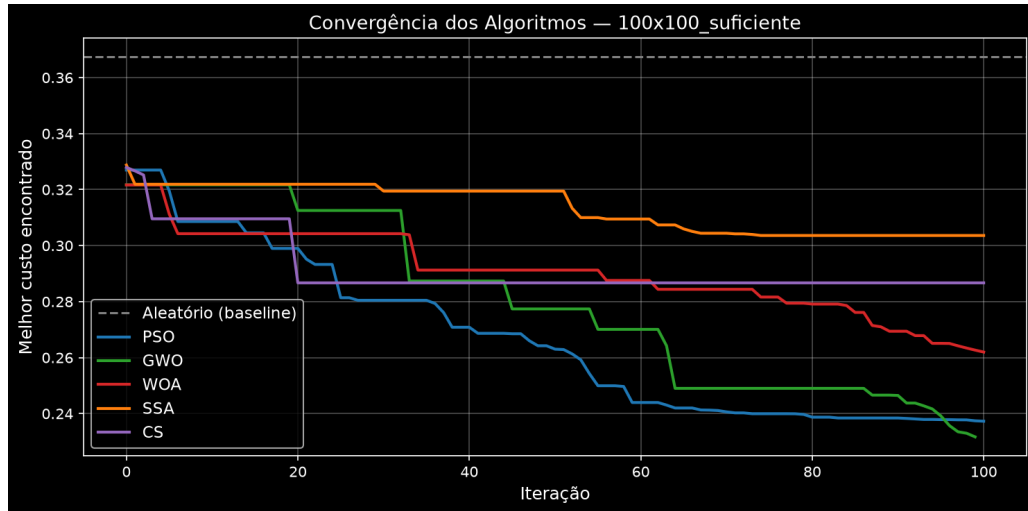


Figura 2 – Curvas de convergência do custo médio — cenário 100×100 suficiente, Perfil A.

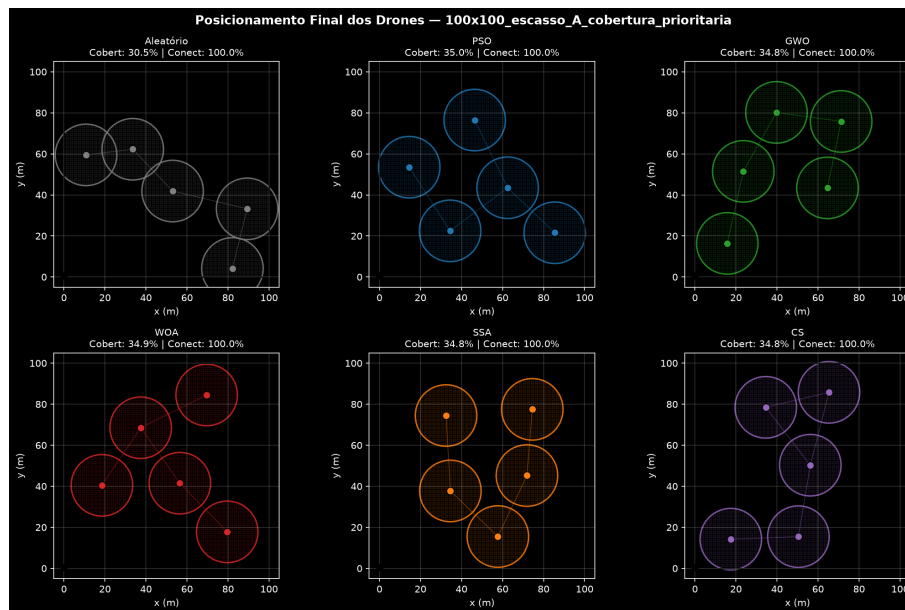


Figura 3 – Posicionamento final dos drones — cenário 100×100 escasso, Perfil A.

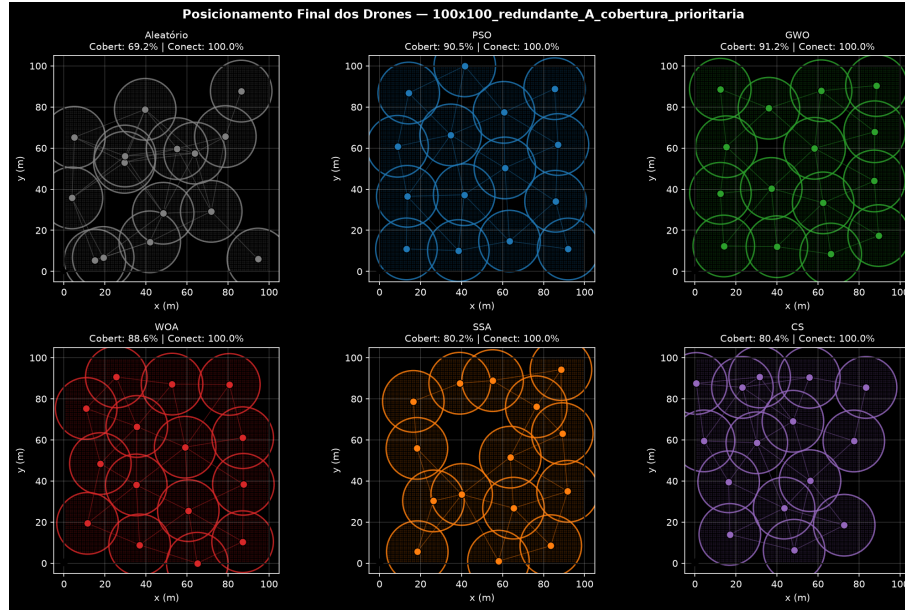


Figura 4 – Posicionamento final dos drones — cenário 100×100 redundante, Perfil A.

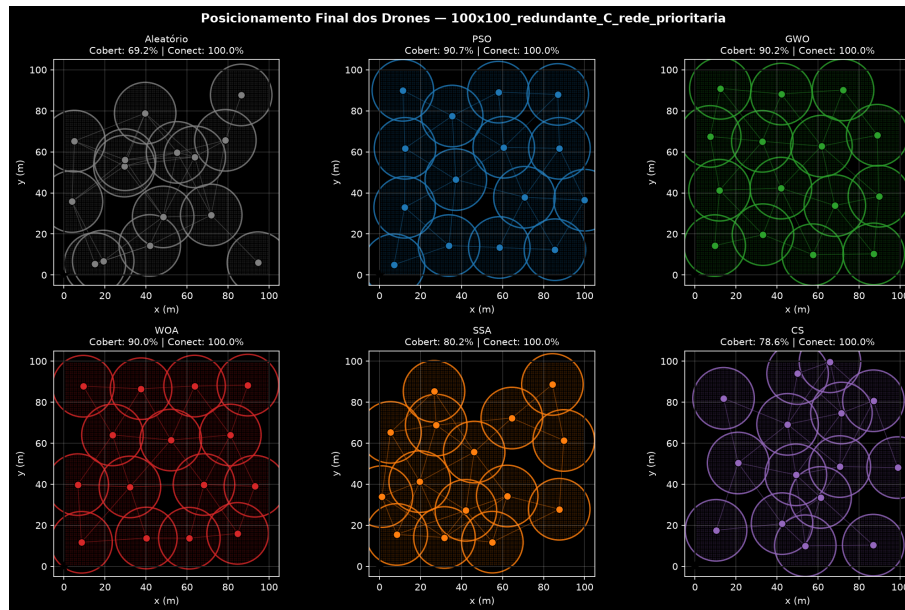


Figura 5 – Posicionamento final dos drones — cenário 100×100 redundante, Perfil C (rede prioritária).

4.3 Robustez entre Perfis de Peso

A Tabela 6 apresenta o ranking médio de cada algoritmo dentro de cada perfil de peso, o ranking médio global e o número de vitórias isoladas. A leitura por colunas revela o achado central: a ordenação dos algoritmos é essencialmente a mesma sob as três prioridades operacionais.

O PSO manteve ranking médio entre 1,00 e 1,22 nos três perfis: foi o primeiro colocado em todas as combinações dos Perfis A e B, e cedeu apenas as duas combinações

Tabela 6 – Ranking médio por perfil de peso, ranking médio global e vitórias isoladas (27 combinações)

Algoritmo	Perfil A	Perfil B	Perfil C	Global	Vitórias
PSO	1,00	1,00	1,22	1,07	25
GWO	2,22	2,00	1,78	2,00	2
WOA	2,78	3,00	3,00	2,93	0
SSA	4,22	4,22	4,22	4,22	0
CS	4,78	4,78	4,78	4,78	0
Aleatório	6,00	6,00	6,00	6,00	0

Fonte: Elaborado pelo autor (2026).

já discutidas no Perfil C. O deslocamento simétrico do GWO (de 2,22 no Perfil A para 1,78 no Perfil C) sugere leve afinidade do algoritmo com o componente de conectividade, mas a magnitude do movimento é pequena e, como visto, as trocas de posição ocorreram dentro de empates estatísticos. SSA e CS mantiveram-se estáveis nas posições inferiores, e o *baseline* não saiu da última posição em nenhuma das 27 combinações.

Cabe aqui a ressalva metodológica que evita a sobreinterpretação do ranking: **ranking de robustez e significância par-a-par contam histórias complementares, não redundantes**. O ranking registra quem ficou à frente, ainda que por margem ínfima; o teste estatístico registra se a margem é distinguível do acaso. O ranking médio de 2,00 do GWO contra 2,93 do WOA, por exemplo, decorre de o GWO vencer o confronto direto “por pouco” com frequência. Mas, em 15 das 27 combinações, essa diferença não é estatisticamente significativa. A posição no ranking responde “quem escolher, na dúvida”; a significância responde “quanta confiança depositar nessa escolha”. As duas leituras são apresentadas em conjunto justamente para que uma discipline a outra.

4.4 Significância Estatística

Foram realizados 405 testes de Wilcoxon *signed-rank* — 15 pares de condições em cada uma das 27 combinações de cenário e perfil —, com correção de Holm-Bonferroni aplicada por combinação, conforme a Seção 3.4.3. O pareamento exigido pelo teste é garantido pelo delineamento: as execuções de mesmo índice compartilham a mesma semente e, portanto, a mesma configuração inicial. A Tabela 7 sintetiza os 405 resultados, reportando, para cada par, o número de combinações com diferença significativa após a correção, segmentado por densidade — os p-valores individuais, inviáveis de listar no corpo do texto, constam por combinação no Apêndice A para os três confrontos mais disputados.

Três padrões estruturam a leitura da tabela. Primeiro, todas as comparações que envolvem o *baseline* aleatório, e todas as comparações do PSO contra WOA, SSA e CS, foram significativas nas 27 combinações. A superioridade dos enxames sobre o acaso, e a do

Tabela 7 – Síntese dos testes de Wilcoxon com correção de Holm-Bonferroni: número de combinações com diferença significativa ($\alpha = 0,05$), por densidade (máximo de 9 por densidade; 27 no total)

Par	Escasso	Suficiente	Redundante	Total
PSO $\times$ GWO	9	5	0	14
PSO $\times$ WOA	9	9	9	27
PSO $\times$ SSA	9	9	9	27
PSO $\times$ CS	9	9	9	27
GWO $\times$ WOA	0	5	7	12
GWO $\times$ SSA	5	9	9	23
GWO $\times$ CS	9	9	9	27
WOA $\times$ SSA	0	9	9	18
WOA $\times$ CS	3	9	9	21
SSA $\times$ CS	6	0	0	6
PSO $\times$ Aleatório	9	9	9	27
GWO $\times$ Aleatório	9	9	9	27
WOA $\times$ Aleatório	9	9	9	27
SSA $\times$ Aleatório	9	9	9	27
CS $\times$ Aleatório	9	9	9	27

Fonte: Elaborado pelo autor (2026).

PSO sobre três dos quatro concorrentes, não admite ambiguidade. Segundo, os confrontos internos aos grupos dependem do regime: PSO e GWO diferem significativamente em todos os cenários escassos, em parte dos suficientes e em nenhum redundante; GWO e WOA exibem o padrão inverso, indistinguíveis no regime escasso e majoritariamente distintos no redundante. Terceiro, SSA e CS confirmam-se como grupo homogêneo de baixo desempenho: distinguem-se apenas em 6 das 27 combinações, todas no regime escasso.

O padrão conjunto tem interpretação coerente: no regime escasso, o problema é simples o bastante para que quase todos os algoritmos se aproximem do teto, comprimindo as diferenças entre os intermediários, mas não a ponto de apagar a vantagem do PSO. No regime redundante, a dificuldade de organizar o excesso separa os grupos (enxames fortes contra fracos), ao mesmo tempo em que os dois melhores algoritmos atingem soluções de qualidade equivalente.

4.5 Comparação com o *Baseline*

O *baseline* aleatório corresponde a um único posicionamento sorteado uniformemente na área, sem qualquer processo de busca: representa, operacionalmente, a implantação de drones sem planejamento, e não um método concorrente de otimização. A Tabela 8 quantifica, em pontos percentuais e por regime de densidade sob o Perfil A, o ganho de cobertura dos algoritmos de enxame sobre essa referência — a resposta direta à pergunta que motiva o trabalho: otimizar compensa o esforço computacional?

Tabela 8 – Ganho médio de cobertura sobre o *baseline* aleatório (pontos percentuais), por densidade — Perfil A, média das três escalas

Algoritmo	Escasso	Suficiente	Redundante
PSO	8,3	21,3	27,5
GWO	8,2	19,1	26,2
WOA	7,9	19,1	24,2
SSA	7,9	14,5	15,8
CS	7,4	13,4	15,5

Fonte: Elaborado pelo autor (2026).

O ganho cresce com a densidade: de 7,4 a 8,3 pontos percentuais no regime escasso, para 13,4 a 21,3 no suficiente e 15,5 a 27,5 no redundante. O padrão é explicável pela própria geometria do problema: com poucos drones em área grande, mesmo o posicionamento aleatório raramente produz sobreposições, e a margem para otimizar é menor; com muitos drones, o posicionamento aleatório desperdiça capacidade em sobreposições e desconexões, e a coordenação passa a valer mais. A conectividade reforça a leitura: os algoritmos de enxame mantiveram 100% dos drones conectados em praticamente todas as condições. As únicas exceções registradas foram do SSA e do CS, com 99,4% no cenário 250×250 suficiente, enquanto o *baseline* oscilou entre 71,0% (escasso) e 84,8% (redundante) de drones no maior componente conexo. Esse contraste confirma que a calibração $k = 2,5$ (Seção 3.4.4) produziu uma restrição efetiva: exigente o bastante para penalizar o posicionamento sem planejamento, folgada o bastante para não sufocar a busca dos algoritmos.

4.6 Discussão dos Achados Geométricos

Dois achados de natureza geométrica emergem dos dados e merecem discussão própria, por condicionarem a interpretação de todo o restante.

4.6.1 Invariância de Escala do Modelo Normalizado

Conforme antecipado na Seção 4.1, os cenários de 100×100 e 500×500 metros produziram resultados numericamente idênticos, e os de 250×250 metros, resultados muito próximos. O fenômeno não é coincidência empírica, e sim consequência dedutível do delineamento: ao escalar o raio de cobertura proporcionalmente ao lado da área (Seção 3.4.1), a razão r_s/L manteve-se constante em 0,15 nas escalas de 100 e 500 metros; como o número de drones, a resolução relativa da grade e as sementes de inicialização também coincidem, e como os três componentes da função custo são adimensionais, o problema de otimização é rigorosamente o mesmo nas duas escalas, a menos de um fator multiplicativo nas coordenadas, que os algoritmos, operando sobre limites igualmente escalados, percorrem de forma equivalente. A escala 250×250 quebra parcialmente a identidade (razão 0,152;

9 e 14 drones nos regimes suficiente e redundante) e produziu variações de cobertura inferiores a 4 pontos percentuais, que praticamente desaparecem quando as coberturas são normalizadas pelos respectivos tetos teóricos (95,9% contra 97,7% do teto, no regime suficiente do PSO).

Disso decorrem duas consequências. Analiticamente, **a variável efetiva do experimento é a densidade de drones, não a escala física** — o que fundamenta a apresentação dos resultados por regime de densidade e a seleção de uma única escala como representativa. Há também um limite do que os resultados permitem afirmar: a invariância verificada é uma propriedade do modelo normalizado, funcionando como teste de consistência interna da implementação, e não uma afirmação sobre operações reais. Em campo, fatores ausentes do modelo, como alcance absoluto de rádio, precisão de posicionamento e condições de vento, dependem da escala física e quebrariam a equivalência — limitação retomada na Seção 5.2.

4.6.2 Sobreposição e Cobertura por Regime de Densidade

O segundo achado diz respeito à relação entre os componentes de sobreposição e cobertura, que muda de natureza conforme o regime. Quando a capacidade nominal do enxame é inferior à área (regimes escasso e suficiente), minimizar sobreposição e maximizar cobertura são objetivos alinhados: cada par de discos sobrepostos desperdiça capacidade que poderia cobrir área nova. Quando a capacidade excede a área (regime redundante), a relação inverte-se parcialmente: alguma sobreposição torna-se condição necessária para aproximar-se da cobertura total, uma vez que discos circulares não tessalam o plano. A Tabela 9 apresenta o número médio de pares de drones em sobreposição por regime. Os valores elevados do próprio PSO no regime redundante não indicam deficiência: como discos circulares não tessalam o plano, alguma sobreposição é geometricamente inevitável quando a capacidade excede a área.

Tabela 9 – Número médio de pares de drones em sobreposição, por densidade — Perfil A, média das três escalas

Algoritmo	Escasso	Suficiente	Redundante
PSO	0,0	5,3	21,5
GWO	0,1	4,7	21,3
WOA	0,1	5,4	18,4
SSA	1,2	8,2	21,2
CS	1,8	7,0	17,7
Aleatório	2,3	9,4	22,1

Fonte: Elaborado pelo autor (2026).

Os dados confirmam a previsão nos dois sentidos. No regime escasso, os melhores algoritmos zeraram (PSO) ou quase zeraram (GWO e WOA, 0,1) as sobreposições, enquanto

SSA, CS e o *baseline* mantiveram sobreposições residuais — ali, *overlap* é de fato evitável, e evitá-lo distingue qualidade de busca. No regime redundante, todos os algoritmos, incluindo o PSO, operaram com 17 a 22 pares sobrepostos em média: o *overlap* deixou de discriminar os melhores dos piores pela sua ausência, e passou a discriminá-los pela forma como é administrado: PSO e GWO aceitam sobreposição comparável à do *baseline*, mas a convertem em 26 a 28 pontos percentuais a mais de cobertura. O comportamento observado é coerente com o *trade-off* estrutural do problema multi-objetivo discutido na fundamentação (Seção 2.2): os fatores competem ou cooperam conforme a região do espaço de soluções em que o exame opera, e o regime de densidade é o que determina essa região.

5 Conclusão

Este trabalho realizou um estudo comparativo de cinco meta-heurísticas de inteligência de enxame — PSO, GWO, WOA, SSA e Cuckoo Search — aplicadas ao problema de posicionamento estático de múltiplos VANTs para cobertura de área, sob uma função custo multi-objetivo que agrega cobertura, sobreposição e conectividade de rede. A comparação abrangeu nove cenários, três perfis de prioridade operacional e vinte execuções independentes por condição, totalizando 3.240 execuções, com validação estatística por teste de Wilcoxon e correção de Holm-Bonferroni.

5.1 Síntese dos Resultados e Resposta aos Objetivos

Os resultados sustentam uma conclusão central: o PSO foi o algoritmo mais eficiente e mais robusto para o problema estudado. Obteve o menor custo médio em 25 das 27 combinações de cenário e perfil, ranking médio global de 1,07, superioridade estatisticamente significativa sobre WOA, SSA, CS e o *baseline* em todas as combinações, e desvios padrão consistentemente baixos. No regime escasso, foram da ordem de 10^{-4} , indicando convergência quase determinística para a mesma região ótima. O GWO consolidou-se como segunda opção (ranking 2,00), estatisticamente equivalente ao PSO no regime redundante. As duas únicas combinações em que o superou ocorreram nesse regime, por margens dentro do empate estatístico. O WOA ocupou posição intermediária, e SSA e CS formaram um grupo homogêneo de menor desempenho, distinguível entre si apenas no regime escasso. O *baseline* aleatório foi inferior a todos os enxames em todas as condições, com ganhos de cobertura da otimização que cresceram de cerca de 8 pontos percentuais no regime escasso para até 27 no redundante.

Quanto aos objetivos específicos definidos na Introdução, todos foram cumpridos. A revisão bibliográfica sobre VANTs, o problema de cobertura e os cinco algoritmos foi consolidada no Capítulo 2. A modelagem matemática do problema resultou na função custo normalizada de três componentes — cobertura, sobreposição e conectividade via maior componente conexo —, formalizada na Seção 3.3. Os cinco algoritmos e o *baseline* foram implementados em Python, com função objetivo vetorizada e execução paralelizada. A metodologia experimental sistemática materializou-se na matriz de nove cenários, três perfis de peso compartilhados, sementes pareadas por índice de execução e protocolo estatístico com correção para múltiplas comparações. Por fim, as simulações foram executadas e analisadas comparativamente no Capítulo 4, em termos de taxa de cobertura final (lida contra o teto teórico de cada cenário), estabilidade entre execuções, aferida pelo desvio padrão, e velocidade de convergência, examinada qualitativamente pelas curvas de

convergência do cenário representativo.

Quanto ao objetivo geral, a comparação de custo computacional foi tratada por controle, e não por medição: todos os algoritmos operaram sob o mesmo orçamento de busca (30 agentes e 100 iterações, correspondendo ao mesmo número de avaliações da função objetivo), de modo que as diferenças de qualidade de solução refletem eficiência de busca em igualdade de custo. Sob esse critério, a resposta à pergunta de pesquisa é direta: para o posicionamento estático de VANTs com os três objetivos considerados, o PSO é a abordagem mais eficiente entre as avaliadas, e essa conclusão é estável frente à variação das prioridades operacionais de missão.

Além da hierarquia entre algoritmos, o trabalho produziu dois achados de caráter estrutural. O primeiro é a invariância de escala do modelo normalizado: mantida a razão entre raio de cobertura e dimensão da área, a escala física não altera o problema de otimização: a variável efetiva é a densidade de drones, e é ela que define os regimes analíticos. O segundo é a mudança de natureza da relação entre sobreposição e cobertura conforme o regime: objetivos alinhados quando a capacidade do enxame é inferior à área, parcialmente conflitantes quando a excede — no regime redundante, sobrepor é condição para cobrir, e a qualidade do algoritmo se manifesta em como administra esse excesso.

5.2 Limitações

Os resultados devem ser lidos dentro das delimitações do modelo adotado, e algumas delas merecem registro explícito.

A primeira é o modelo binário de conectividade: um par de drones está conectado ou não está, sem gradação por força de sinal. A escolha foi deliberada (preserva a independência entre os componentes de cobertura e conectividade, como discutido na Seção 3.3), mas afasta o modelo da física real da comunicação sem fio, na qual a qualidade do enlace degrada continuamente com a distância.

A segunda é o caráter estritamente estático e bidimensional do problema: as posições são avaliadas apenas em sua configuração final, a altitude é parâmetro de cenário e não variável de busca, e não há consideração de trajetória, consumo energético de deslocamento ou reposicionamento ao longo da missão.

A terceira decorre do modelo de Disco Binário em ambiente idealizado: a área é livre de obstáculos, relevo e zonas de exclusão, e a cobertura de cada sensor é determinística e uniforme. Essas condições superestimam a regularidade do sensoriamento real.

A quarta diz respeito ao *baseline*: o posicionamento aleatório corresponde a uma única amostra uniforme, sem processo de busca. Ele quantifica o valor de otimizar frente a não planejar (que era o propósito da comparação), mas não equivale a uma busca

aleatória com orçamento de avaliações pareado ao dos enxames, referência mais exigente que permanece não testada.

A quinta é consequência do próprio delineamento: ao escalar o raio de cobertura proporcionalmente à área, a escala física deixou de ser um fator experimental independente — os cenários de mesma densidade tornaram-se equivalentes sob normalização. O desenho ganhou coerência interna, mas abriu mão de investigar efeitos genuínos de escala, que em operações reais existem por fatores externos ao modelo.

Por fim, registram-se duas limitações de escopo estatístico e paramétrico: os algoritmos operaram com os hiperparâmetros recomendados em suas formulações consagradas, sem ajuste fino ao problema (o que é adequado a uma comparação em condições padronizadas, mas deixa em aberto o desempenho de versões calibradas), e o método da soma ponderada, ainda que avaliado sob três perfis de peso, não substitui a caracterização completa da fronteira de Pareto do problema multi-objetivo.

5.3 Trabalhos Futuros

As limitações apontadas delineiam extensões naturais desta pesquisa.

A primeira direção é a adoção de um modelo de conectividade graduado por força de sinal, em que a qualidade do enlace decai continuamente com a distância, extensão já antecipada na metodologia (Seção 3.3), que exigirá tratar deliberadamente o acoplamento entre cobertura e conectividade que o modelo binário evita, além da calibração do parâmetro de decaimento.

A segunda é a extensão do problema estático para o planejamento de trajetórias (*path planning*), reintroduzindo a dimensão temporal e, com ela, o consumo de bateria como componente legítimo da função custo. Recupera-se, assim, em contexto adequado, o objetivo substituído na concepção deste trabalho.

A terceira é a incorporação da altitude como variável de otimização, elevando o problema a três dimensões e permitindo enxames com raios de cobertura heterogêneos: configuração na qual drones em altitudes distintas cobrem áreas de tamanhos distintos, ampliando substancialmente o espaço de soluções.

Quatro extensões adicionais completam a agenda: a caracterização da fronteira de Pareto por algoritmos multi-objetivo dedicados, como o NSGA-II, em substituição à soma ponderada; o ajuste sistemático de hiperparâmetros por procedimentos automatizados, verificando se a hierarquia observada se mantém entre versões calibradas; a introdução de cenários com obstáculos e zonas de exclusão, aproximando o modelo de missões reais; e a validação experimental do posicionamento otimizado em plataformas físicas, etapa final de transferência dos resultados de simulação para a operação.

Referências

- AKBARZADEH, V.; LÉVESQUE, J.-C.; GAGNÉ, C.; PARIZEAU, M. Efficient sensor placement optimization using gradient descent and probabilistic coverage. *Sensors*, MDPI, v. 14, n. 8, p. 15525–15552, 2014. Citado na página 33.
- ALSAMHI, S. H.; MA, O.; ANSARI, M. S.; ALMALKI, F. A. A review of UAV platform, key technologies, and civil applications. *Sensors*, MDPI, v. 22, n. 15, p. 5543, 2022. Citado 3 vezes nas páginas 15, 18 e 20.
- BLUM, C.; ROLI, A. Metaheuristics in combinatorial optimization: Overview and conceptual comparison. *ACM computing surveys (CSUR)*, ACM, v. 35, n. 3, p. 268–308, 2003. Citado 2 vezes nas páginas 15 e 23.
- BONABEAU, E.; DORIGO, M.; THERAULAZ, G. *Swarm intelligence: from natural to artificial systems*. [S.l.]: Oxford University Press, 1999. Citado 2 vezes nas páginas 15 e 23.
- CABREIRA, T. M.; BRISOLARA, L. B.; FERREIRA, P. R. J. Survey on coverage path planning with unmanned aerial vehicles. *Drones*, MDPI, v. 3, n. 1, p. 4, 2019. Citado 2 vezes nas páginas 19 e 20.
- CAVALCANTI, R. C.; AMORIM, J. S. d.; SANTOS, J. B. d.; SILVA, I. A. d.; SILVA, C. M. d.; SILVA, A. V. d. Drones no georreferenciamento rural: tendências atuais e perspectivas futuras. *Revista de Gestão e Secretariado*, v. 16, n. 3, p. 779–788, 2025. Citado na página 20.
- CHIARADONNA, S.; BONDAVALLI, A.; GIANDOMENICO, F. On enhancing efficiency and accuracy of particle swarm optimization. *Computational Intelligence and Neuroscience*, Hindawi, v. 2020, p. 1–12, 2020. Citado na página 25.
- DEB, K. Multi-objective optimization. In: *Search methodologies*. [S.l.]: Springer, 2014. p. 403–449. Citado na página 34.
- DERRAC, J.; GARCÍA, S.; MOLINA, D.; HERRERA, F. A practical tutorial on the use of nonparametric statistical tests as a methodology for comparing evolutionary and swarm intelligence algorithms. *Swarm and Evolutionary Computation*, Elsevier, v. 1, n. 1, p. 3–18, 2011. Citado na página 42.
- DOKEROGLU, T.; DENIZ, A.; KIZILOZ, H. E. A comprehensive survey on recent metaheuristics for data clustering. *Multimedia Tools and Applications*, Springer, v. 78, p. 21961–22002, 2019. Citado na página 16.
- ELIJAH, O.; RAHMAN, T. A.; OYINLOLA, I.; OYELERE, F.; ADESINA, A. An overview of Internet of Things (IoT) and data analytics in agriculture: Benefits and challenges. *IEEE Internet of Things Journal*, IEEE, v. 5, n. 5, p. 3758–3773, 2018. Citado na página 15.
- FRIHA, O.; FERRAG, M. A.; SHU, L.; MAGLARAS, L.; WANG, X. Internet of Things for the future of smart agriculture: A comprehensive survey of emerging technologies.

- IEEE/CAA Journal of Automatica Sinica*, IEEE, v. 8, n. 4, p. 718–752, 2021. Citado na página 15.
- GAREY, M. R.; JOHNSON, D. S. *Computers and intractability: A guide to the theory of NP-completeness*. [S.l.]: W. H. Freeman & Co., 1979. Citado 2 vezes nas páginas 22 e 38.
- HOLM, S. A simple sequentially rejective multiple test procedure. *Scandinavian Journal of Statistics*, JSTOR, v. 6, n. 2, p. 65–70, 1979. Citado na página 42.
- HUSSAIN, K.; SALLEH, M. N. M.; CHENG, S.; SHI, Y. On the exploration and exploitation in popular swarm-based metaheuristic algorithms. *Neural Computing and Applications*, Springer, v. 31, p. 7665–7683, 2019. Citado na página 23.
- KENNEDY, J.; EBERHART, R. Particle Swarm Optimization. In: IEEE. *Proceedings of ICNN'95-international conference on neural networks*. [S.l.], 1995. v. 4, p. 1942–1948. Citado 2 vezes nas páginas 15 e 24.
- LUHMANN, T.; ROBSON, S.; KYLE, S.; BOEHM, J. *Close-range photogrammetry and 3D imaging*. 2nd. ed. [S.l.]: De Gruyter, 2013. Citado 2 vezes nas páginas 19 e 37.
- MARLER, R. T.; ARORA, J. S. The weighted sum method for multi-objective optimization: new insights. *Structural and Multidisciplinary Optimization*, Springer, v. 41, n. 6, p. 853–862, 2010. Citado na página 38.
- MIRJALILI, S.; GANDOMI, A. H.; MIRJALILI, S. Z.; SAREMI, S.; FARIS, H.; MIRJALILI, S. M. Salp Swarm Algorithm: A bio-inspired optimizer for engineering design problems. *Advances in Engineering Software*, Elsevier, v. 114, p. 163–191, 2017. Citado 2 vezes nas páginas 16 e 29.
- MIRJALILI, S.; LEWIS, A. The Whale Optimization Algorithm. *Advances in Engineering Software*, Elsevier, v. 95, p. 51–67, 2016. Citado 2 vezes nas páginas 16 e 28.
- MIRJALILI, S.; MIRJALILI, S. M.; LEWIS, A. Grey Wolf Optimizer. *Advances in Engineering Software*, Elsevier, v. 69, p. 46–61, 2014. Citado 2 vezes nas páginas 16 e 27.
- NWAOGU, J. M.; YANG, Y.; CHAN, A. P.; CHI, H.-l. Application of drones in the architecture, engineering, and construction (AEC) industry. *Automation in Construction*, v. 150, p. 104827, 2023. Citado na página 15.
- SANKARAN, S.; KHOT, L. R.; ESPINOZA, C.-Z. et al. Low-altitude remote sensing for phenotyping in agriculture and plant breeding: A review. *Computers and Electronics in Agriculture*, Elsevier, v. 115, p. 44–53, 2015. Citado 2 vezes nas páginas 15 e 20.
- SHI, Y.; EBERHART, R. A modified particle swarm optimizer. In: IEEE. *1998 IEEE International Conference on Evolutionary Computation Proceedings. IEEE World Congress on Computational Intelligence*. [S.l.], 1998. p. 69–73. Citado na página 25.
- TOSSA, F.; ABDOU, W.; ANSARI, K.; EZIN, E. C.; GOUTON, P. Area coverage maximization under connectivity constraint in wireless sensor networks. *Sensors*, MDPI, v. 22, n. 5, p. 1712, 2022. Citado 3 vezes nas páginas 20, 21 e 34.
- TSOUROS, D. C.; BIBI, S.; SARIGIANNIDIS, P. G. A review on UAV-based applications for precision agriculture. *Information*, MDPI, v. 10, n. 11, p. 349, 2019. Citado 3 vezes nas páginas 15, 18 e 19.

TUBA, E.; ALIHODZIC, A.; TUBA, M. Salp Swarm Algorithm for constrained optimization problems. In: IEEE. *2018 26th Telecommunications Forum (TELFOR)*. [S.l.], 2018. p. 1–4. Citado na página 16.

WOLPERT, D. H.; MACREADY, W. G. No free lunch theorems for optimization. *IEEE Transactions on Evolutionary Computation*, IEEE, v. 1, n. 1, p. 67–82, 1997. Citado 2 vezes nas páginas 16 e 23.

YANG, X.-S.; DEB, S. Cuckoo search via Lévy flights. In: IEEE. *2009 World Congress on Nature & Biologically Inspired Computing (NaBIC)*. [S.l.], 2009. p. 210–214. Citado 2 vezes nas páginas 16 e 31.

ZHANG, C.; KOVACS, J. M. The application of small unmanned aerial systems for precision agriculture: a review. *Precision Agriculture*, Springer, v. 13, p. 693–712, 2012. Citado 2 vezes nas páginas 15 e 18.

Apêndices

APÊNDICE A – Tabelas Completas de Resultados

Este apêndice reúne as tabelas de verificação que complementam as tabelas de argumento apresentadas no Capítulo 4: o custo médio e o desvio padrão de todas as condições experimentais, nos três perfis de peso e nas escalas de 100×100 e 250×250 metros, e os p-valores individuais dos três confrontos estatisticamente mais disputados. Os cenários de 500×500 metros são numericamente idênticos aos de 100×100 , pela invariância de escala discutida na Seção 4.6, e por isso não são repetidos. Os dados brutos completos — incluindo os 405 testes estatísticos — acompanham o código-fonte no repositório do projeto.

A.1 Custo Médio por Perfil de Peso

A Tabela 10 apresenta os resultados sob o Perfil A (cobertura prioritária).

Tabela 10 – Custo médio (desvio padrão) por algoritmo, cenário e densidade — escalas 100×100 e 250×250 m, Perfil A (cobertura prioritária)

Algoritmo	100 × 100 m			250 × 250 m		
	Escasso	Suficiente	Redundante	Escasso	Suficiente	Redundante
PSO	0,4555 (0,0002)	0,2266 (0,0085)	0,0881 (0,0128)	0,4487 (0,0002)	0,2540 (0,0068)	0,0962 (0,0119)
GWO	0,4566 (0,0005)	0,2467 (0,0247)	0,1032 (0,0419)	0,4499 (0,0020)	0,2624 (0,0166)	0,0982 (0,0076)
WOA	0,4588 (0,0071)	0,2400 (0,0088)	0,1119 (0,0145)	0,4516 (0,0073)	0,2731 (0,0119)	0,1193 (0,0094)
SSA	0,4601 (0,0042)	0,2813 (0,0131)	0,1786 (0,0160)	0,4531 (0,0046)	0,3039 (0,0137)	0,1777 (0,0179)
CS	0,4643 (0,0045)	0,2909 (0,0124)	0,1748 (0,0126)	0,4584 (0,0046)	0,3068 (0,0075)	0,1870 (0,0100)
Aleatório	0,5560 (0,0333)	0,4153 (0,0393)	0,3042 (0,0472)	0,5520 (0,0339)	0,4383 (0,0373)	0,3208 (0,0451)

Fonte: Elaborado pelo autor (2026).

A Tabela 11 apresenta os resultados equivalentes sob o Perfil B (transição).

Tabela 11 – Custo médio (desvio padrão) por algoritmo, cenário e densidade — escalas 100×100 e 250×250 m, Perfil B (transição)

Algoritmo	100 × 100 m			250 × 250 m		
	Escasso	Suficiente	Redundante	Escasso	Suficiente	Redundante
PSO	0,3254 (0,0003)	0,1603 (0,0042)	0,0643 (0,0074)	0,3206 (0,0001)	0,1808 (0,0041)	0,0711 (0,0082)
GWO	0,3273 (0,0037)	0,1692 (0,0128)	0,0679 (0,0173)	0,3212 (0,0009)	0,1931 (0,0164)	0,0771 (0,0215)
WOA	0,3285 (0,0060)	0,1772 (0,0094)	0,0815 (0,0102)	0,3235 (0,0062)	0,1975 (0,0090)	0,0870 (0,0076)
SSA	0,3299 (0,0039)	0,2033 (0,0116)	0,1299 (0,0100)	0,3251 (0,0041)	0,2171 (0,0080)	0,1302 (0,0118)
CS	0,3330 (0,0036)	0,2099 (0,0064)	0,1255 (0,0083)	0,3283 (0,0043)	0,2217 (0,0066)	0,1351 (0,0090)
Aleatório	0,4674 (0,0665)	0,3557 (0,0623)	0,2543 (0,0599)	0,4646 (0,0664)	0,3734 (0,0645)	0,2727 (0,0659)

Fonte: Elaborado pelo autor (2026).

A Tabela 12 completa a série sob o Perfil C (rede prioritária).

Tabela 12 – Custo médio (desvio padrão) por algoritmo, cenário e densidade — escalas 100×100 e 250×250 m, Perfil C (rede prioritária)

Algoritmo	100 × 100 m			250 × 250 m		
	Escasso	Suficiente	Redundante	Escasso	Suficiente	Redundante
PSO	0,1952 (0,0002)	0,0981 (0,0038)	0,0438 (0,0050)	0,1923 (0,0001)	0,1096 (0,0033)	0,0466 (0,0061)
GWO	0,1962 (0,0019)	0,1014 (0,0079)	0,0432 (0,0110)	0,1927 (0,0006)	0,1139 (0,0087)	0,0483 (0,0111)
WOA	0,1976 (0,0040)	0,1082 (0,0047)	0,0507 (0,0059)	0,1941 (0,0037)	0,1192 (0,0076)	0,0552 (0,0055)
SSA	0,1980 (0,0028)	0,1261 (0,0082)	0,0811 (0,0075)	0,1949 (0,0028)	0,1318 (0,0063)	0,0810 (0,0068)
CS	0,2000 (0,0028)	0,1272 (0,0053)	0,0794 (0,0057)	0,1974 (0,0030)	0,1345 (0,0057)	0,0849 (0,0064)
Aleatório	0,3788 (0,1061)	0,2962 (0,0964)	0,2044 (0,0817)	0,3773 (0,1060)	0,3086 (0,0991)	0,2247 (0,0972)

Fonte: Elaborado pelo autor (2026).

A.2 P-valores dos Confrontos Disputados

As tabelas a seguir reportam os p-valores brutos do teste de Wilcoxon *signed-rank* para os três pares de algoritmos cujas diferenças nem sempre sobreviveram à correção de Holm-Bonferroni — PSO × GWO, GWO × WOA e SSA × CS. Os demais doze pares foram significativos em todas (ou quase todas) as combinações, conforme a Tabela 7, e seus p-valores constam nos dados brutos do repositório.

Tabela 13 – P-valores do teste de Wilcoxon para os confrontos PSO × GWO, GWO × WOA e SSA × CS — Perfil A (cobertura prioritária) (* indica diferença significativa após correção de Holm-Bonferroni, $\alpha = 0,05$)

Perfil	Cenário	PSO × GWO	GWO × WOA	SSA × CS
A	100×100_escasso	<0,0001*	0,9306	0,0073*
A	100×100_suficiente	0,0001*	0,7841	0,0400
A	100×100_redundante	0,5706	0,0973	0,4980
A	250×250_escasso	<0,0001*	0,2856	0,0049*
A	250×250_suficiente	0,0296	0,0400	0,2943
A	250×250_redundante	0,3683	<0,0001*	0,0583
A	500×500_escasso	<0,0001*	0,9306	0,0073*
A	500×500_suficiente	0,0001*	0,7841	0,0400
A	500×500_redundante	0,5706	0,0973	0,4980

Fonte: Elaborado pelo autor (2026).

A Tabela 14 repete o levantamento sob o Perfil B.

Tabela 14 – P-valores do teste de Wilcoxon para os confrontos PSO $\times$ GWO, GWO $\times$ WOA e SSA $\times$ CS — Perfil B (transição) (* indica diferença significativa após correção de Holm-Bonferroni, $\alpha = 0,05$)

Perfil	Cenário	PSO $\times$ GWO	GWO $\times$ WOA	SSA $\times$ CS
B	100 $\times$ 100_escasso	<0,0001*	0,7048	0,0083*
B	100 $\times$ 100_suficiente	<0,0001*	0,0153*	0,0532
B	100 $\times$ 100_redundante	0,9563	0,0023*	0,2162
B	250 $\times$ 250_escasso	<0,0001*	0,6947	0,0153
B	250 $\times$ 250_suficiente	0,0003*	0,1650	0,0583
B	250 $\times$ 250_redundante	0,1536	0,0010*	0,1769
B	500 $\times$ 500_escasso	<0,0001*	0,7048	0,0083*
B	500 $\times$ 500_suficiente	<0,0001*	0,0153*	0,0532
B	500 $\times$ 500_redundante	0,9563	0,0023*	0,2162

Fonte: Elaborado pelo autor (2026).

A Tabela 15 conclui a série sob o Perfil C.

Tabela 15 – P-valores do teste de Wilcoxon para os confrontos PSO $\times$ GWO, GWO $\times$ WOA e SSA $\times$ CS — Perfil C (rede prioritária) (* indica diferença significativa após correção de Holm-Bonferroni, $\alpha = 0,05$)

Perfil	Cenário	PSO $\times$ GWO	GWO $\times$ WOA	SSA $\times$ CS
C	100 $\times$ 100_escasso	0,0001*	0,6602	0,0121
C	100 $\times$ 100_suficiente	0,1769	0,0007*	0,5958
C	100 $\times$ 100_redundante	0,4091	0,0032*	0,5217
C	250 $\times$ 250_escasso	0,0001*	0,7508	0,0094*
C	250 $\times$ 250_suficiente	0,0583	0,0107*	0,0897
C	250 $\times$ 250_redundante	0,9563	0,0010*	0,0583
C	500 $\times$ 500_escasso	0,0001*	0,6602	0,0121
C	500 $\times$ 500_suficiente	0,1769	0,0007*	0,5958
C	500 $\times$ 500_redundante	0,4091	0,0032*	0,5217

Fonte: Elaborado pelo autor (2026).

APÊNDICE B – Trechos-Chave do Código de Simulação

O ambiente de simulação completo — incluindo as implementações dos cinco algoritmos, o *runner* experimental, a análise estatística e a geração de figuras — está disponível no repositório público do projeto¹. Este apêndice reproduz apenas os três trechos que materializam as decisões metodológicas centrais do trabalho: a função objetivo normalizada, o cálculo do maior componente conexo e o mecanismo de sementes pareadas.

B.1 Função Objetivo Normalizada e Vetorizada

A Listagem B.1 implementa a Equação 3.1: os três componentes normalizados são calculados sobre operações vetorizadas do NumPy — as distâncias entre todos os pares de drones são computadas uma única vez e reutilizadas tanto para a penalidade de sobreposição quanto para o grafo de conectividade — e agregados pelos pesos do perfil em avaliação, recebidos como parâmetros explícitos.

Listing B.1 – Função objetivo normalizada e vetorizada.

```

1 def objective_fast(flat_positions, n_drones, radius, grid,
2                   comm_radius, max_overlap, w_cobertura, w_overlap,
3                   w_conectividade):
4     """
5     Função objetivo parametrizada para uso no experimento completo.
6
7     Diferente da versão anterior, os PESOS também são parâmetros
8     explícitos (w_cobertura, w_overlap, w_conectividade) -- não dependem
9     de W_COVERAGE/W_OVERLAP/W_CONNECTIVITY globais. Isso permite que o
10    runner alterne entre os perfis de peso (Seção 3.4.3) sem qualquer
11    risco de um perfil vazar para o cálculo de outro.
12
13    Componentes (seção 3.3 do TCC):
14    c1 -- área não coberta
15    c2 -- penalidade de overlap
16    c3 -- fragmentação da rede de comunicação
17    """
18    positions = flat_positions.reshape((n_drones, 2))
19    dmin      = 2 * radius
20
21    coverage_ratio, _ = compute_coverage_fast(positions, radius, grid)
22    c1 = 1.0 - coverage_ratio
23
24    """
25    Calculamos as distâncias entre todos os pares de drones uma única

```

¹ Disponível em: <<https://github.com/renansfreire/tcc-cobertura-drones-enxame>>.

```

25     vez, e reutilizamos esse mesmo resultado tanto para o overlap (c2)
26     quanto para a conectividade (c3) -- evita repetir o mesmo cálculo
27     duas vezes.
28     """
29     if n_drones > 1:
30         idx_i, idx_j = np.triu_indices(n_drones, k=1)
31         dists = np.linalg.norm(positions[idx_i] - positions[idx_j],
32                                axis=1)
33     else:
34         idx_i, idx_j, dists = (np.array([], dtype=int),
35                                np.array([], dtype=int), np.array([]))
36
37     if n_drones > 1:
38         overlap_sum = np.maximum(0.0, dmin - dists).sum()
39     else:
40         overlap_sum = 0.0
41     c2 = overlap_sum / max_overlap if max_overlap > 0 else 0.0
42
43     if n_drones > 1:
44         conectados = dists <= comm_radius
45         maior_grupo = compute_largest_component(
46             n_drones, idx_i[conectados], idx_j[conectados]
47         )
48         c3 = 1.0 - (maior_grupo / n_drones)
49     else:
50         c3 = 0.0
51
52     return w_cobertura * c1 + w_overlap * c2 + w_conectividade * c3

```

B.2 Maior Componente Conexo via Union-Find

A Listagem B.2 implementa a identificação do maior grupo de drones mutuamente alcançáveis — direta ou indiretamente —, que fundamenta o componente de conectividade c_3 (Equação 3.4) e a modelagem *multi-hop* discutida na Seção 3.3.

Listing B.2 – Identificação do maior componente conexo via *Union-Find*.

```

1  def compute_largest_component(n_drones, idx_i, idx_j):
2      """
3      Calcula o tamanho do maior grupo de drones mutuamente conectados,
4      usando a estrutura Union-Find (Disjoint Set Union).
5
6      Recebe os pares de drones já identificados como conectados
7      (distância <= raio de comunicação) e agrupa drones ligados direta
8      ou indiretamente -- dois drones que não se enxergam diretamente
9      ainda podem estar no mesmo grupo se conectados via um terceiro,
10     formando uma rede de retransmissão em múltiplos saltos.
11     """
12     parent = list(range(n_drones))
13
14     def find(x):
15         """Sobe na árvore até achar o representante do grupo de x,
16         comprimindo o caminho percorrido para acelerar buscas futuras."""
17         while parent[x] != x:

```

```

18         parent[x] = parent[parent[x]]
19         x = parent[x]
20     return x
21
22     def union(x, y):
23         """Funde os grupos de x e y, ligando um representante ao outro."""
24         root_x, root_y = find(x), find(y)
25         if root_x != root_y:
26             parent[root_x] = root_y
27
28     for i, j in zip(idx_i, idx_j):
29         union(int(i), int(j))
30
31     grupos = Counter(find(i) for i in range(n_drones))
32     return max(grupos.values())

```

B.3 Execução Independente com Semente Pareada

A Listagem B.3 implementa o protocolo de sementes descrito na Seção 3.4.5: o índice da execução é a própria semente, de modo que a n -ésima execução de um algoritmo parte da mesma configuração inicial nos três perfis de peso — condição que garante o pareamento exigido pelo teste de Wilcoxon.

Listing B.3 – Execução de uma *run* independente com semente pareada.

```

1 def executar_run_unica(seed, algo_func, obj, bounds, n_agents, n_iterations,
2                       n_drones, radius, grid, comm_radius):
3
4     """
5     Executa uma única run de um algoritmo com seed fixo, retornando
6     posição final, custo e métricas detalhadas. Isolada em função
7     própria para poder ser despachada tanto sequencialmente quanto em
8     paralelo (joblib) sem duplicar lógica.
9
10    Em caso de erro numérico inesperado, não propaga a exceção -- registra
11    um aviso e devolve um resultado de pior caso (custo=1.0), para que
12    uma falha isolada não derrube um experimento de horas.
13    """
14    try:
15        np.random.seed(seed)
16        pos, custo, _ = algo_func(obj, bounds, n_agents, n_iterations)
17        m = get_metrics_cenario(pos, n_drones, radius, grid, comm_radius)
18        return pos, custo, m
19    except Exception as e:
20        print(f"\n [AVISO] ERRO na run seed={seed}: {e} -- usando resultado de
21              pior caso")
22        pos_fallback = np.random.uniform(bounds[:, 0], bounds[:, 1])
23        m_fallback = get_metrics_cenario(pos_fallback, n_drones, radius, grid,
24                                         comm_radius)
25        return pos_fallback, 1.0, m_fallback

```