



UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ
PROGRAMA DE PÓS-GRADUAÇÃO EM INFORMÁTICA

GABRIEL LECHENCO VARGAS PEREIRA

**IMPLEMENTAÇÃO DE ALGORITMO DE PLANEJAMENTO DE CAMINHO DE
COBERTURA COM REGIÕES HETEROGÊNEAS PARA VEÍCULOS AÉREOS NÃO
TRIPULADOS**

DISSERTAÇÃO DE MESTRADO

CORNÉLIO PROCÓPIO
2024

GABRIEL LECHENCO VARGAS PEREIRA

IMPLEMENTAÇÃO DE ALGORITMO DE PLANEJAMENTO DE CAMINHO DE COBERTURA COM REGIÕES HETEROGÊNEAS PARA VEÍCULOS AÉREOS NÃO TRIPULADOS

Implementation of coverage path planning algorithms with heterogeneous regions for Unmanned Aerial Vehicles

Dissertação apresentada ao Programa de Pós-Graduação em Informática da Universidade Tecnológica Federal do Paraná – UTFPR, como requisito parcial para obtenção do título de Mestre em Informática.

Orientadora: Profa. Dra. Natássya Barlate Floro da Silva

CORNÉLIO PROCÓPIO
2024



[4.0 Internacional](https://creativecommons.org/licenses/by-nc-sa/4.0/)

Esta licença permite remixe, adaptação e criação a partir do trabalho, para fins não comerciais, desde que sejam atribuídos créditos ao(s) autor(es) e que licenciem as novas criações sob termos idênticos. Conteúdos elaborados por terceiros, citados e referenciados nesta obra não são cobertos pela licença.



Ministério da Educação
Universidade Tecnológica Federal do Paraná
Campus Cornélio Procopio



GABRIEL LECHENCO VARGAS PEREIRA

**IMPLEMENTAÇÃO DE ALGORITMO DE PLANEJAMENTO DE CAMINHO DE COBERTURA COM REGIÕES
HETEROGÊNEAS PARA VEÍCULOS AÉREOS NÃO TRIPULADOS**

Trabalho de pesquisa de mestrado apresentado como requisito para obtenção do título de Mestre Em Informática da Universidade Tecnológica Federal do Paraná (UTFPR). Área de concentração: Computação Aplicada.

Data de aprovação: 08 de Agosto de 2024

Dra. Natassya Barlate Floro Da Silva, Doutorado - Universidade Tecnológica Federal do Paraná

Dr. Danilo Sipoli Sanches, Doutorado - Universidade Tecnológica Federal do Paraná

Dr. Joao Vitor De Carvalho Fontes, Doutorado - Universidade Federal de São Carlos (Ufscar)

Documento gerado pelo Sistema Acadêmico da UTFPR a partir dos dados da Ata de Defesa em 08/08/2024.

RESUMO

PEREIRA, Gabriel L. V.. IMPLEMENTAÇÃO DE ALGORITMO DE PLANEJAMENTO DE CAMINHO DE COBERTURA COM REGIÕES HETEROGÊNEAS PARA VEÍCULOS AÉREOS NÃO TRIPULADOS. 66 f. Dissertação – Programa de Pós-Graduação em Informática, Universidade Tecnológica Federal do Paraná. Cornélio Procópio, 2024.

O barateamento e a popularização de tecnologias como os Veículos Aéreos Não Tripulados possibilita o crescimento da Agricultura de Precisão, por ser mais acessível do que outras formas de coleta de dados de grandes áreas de forma parcialmente ou totalmente autônoma. Isso é possível devido aos algoritmos de planejamento de caminho de cobertura, que traçam rotas para que o veículo percorra uma determinada área. Entretanto, os algoritmos existentes apresentam limitações, não permitindo regiões com diferentes características de voo para coleta de dados com outras propriedades. Assim, este trabalho visa implementar uma solução de planejamento de caminho de cobertura que considere regiões com voos heterogêneos para a coleta de imagens em diferentes altitudes durante a mesma missão. Ao alterar a altitude do voo para mais próxima do solo, é possível coletar uma maior quantidade de dados da superfície que está sendo observada. Porém, um voo todo em baixa altitude eleva a quantidade de recursos utilizados durante a missão. Portanto, um voo heterogêneo com múltiplas altitudes é uma forma de otimizar a qualidade dos dados coletados reduzindo o consumo desnecessário de recursos. A implementação do algoritmo para voos heterogêneos é realizada com foco principalmente em aeronaves multirrotores da DJI e suas ferramentas de desenvolvimento, possibilitando a implementação de um aplicativo controlador que, ao utilizar informações de GPS, consiga planejar e enviar coordenadas referentes ao caminho que o veículo deverá percorrer. Para validação da implementação, foram utilizadas coordenadas de lavouras agrícolas de diversos formatos e dimensões, demonstrando os resultados nestas diferentes localizações para voos de cobertura homogêneos tradicionais e sobrevoos heterogêneos com uma ou múltiplas subáreas especiais em uma altitude menor. Ao avaliar estes cenários, constatou-se que o planejamento de caminho de cobertura heterogêneo pode resultar em um caminho otimizado com um comprimento até 35,4% menor do que um voo homogêneo em baixa altitude, com todo o processo levando poucos milissegundos para executar.

Palavras-chave: Veículo Aéreo Não Tripulável, Planejamento de Caminho de Cobertura, Monitoramento aéreo

ABSTRACT

PEREIRA, Gabriel L. V.. Implementation of coverage path planning algorithms with heterogeneous regions for Unmanned Aerial Vehicles. 66 f. Dissertação – Programa de Pós-Graduação em Informática, Universidade Tecnológica Federal do Paraná. Cornélio Procópio, 2024.

The lower cost and popularization of technologies such as Unmanned Aerial Vehicles enable the spread of Precision Agriculture, as it is more accessible than other forms of data collection and which can fly partially or completely autonomously. This is possible due to coverage path planning algorithms, which trace vehicle routes to a specific area. However, available algorithms have limitations, not allowing regions with different flight characteristics and collecting data with other properties. This work aims to implement a coverage path planning solution that considers regions with heterogeneous flights to collect images at different altitudes during the same mission. Changing the flight altitude closer to the ground makes it possible to collect a greater amount of data from the surface being observed. However, a flight entirely at a low altitude adds resources that are used during the mission. Therefore, a heterogeneous flight with multiple altitudes is a way to optimize the quality of the data collected by reducing unnecessary resource consumption. The development uses the DJI's multirotor aircraft and its development tools, allowing the implementation of a controller application, with GPS information, that can plan and send coordinates referring to the path that the vehicle must travel. To validate the method, coordinates of agricultural crops with different shapes and dimensions were used, demonstrating the results in different locations for traditional homogeneous coverage flights and heterogeneous overflights with one or multiple special subareas at low altitude. By evaluating these scenarios, it was found that heterogeneous coverage path planning can result in an optimized path with a length up to 35.4% shorter than a homogeneous low-altitude flight, with the entire process taking a few milliseconds to execute.

Keywords: Unmanned Aerial Vehicle, Coverage Path Planning, Aerial monitoring

LISTA DE FIGURAS

FIGURA 1	– Exemplo de Caminho de Cobertura gerado em uma área de interesse côncava.	11
FIGURA 2	– Representação de VANT multirrotor	15
FIGURA 3	– Exemplos de padrões de caminhos usados em CPP	17
FIGURA 4	– Exemplo de decomposição aproximada	19
FIGURA 5	– Caminho gerado por um algoritmo de CPP com decomposição celular exata	20
FIGURA 6	– Exemplo de células da decomposição trapezoidal (à esquerda) contra a combinação de células feitas pelo algoritmo de bustrofédon (à direita)	21
FIGURA 7	– Ao aumentar a altitude de voo o sensor da câmera se projeta sobre uma superfície maior	23
FIGURA 8	– Exemplo do GSD, que representa o tamanho individual de cada pixel na imagem	24
FIGURA 9	– Diagrama de classe das classes primitivas	31
FIGURA 10	– Fluxo das etapas do algoritmo de planejamento de caminho de cobertura heterogêneo	32
FIGURA 11	– Fluxo de decomposição sem e com regiões especiais	34
FIGURA 12	– Exemplo de recorte de polígono	36
FIGURA 13	– Tipos de eventos para a divisão de polígonos côncavos	37
FIGURA 14	– Segmentos de retas perpendiculares e interseções a fim de encontrar concavidades	38
FIGURA 15	– Exemplo de iteração da decomposição de um evento MIDDLE	39
FIGURA 16	– Aplicação de uma função objetiva para todos os pares de células presentes em um grafo, gerando uma matriz objetiva	40
FIGURA 17	– Ilustração dos parâmetros utilizados durante a geração de caminho	42
FIGURA 18	– Regiões dos experimentos em alta e baixa altitude, homogêneas .	44
FIGURA 19	– Regiões dos experimentos com subáreas especiais, heterogêneas	45
FIGURA 20	– Caminhos gerados em alta e baixa altitude na região 1	47
FIGURA 21	– Caminho gerado com subáreas especiais na região 1	48
FIGURA 22	– Caminho gerado em alta e baixa altitude na região 2	50
FIGURA 23	– Caminho gerado com subáreas especiais na região 2	52
FIGURA 24	– Caminho gerado em alta e baixa altitude na região 3	53
FIGURA 25	– Caminho gerado com subáreas especiais na região 3	54

LISTA DE TABELAS

TABELA 1	– Síntese dos trabalhos relacionados	27
TABELA 2	– Cenários dos experimentos	45
TABELA 3	– Métricas dos experimentos da região 1	47
TABELA 4	– Métricas dos experimentos da região 2	51
TABELA 5	– Métricas dos experimentos da região 3	55
TABELA 6	– Métricas de todos os experimentos	56
TABELA 7	– Duração de cada etapa do algoritmo para todos os experimentos ..	65

LISTA DE SIGLAS

PIB	Produto Interno Bruto
AP	Agricultura de Precisão
GPS	<i>Global Positioning System</i>
VANTs	Veículos Aéreos Não Tripulados
CPP	<i>Coverage Path Planning</i>
LiDAR	<i>Light Detection and Ranging</i>
GSD	<i>Ground Sample Distance</i>
TSP-CPP	<i>Traveling-Salesman Problem CPP</i>
SDK	<i>Software Development Kit</i>
JSON	<i>JavaScript Object Notation</i>
JVM	<i>Java Virtual Machine</i>

SUMÁRIO

1	INTRODUÇÃO	10
1.1	Objetivos	12
1.1.1	Objetivos específicos	12
1.2	Organização do texto	13
2	REVISÃO DA LITERATURA	14
2.1	VANTs	14
2.1.1	Tipos de VANTs	14
2.1.2	Multirrotores	15
2.1.3	Aplicações na Agricultura de Precisão	16
2.2	Planejamento de Caminho de Cobertura	17
2.2.1	Área de interesse	18
2.2.2	Decomposição Celular Aproximada	19
2.2.3	Decomposição Celular Semi-aproximada	19
2.2.4	Decomposição Celular Exata	20
2.3	Captura de imagens	22
2.4	Trabalhos Relacionados	24
3	DESENVOLVIMENTO	28
3.1	Organização do algoritmo	29
3.1.1	Informações de entrada	29
3.1.2	Classes primitivas	31
3.1.3	Algoritmo de planejamento de caminho de cobertura heterogêneo	32
3.2	Decomposição celular	33
3.2.1	Sobreposição de áreas	33
3.2.2	Decomposição celular das subáreas	36
3.3	Geração do melhor caminho	39
3.3.1	Ordem de visita das células	39
3.3.2	Caminho de cobertura	41
3.3.3	Mapeamento de <i>waypoints</i>	43
3.4	Experimentos	43
4	RESULTADOS	46
4.1	Região 1	46
4.2	Região 2	50
4.3	Região 3	52
4.4	Comparação dos experimentos	55

5 CONCLUSÃO	57
5.1 Trabalhos futuros	58
REFERÊNCIAS	61
Apêndice A – TEMPO DE EXECUÇÃO DE CADA ETAPA DO ALGORITMO PARA OS EXPERIMENTOS	65

1 INTRODUÇÃO

A agricultura brasileira ocupa uma posição de grande destaque no cenário internacional, gerando empregos, alimentos e energia para o Brasil e outros países, sendo responsável por 43% das exportações em 2019 (GOV, 2020) e por 24,8% do PIB (Produto Interno Bruto) nacional no ano de 2022 (LUCCHI; SILVA; BARROS, 2023). Esse cenário de abundância é relativamente recente, no qual em pouco mais de 4 décadas a produção de grãos no país foi de 38 milhões de toneladas em 1975 para 236 milhões em 2017. Esse crescimento, porém, não vem do aumento da área plantada, o qual apenas dobrou durante o mesmo período (EMBRAPA, 2020), e sim da evolução e desenvolvimento de novas tecnologias referentes à produção e ao gerenciamento das plantações.

A Agricultura de Precisão (AP) refere-se ao conjunto de sistemas e tecnologias que visam coletar e processar dados a fim de melhorar o manejo do solo e a produção de insumos e culturas (EMBRAPA, 1997). Isso faz com que o produtor tenha um conhecimento de características específicas de cada parte de sua propriedade, e não trate a plantação como algo uniforme (TSCHIEDEL; FERREIRA, 2002). Partindo da análise dos dados coletados, a AP também se beneficia de automatizações de processos do campo, podendo otimizar estes para favorecer a própria produção e diminuir o consumo de recursos necessários para a manutenção das plantas e do solo.

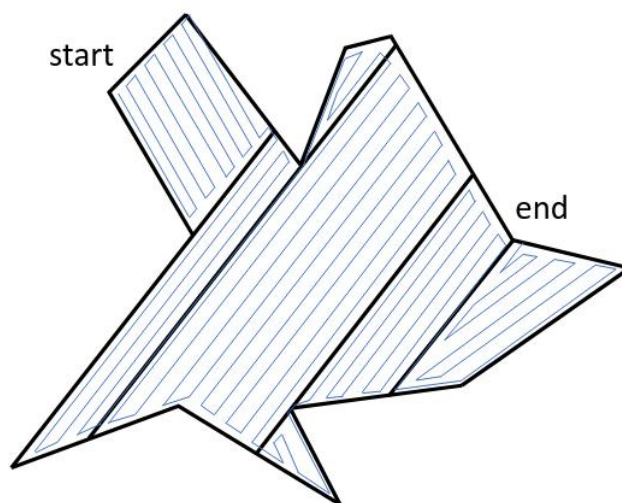
O uso de imagens de satélites ou coletadas por um avião, junto com informações do Sistema de Posicionamento Global (do inglês *Global Positioning System* – GPS), possibilitam uma análise detalhada das peculiaridades do terreno e da plantação e de como essas variam no espaço e no tempo. A partir dessa avaliação o produtor pode tomar decisões mais precisas e ter um controle mais fino sobre os recursos disponíveis (ZHANG; WANG; WANG, 2002), já que as imagens geradas em conjunto com dados de outros sensores podem indicar deficiências ou características em áreas específicas, permitindo o gerenciamento por meio de ações pontuais e que não envolvam toda a plantação, evitando o uso desnecessário de recursos. Entretanto, esses métodos de coleta de imagens podem ter um custo elevado devido ao uso de

equipamentos e mão de obra especializada (SHIRATSUCHI, 2014).

O uso de Veículos Aéreos Não Tripulados (VANTs), também conhecidos como *drones*, tem surgido como uma alternativa mais acessível para realizar a coleta de dados em plantações, sendo um importante aliado na implementação da AP (OLIVEIRA et al., 2020). Com seu custo relativamente baixo e o crescente interesse no desenvolvimento de novas soluções de software, os *drones* acabam sendo equipamentos versáteis, podendo sobrevoar uma área determinada e gerando dados de interesse do produtor rural. Sendo capaz de gerar, por exemplo, mapas topográficos e modelos para nivelamento e drenagem, podendo ser utilizados também para medir a altura das plantas e outras condições gerais da lavoura, localizando possíveis infestações de plantas daninhas e pragas pela plantação (SHIRATSUCHI, 2014).

Apesar das inúmeras vantagens e praticidade destes veículos, suas missões são limitadas principalmente pelo tempo de duração do voo com uma única carga de bateria, diminuindo o seu alcance e sendo necessário um reabastecimento caso se queira cobrir regiões maiores (SHARMA; HEMA, 2021). Para aumentar esse alcance e tempo de voo, muitos estudos da literatura procuram aumentar a eficiência da missão, seja otimizando os caminhos planejados (Figura 1) por meio de novos algoritmos (COOMBES; CHEN; LIU, 2018; RABELLO et al., 2020), ou considerando os melhores valores para as variáveis físicas que impactam na performance de uma determinada aeronave (FRANCO; BUTTAZZO, 2015).

Figura 1: Exemplo de Caminho de Cobertura gerado em uma área de interesse côncava.



Fonte: Cabreira, Brisolara e Ferreira Jr (2019)

Por outro lado, um tópico que ainda não foi amplamente abordado é o de planejamento de caminhos contendo subregiões com diferentes propriedades de voos, pois a maioria das soluções utiliza dos mesmos parâmetros para toda a área a ser monitorada. Essas propriedades podem influenciar diretamente na coleta de dados e na duração da missão da aeronave, sendo necessário um maior número de reabastecimentos em sobrevoos maiores. Se o planejamento de caminho de cobertura considerar diferentes parâmetros em uma mesma missão, pode-se mitigar o aumento da duração do voo ao considerar parâmetros mais custosos apenas em subregiões que necessitam de uma análise mais detalhada, proporcionando missões de cobertura dinâmicas. Dessa forma, aproveitando-se um único sobrevoo para realizar coletas com propriedades variadas.

1.1 OBJETIVOS

Visto as deficiências citadas anteriormente, este trabalho tem como objetivo principal a implementação de um novo algoritmo de Planejamento de Caminho de Cobertura (do inglês *Coverage Path Planning* - CPP) com a possibilidade de configurar a propriedade de altitude do voo em diferentes subregiões. Isso possibilita a coleta de imagens mais próximas do solo em áreas suscetíveis às doenças ou que necessitam de atenção, a fim de capturar uma maior quantidade de detalhes para serem processadas posteriormente. Dessa forma, o algoritmo deve gerar rotas para sobrevoos de *drones* multirrotores em lavouras, mapeando as posições que a câmera deve ser acionada para capturar imagens da plantação. O algoritmo deve ser capaz de realizar essa tarefa para áreas de diversos formatos, convexas e não convexas, e que não contenham obstáculos que possam interferir no voo da aeronave.

1.1.1 OBJETIVOS ESPECÍFICOS

Assim, este trabalho descreve um novo algoritmo capaz de planejar caminhos de coberturas com múltiplas altitudes no decorrer da área de interesse, a fim de coletar imagens com características distintas. O mesmo deve realizar um planejamento considerado como exato e que produza um caminho que realize a cobertura total da região solicitada.

1.2 ORGANIZAÇÃO DO TEXTO

Este documento está organizado da seguinte forma: após esta breve introdução, o Capítulo 2 comporta os diferentes conhecimentos necessários para entender este trabalho como um todo, além de revisar diferentes fontes da literatura. Após estabelecidos os conceitos e o estado da arte atual, o Capítulo 3 descreve o método proposto e as decisões tomadas durante o desenvolvimento, ao final também são descritos os experimentos realizados para avaliar a solução desenvolvida. O Capítulo 4 lista e compara os diferentes resultados obtidos pelos experimentos, demonstrando os pontos fortes e de melhoria do algoritmo. Por fim, o Capítulo 5 conclui o trabalho com uma breve revisão do documento e as diversas possibilidades para futuros trabalhos.

2 REVISÃO DA LITERATURA

Ao projetar e implementar um algoritmo de Planejamento de Caminho de Cobertura para VANTs, deve-se entender como estes últimos funcionam e as diferentes abordagens para se resolver o problema de cobertura. Dessa forma, a Seção 2.1 relata as principais características dos VANTs, como estes podem ser classificados, que tipo de dados conseguem coletar e como podem ser utilizados em soluções na agricultura. Em seguida, a Seção 2.2 apresenta o problema de CPP, citando suas peculiaridades e abordagens já conhecidas para solucioná-lo. Por último, a Seção 2.4 cita trabalhos encontrados na literatura com objetivos similares, descrevendo e analisando suas principais contribuições, com o fim de entender como este trabalho pode utilizar destas contribuições para propor um método posteriormente.

2.1 VANTS

Os VANTs, popularmente conhecidos como *drones*, são máquinas aéreas que não necessitam de uma pessoa a bordo para que funcionem de forma adequada. Seu funcionamento pode ser controlado de maneira remota ou mesmo totalmente autônoma, sem a obrigação de intervenção humana nas operações de decolagem e nas manobras de voo e de pouso da aeronave (VALAVANIS, 2007).

2.1.1 TIPOS DE VANTS

Existem diversos tipos de veículos aéreos que se encaixam na definição de VANT, pois esta se refere apenas à ausência de uma tripulação a bordo, não limitando as possíveis formas, tamanhos ou qualquer outra característica destes. A forma mais comum de se classificar um veículo aéreo não tripulado é tomando como base o modo como este se sustenta no ar, podendo ser com asas fixas, rotores, asas de parapólio, balões e dirigíveis, etc. (SHARMA; HEMA, 2021).

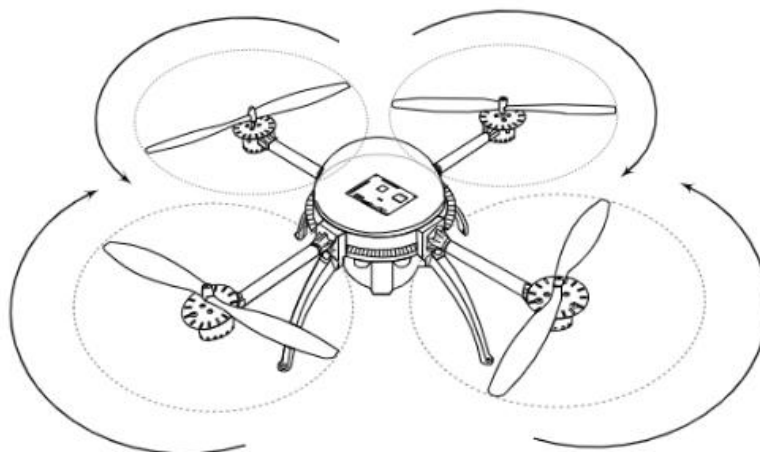
VANTs de asas fixas e com rotores são amplamente utilizados na Agricultura

de Precisão, no qual o primeiro é frequentemente utilizado em missões que necessitam de um sobrevoo em uma área maior, enquanto o segundo é aproveitado em missões que preferem um voo mais controlado e próximo do chão (HRISTOV et al., 2020). Assim sendo, a melhor aeronave será aquela que irá atender de maneira satisfatória os requisitos da aplicação proposta.

2.1.2 MULTIRROTORES

Neste trabalho, optou-se por utilizar aeronaves multirrotores (Figura 2), por conta de suas características de voo que permitem um sobrevoo mais próximo do chão e mais estável, facilitando a coleta de dados e a realização de manobras. Contudo, os multirrotores tem como principal desvantagem em relação aos VANTs de asa fixa a necessidade de produzir toda a energia para se manter no ar, enquanto que com asas é possível planar e percorrer uma área maior consumindo menos energia (VALAVANIS; VACHTSEVANOS, 2015). Sendo então o consumo de energia e o tempo máximo de voo limitantes para aeronaves semelhantes à ilustrada pela Figura 2, e variáveis importantes para as aplicações que se utilizam destas.

Figura 2: Representação de VANT multirrotor



Fonte: Silva (2018)

Aeronaves como as da Figura 2 tem sua força de propulsão produzida pelos seus rotores, que giram de modo a anularem o torque uns dos outros, ainda permitindo o controle independente da velocidade de cada rotor para criar condições que movam a aeronave livremente por qualquer direção de um espaço 3D (SILVA, 2018). Essa facilidade em realizar manobras possibilita um equipamento seguro e com custos relativamente baixos para fazer experimentos com mapeamento, e navegação em

3 dimensões. Uma vantagem em comparação com as aeronaves de asa fixa é a habilidade de ficar parado no ar por um tempo arbitrário, além da possibilidade de decolar e pousar verticalmente (VALAVANIS; VACHTSEVANOS, 2015).

2.1.3 APLICAÇÕES NA AGRICULTURA DE PRECISÃO

A agricultura tem se beneficiado com a utilização de novas tecnologias, oferecendo o melhoramento dos sistemas produtivos, otimizando recursos e reduzindo efeitos adversos no ambiente. Viabilizando a padronização e eficiência da área plantada, a Agricultura de Precisão tem como principal finalidade a otimização da produção e a manutenção da qualidade do cultivo (OLIVEIRA et al., 2020). Uma tecnologia que pode ser utilizada nesse contexto justamente são os VANTs, que permitem a coleta de dados de toda a área plantada, cobrindo até alguns hectares de terra em um único voo (MOGILI; DEEPAK, 2018).

Algumas aeronaves são capazes de carregar, além dos componentes necessários para o voo em si, sensores que possibilitam a captura de dados e as mais variadas aplicações. Entre os sensores frequentemente utilizados estão diversos tipos de câmeras: RGBs, térmicas, multiespectrais e hiperespectral, além de outros como LiDAR (do inglês *Light Detection and Ranging*) e infravermelho (SHARMA; HEMA, 2021).

Com o uso destes, o VANT se torna útil no cenário agrícola já que permite um voo mais baixo do que aviões convencionais sem perder a qualidade dos dados gerados. Para isso, as aeronaves devem sobrevoar toda uma área de interesse a fim de registrar com suas câmeras e sensores dados relevantes. Esse processo de coleta de imagens pode ser feita de forma totalmente autônoma pelo VANT resultando, após uma fase de processamento, em mapas de alta qualidade da região do sobrevoo (HRISTOV et al., 2020).

A partir desses mapas da plantação, segundo Sharma e Hema (2021), é possível realizar análises de qualidade do solo, fazer o controle do plantio, identificar a presença ou o avanço de pragas, monitorar e realizar estimativas sobre as colheitas, controlar a irrigação, e verificar outras condições que possam impactar na saúde das plantas.

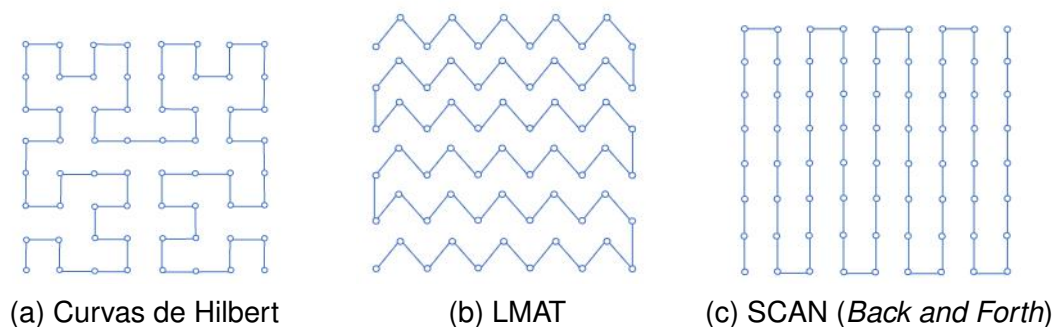
Para manter a eficiência das missões e aproveitar ao máximo o tempo de voo já limitado desses veículos aéreos, diversos estudos propõem algoritmos para

otimizar o caminho que o VANT deve sobrevoar. Estes estudos serão explorados pela Seção 2.4, mostrando diferentes métodos para se alcançar um resultado perto do ótimo. A seguir, porém, será descrito o problema de planejamento de caminho de cobertura e os desafios encontrados.

2.2 PLANEJAMENTO DE CAMINHO DE COBERTURA

O Planejamento de Caminho de Cobertura se refere a ação de especificar um percurso para que o robô possa percorrer todos os pontos de uma área ou volume pré-determinado (GALCERAN; CARRERAS, 2013), permitindo que o robô possa cobrir todo o ambiente de forma autônoma. Na Figura 3 pode-se observar alguns dos padrões gerados por essas soluções a fim de cumprir o seu objetivo. Essa cobertura, porém, deve considerar as restrições de movimento do robô e as propriedades dos seus sensores (CABREIRA; BRISOLARA; FERREIRA JR, 2019).

Figura 3: Exemplos de padrões de caminhos usados em CPP



Fonte: Cabreira, Brisolara e Ferreira Jr (2019)

A classificação de algoritmos de CPP pode ser feita de duas formas distintas: entre *offline* e *online* ou entre algoritmos heurísticos e completos (GALCERAN; CARRERAS, 2013). A primeira classificação se refere a quando o planejamento é feito, caso o algoritmo interprete a área de interesse durante a movimentação do robô, ele é classificado como um algoritmo *online*, caso o percurso seja definido previamente, e o equipamento apenas o execute, é considerado um algoritmo *offline*. A segunda classificação é definida por Choset (2001) e julga a presença de informação inicial sobre a área de interesse. Em uma abordagem heurística não se tem todas as informações da região na qual o robô irá atuar, sendo necessário a adição de regras que determinem o comportamento do mesmo e a utilização de seus sensores para se localizar e identificar obstáculos, não garantindo a cobertura completa da área

(CABREIRA; BRISOLARA; FERREIRA JR, 2019). Entretanto, a abordagem completa costuma ter à disposição o *layout* da área de interesse e seus obstáculos previamente, podendo definir um padrão de movimento simples que cubra a mesma de forma eficiente.

Como este trabalho visa realizar o planejamento de caminho em áreas rurais em um contexto que estão disponíveis imagens de satélites e softwares que podem facilitar a delimitação de uma região de interesse, a seguir o texto irá detalhar mais sobre as características de algoritmos de planejamento de caminho de cobertura completos.

2.2.1 ÁREA DE INTERESSE

A área de interesse A se refere a uma região predeterminada a qual se deve percorrer tanto para coletar dados ou executar uma ação sobre ela, podendo ser aplicada para regiões 2D ou 3D. Sua forma pode ser delimitada por um conjunto V de n vértices, $V = \{v_1, v_2, \dots, v_n\}$. Ao conectar cada um desses vértices v_i com o próximo $v_{proximo(i)}$ obtêm-se as arestas que compõem A , sendo $proximo(i) = i + 1$ para $i < n - 1$ e $proximo(i) = 1$ para $i = n$ (CABREIRA; BRISOLARA; FERREIRA JR, 2019).

Para poder cumprir seu objetivo de cobrir a área de interesse, os algoritmos de CPP completos costumam utilizar um padrão de movimento simples para percorrer todo o local requisitado, sendo comum utilizar padrões de ida-e-volta (*back and forth* em inglês) e espiral no planejamento do caminho. Portanto, o formato da região de interesse é relevante para o algoritmo de CPP, já que polígonos constituídos de muitas arestas acabam adicionando complexidade para o problema, principalmente caso o algoritmo se proponha a solucionar o planejamento de caminho de cobertura para regiões côncavas. Para abordagens completas é comum subdividir a região A em unidades menores e mais simples nesses casos, facilitando o trabalho do algoritmo de planejamento por meio dessas decomposições, já que o planejamento pode ser realizado cobrindo cada sub-região e navegando entre elas (CABREIRA; BRISOLARA; FERREIRA JR, 2019).

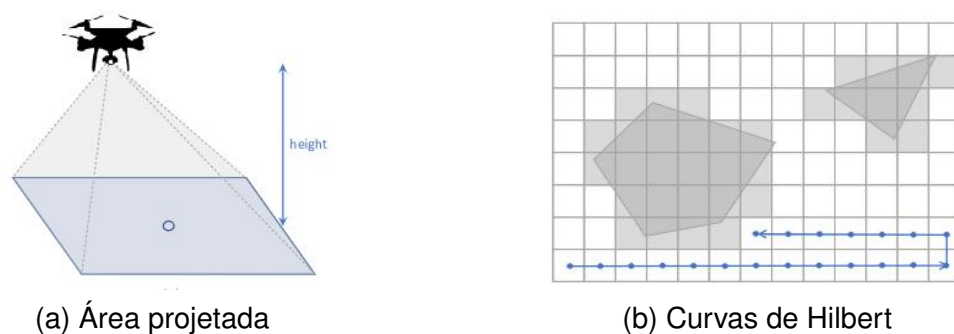
Como essas subdivisões são feitas e como se comportam com regiões de diferentes complexidades dependem do método aplicado, sendo geralmente divididos entre método de decomposição aproximada, decomposição semi-aproximada e decomposição exata (CHOSSET, 2001).

2.2.2 DECOMPOSIÇÃO CELULAR APROXIMADA

A decomposição celular aproximada é caracterizada principalmente pelo emprego de uma grade quadriculada para representar a região de interesse, sendo cada quadrado uma célula que representa um espaço que já foi visitado, que deve ser visitado ou se apresenta um obstáculo. Métodos de decomposição aproximada foram propostos a princípio por Moravec e Elfes (1985) ao desenvolverem um sistema autônomo de mapeamento de áreas desconhecidas com sonares.

O tamanho das células utilizado nesse tipo de decomposição normalmente é menor ou igual à área que o sensor consegue capturar (Figura 4), facilitando o gerenciamento do posicionamento do robô e de quais pontos já foram visitados. Dessa forma, o algoritmo deve planejar um caminho para que todas as células sejam visitadas. Entretanto, Thrun (1998) destaca as limitações deste método: o planejamento pode ser ineficiente em espaços complexos, já que o tamanho das células não dependem dessa complexidade, além da quantidade de memória necessária em regiões maiores; outro detalhe é que a determinação da localização do robô deve ser precisa, pois discrepâncias entre os sensores de movimento e a movimentação em si podem gerar equívocos no resultado final. Esse método acaba sendo utilizado normalmente em ambientes internos e com sensores com alcances relativamente pequenos (GALCERAN; CARRERAS, 2013).

Figura 4: Exemplo de decomposição aproximada



Fonte: Cabreira, Brisolara e Ferreira Jr (2019)

2.2.3 DECOMPOSIÇÃO CELULAR SEMI-APROXIMADA

Essa classificação, segundo Choset (2001), se dá para as soluções que realizam uma discretização parcial do espaço, sendo composto por células de largura

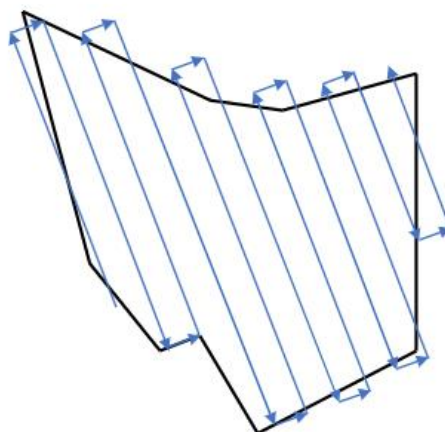
fixa, mas com alturas flexíveis. Dessa forma, o robô caminha em um movimento de ida e volta pelo espaço, respeitando a largura das células e prevendo possíveis obstáculos para que não tenha necessidade de passar mais de uma vez pela mesma região.

Um exemplo de solução com esta estratégia é demonstrada por Hert e Tiwari (1996) com um veículo autônomo subaquático para realizar o mapeamento de áreas do fundo do mar, podendo ser aplicado tanto em 2D quanto 3D.

2.2.4 DECOMPOSIÇÃO CELULAR EXATA

Como o nome sugere, abordagens de Decomposição Celular Exata buscam dividir a área de interesse A em um conjunto de n subáreas menores e que não se intersectam, de forma que a concatenação dessas subáreas deve ser igual à área original, ou seja $A = \{A_1, A_2, \dots, A_n\}$ (CHOSSET; PIGNON, 1998). Uma vez que as células simplificadas são obtidas, pode-se utilizar um algoritmo simples para realizar a cobertura de cada uma delas e ordenar a ordem que devem ser visitadas. Por fim, obtêm-se um caminho que percorre toda a área A , como ilustrado pela Figura 5.

Figura 5: Caminho gerado por um algoritmo de CPP com decomposição celular exata



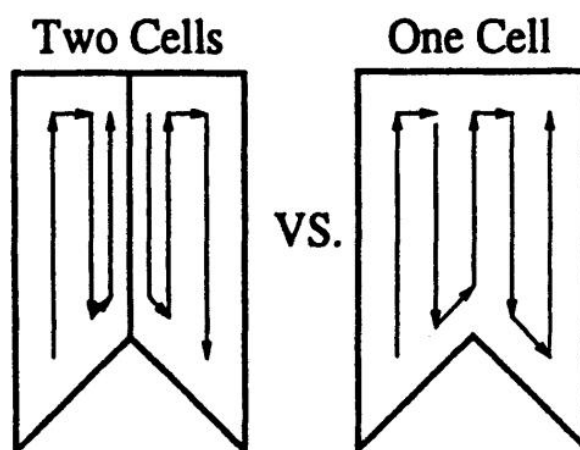
Fonte: Cabreira, Brisolara e Ferreira Jr (2019)

Um algoritmo clássico que segue esse comportamento é a decomposição trapezoidal, proposto em Latombe (1991), o qual varre o polígono e traça uma linha da aresta mais próxima acima e abaixo para cada vértice encontrado, segmentando a área em células com formatos trapezoidais, por isso o nome. Logo após, um grafo de conectividade é gerado, o qual abstrai as células e como essas se dispõem entre elas.

A partir desse grafo é gerado um caminho que percorre todas as células uma única vez, cobrindo todo o polígono.

Outra solução para esse problema é a apresentada por Choset e Pignon (1998), conhecida como Decomposição Celular de Bustrofédon (se referindo ao caminho de ida e volta que um touro faz com o arado). A princípio, essa solução é bastante similar ao algoritmo anterior, porém com a contribuição de trazer uma decomposição que combina células similares com o propósito de reduzir o número final de células (Figura 6) e evitar que o robô percorra duas vezes uma mesma região por ser perto da fronteira de duas células. Dessa forma, há uma redução do tamanho total do percurso e aumenta-se a eficiência das soluções.

Figura 6: Exemplo de células da decomposição trapezoidal (à esquerda) contra a combinação de células feitas pelo algoritmo de bustrofédon (à direita)



Fonte: Choset e Pignon (1998)

Para que fosse possível a aplicação de CPP em espaços não-poligonais, Acar et al. (2002) apresenta a Decomposição Celular Morse, a qual se baseia na solução de Choset e Pignon (1998), mas com a vantagem de ter à disposição múltiplas funções Morse que alteram o formato das células e permitem que o robô se movimente em outros padrões, como, por exemplo, um movimento circular. Entretanto, diferente dos demais, este método pode ser considerado um algoritmo de CPP online, por utilizar informações obtidas por sensores para encontrar todos os pontos críticos da área de interesse (GALCERAN; CARRERAS, 2013).

Outros detalhes podem ser considerados para aumentar a eficiência em cada cenário. Por exemplo, no caso de VANTs as curvas são manobras que consomem

mais energia, pois requerem que o robô reduza velocidade, rotacione e aumente a velocidade novamente (CABREIRA; BRISOLARA; FERREIRA JR, 2019). Logo, reduzir o número de curvas é uma possibilidade para aumentar a eficiência da solução. Na Seção 2.4, serão descritas outras implementações e será discutido como cada uma destas se beneficiam de características diferentes para buscar um melhor resultado.

2.3 CAPTURA DE IMAGENS

Uma vez com o caminho definido, este pode ser repassado para o controlador de missão do veículo não tripulado, que irá sobrevoar a região planejada para a coleta de dados. Diversos são os dados que podem ser registrados pelos sensores acoplados a um VANT. Dito isso, este trabalho terá como foco a captura de imagens geradas por uma câmera perpendicular ao solo.

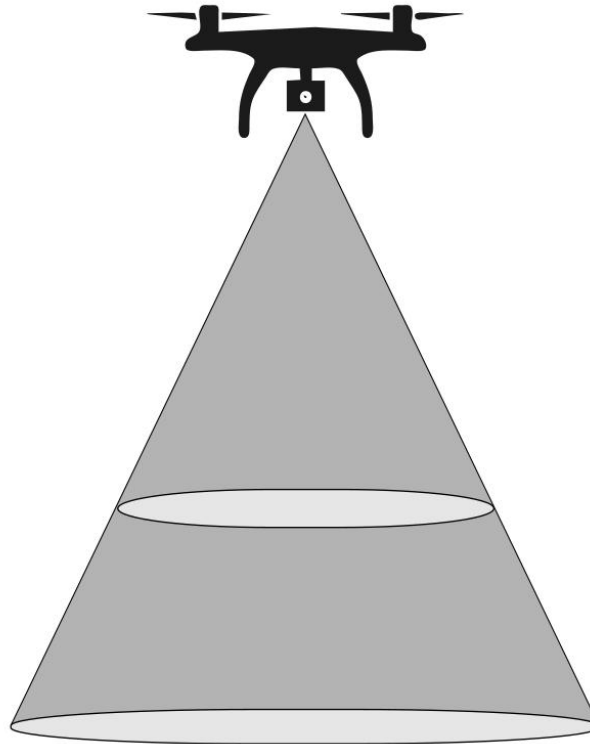
Ao realizar os registros de inúmeras imagens que serão analisadas ou processadas posteriormente, é importante ter um cuidado para escolher e manter a distância adequada entre a câmera e a superfície de estudo para que as imagens tenham propriedades de qualidade de detalhes e escala similares. Como é ilustrado pelo trabalho de Wu et al. (2024), o qual calcula o melhor posicionamento que o VANT deve tomar para que registre fotos com a mesma resolução de diferentes objetos de interesse presentes em um campo de obras.

Como é ilustrado pela Figura 7, quanto mais distante da superfície de interesse, maior é a área que a câmera consegue capturar em uma única fotografia. Entretanto, devido às limitações das câmeras digitais, isso afeta diretamente os detalhes que podem ser observados na imagem produzida, pois quanto maior a área de captura maior também será a área registrada por cada pixel.

Portanto, é necessário estar ciente de tais propriedades da captura de imagens digitais e realizar as fotografias normalizando a resolução de cada imagem digital em relação à área da superfície que está sendo observada pela câmera. Para isso, utiliza-se a distância de amostra terrestre (do inglês, *Ground Sample Distance – GSD*) para entender esta relação a partir das propriedades físicas da câmera (WU et al., 2024).

O GSD informa a dimensão que um único pixel representa da superfície observada, como visto na Figura 8. Para aferir este valor deve-se primeiro ter informações específicas sobre a câmera que está sendo utilizada, como a distância

Figura 7: Ao aumentar a altitude de voo o sensor da câmara se projeta sobre uma superfície maior



Fonte: Elaborado pelo autor

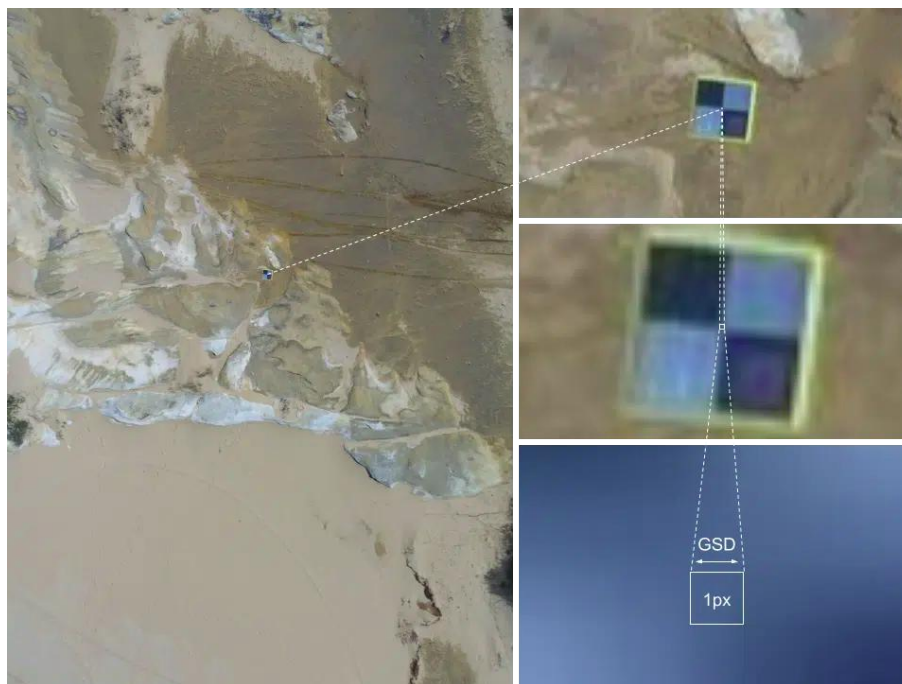
focal f e o tamanho do sensor da câmera $sensor_{size}$. Com este último e a resolução da imagem, pode-se calcular o tamanho que cada pixel $pixel_{size}$ representa no sensor por meio da Equação (1) em $mm/pixel$. Uma vez com o valor obtido, é possível calcular o GSD pela Equação (2) (FELIPE-GARCIA; HERNANDEZ-LOPEZ; LERMA, 2012), onde H' é a altura do voo da aeronave.

$$pixel_{size} = \frac{sensor_{size}}{image_{size}} \quad (1)$$

$$\frac{f}{H'} = \frac{GSD}{pixel_{size}} \quad (2)$$

Da mesma forma que a resolução da imagem é influenciada pela distância entre a câmera e a superfície de interesse, ao transportar isso para a região de interesse em um planejamento de caminho de cobertura, quanto menor a distância entre a câmera e a superfície menor a área registrada, e maior será o número de

Figura 8: Exemplo do GSD, que representa o tamanho individual de cada pixel na imagem



Fonte: Wawrzyn (2023)

fotos necessárias para se realizar a cobertura completa. Portanto, essa distância não afeta apenas os dados coletados, mas inclusive a missão de coleta de dados como um todo. Visto que VANTs apresentam um baixo tempo de voo antes do descarregamento total de sua bateria interna, é necessário equilibrar o GSD alvo entre uma resolução de imagem interessante para visualizar as características exigidas pelo estudo e considerando também o tamanho total da área e a duração da missão que será executada pela aeronave não tripulada.

2.4 TRABALHOS RELACIONADOS

Ao analisar os trabalhos da literatura que investigam formas de um veículo aéreo não tripulado sobrevoar toda uma área determinada de forma eficiente são encontrados diversos tipos de algoritmos e soluções como redes neurais e outras técnicas de aprendizado de máquina (YU et al., 2021; LEI et al., 2022) e autômatos celulares (ZELENKA; KASANICKÝ, 2014; SONG et al., 2019), além de algoritmos clássicos como o de Bustrofédon (CHOSSET; PIGNON, 1998; LIU; LI; LIU, 2021; BAHNEMANN et al., 2021). Apesar de cada trabalho buscar solucionar ou aperfeiçoar um tópico específico, sua seleção proporciona uma visão do estado da arte atual.

Como dito anteriormente, uma limitação dos VANTs é o tempo de voo por conta do consumo de energia, portanto uma área de interesse abordada em pesquisas é a de encontrar novos métodos que aumentem a eficiência e o tempo de missão, tornando o voo mais eficiente. Franco e Buttazzo (2015), por exemplo, fazem uma análise minuciosa do seu caso de uso ao procurar estimar o consumo de energia de um modelo específico de VANT em função da sua velocidade de voo e outras condições de operação. Seu modelo energético investiga a relação dessas diferentes variáveis e cenários, e como trechos de aceleração e desaceleração podem afetar o consumo de energia. Em seguida, esses parâmetros são utilizados por um algoritmo de CPP ciente do consumo de energia da aeronave, junto com um mecanismo que checa o estado atual da bateria e garante um retorno seguro para o local inicial caso não haja energia suficiente para concluir a missão.

Um modo para se solucionar o problema de CPP com uma abordagem aproximada é apresentada por Song et al. (2019), que propõem o uso de um autômato celular que consegue preencher todas as células de um espaço limitado e com possíveis obstáculos, formando então um caminho com padrão em espiral. Para determinar o tamanho da célula foi utilizada a área que o sensor do VANT consegue capturar em uma única leitura, simplificando o problema. Entre as vantagens de seu algoritmo, Song et al. (2019) cita a facilidade e flexibilidade para se implementar, e que, apesar de ter sido feito com um padrão em espiral, pode ser facilmente ajustado para prover outros.

Modelos de aprendizado de máquina também podem ser utilizados, como demonstrado em Yu et al. (2021), no qual um algoritmo de *Q-learning* é treinado para tomar decisões que otimizem o consumo de energia durante o voo e comunicação com diversas estações-base. Ao final, o método pode encontrar um caminho otimizado em um tempo finito, alcançando um gasto energético mais eficiente quando comparado com os outros algoritmos testados pelo estudo, entre eles o Zigzag e o EG-CPP. Em adição a esses resultados, o trabalho também contribuiu com um modelo de consumo de energia de todo o voo e uma nova estratégia de voo utilizando computação de borda.

Outro exemplo de aprendizado de máquina é o algoritmo de aprendizagem profunda proposto por Lei et al. (2022). Ao analisar o ambiente por diferentes camadas, a solução acaba focando em uma parte específica do problema em cada uma delas, conseguindo ao final resolver o mapeamento do ambiente, realizar a

geração do caminho com uma solução de CPP completo, e fazer o ajuste para desviar de obstáculos de forma dinâmica. Portanto, o trabalho tem como produto um algoritmo leve que separa o problema em três camadas hierárquicas, permitindo o planejamento de caminho offline e o replanejamento da rota em tempo de execução para contornar possíveis obstáculos encontrados pelos sensores.

Um trabalho que considera áreas de diferentes prioridades, necessitando de características de voos específicos em cada uma destas, é demonstrado por Liu, Li e Liu (2021), no qual o algoritmo é informado não só da área de interesse, mas também de sub-áreas que devem ser ou evitadas durante o planejamento, ou tratadas com uma maior prioridade, permitindo o desvio da aeronave de sua rota tradicional para coletar informações com maior qualidade destas sub-áreas. Ao possibilitar que essas restrições sejam fornecidas de forma explícita, e as considerar ao se construir o caminho de cobertura, a solução permite uma maior flexibilidade ao se preparar as missões de voo.

Além de resolver o planejamento de caminho de cobertura, Xie, Carrillo e Jin (2020) associam o problema de CPP ao do caixeiro viajante (TSP-CPP, do inglês *Traveling-Salesman Problem CPP*) para tentar encontrar a melhor rota para cobrir múltiplas áreas em uma única missão. Com isso, é possível encontrar soluções próximas do ótimo ao sobrevoar e ordenar as áreas dadas como entrada.

Outro trabalho de otimização de caminhos de cobertura é descrito por Stefanopoulou et al. (2024), no qual são considerados múltiplos parâmetros para coordenar o comportamento de inúmeras aeronaves em uma mesma área de interesse podendo ter a presença de subregiões proibidas. Tendo como principal contribuição a demonstração de como as diferentes posições iniciais de cada aeronave impactam nos trajetos finais, otimizando dessa forma o consumo de energia e minimizando a revisitação de um mesmo ponto por mais de uma aeronave.

A Tabela 1 sintetiza todas as informações apresentadas nesta seção de forma a facilitar a comparação dos trabalhos, além de identificar suas principais contribuições. Entretanto, nenhum dos trabalhos apresentados traz uma solução que abrange um planejamento de caminho de cobertura com a presença de regiões heterogêneas com diferentes estratégias de cobertura. Liu, Li e Liu (2021) desenvolvem uma solução que considera sim desvios para priorizar regiões de alta prioridade e evitar zonas proibidas, mas estas regiões de alta prioridade são apenas pequenos pontos de interesses para os quais o veículo realiza um pequeno desvio

para centralizar o sensor da câmera na região.

Tabela 1: Síntese dos trabalhos relacionados

Trabalho	Regiões Heterogêneas	Principal Contribuição
Franco e Buttazzo (2015)	Não	Algoritmo otimizado e ciente do gasto energético da aeronave.
Song et al. (2019)	Não	Algoritmo flexível e fácil de implementar, podendo utilizar outros padrões.
Yu et al. (2021)	Não	Apresenta um planejamento mais eficiente comparado com outros algoritmos.
Lei et al. (2022)	Não	Algoritmo em 3 camadas com replanejamento em tempo real.
Xie, Carrillo e Jin (2020)	Não	Planejar caminhos para cobrir regiões distintas de forma eficiente.
Liu, Li e Liu (2021)	Sim	Considera áreas restritas e de alta prioridade.
Stefanopoulou et al. (2024)	Sim	Considera posição de lançamento e múltiplos veículos.

Fonte: Elaborado pelo autor

Como citado anteriormente, os estudos encontrados e descritos aqui não procuram apresentar uma solução que gere um sobrevoo considerando regiões heterogêneas as quais podem ter diferentes configurações de voo e resolução de coleta de dados. No Capítulo 3 será descrito um novo método que busque suprir essa necessidade.

3 DESENVOLVIMENTO

Soluções para o planejamento de caminho de cobertura são amplamente encontradas na literatura, como os exemplos listados na Seção 2.4, nos quais cada trabalho busca acrescentar diferentes propriedades aos seus algoritmos para que se adaptem melhor aos cenários e problemas propostos por cada autor.

Pode-se notar pela Seção 2.3 que um voo em baixa altitude é preferível para obter-se mais dados, porém necessita de uma maior quantidade de recursos para se executar uma missão de sobrevoo em altitudes menores. Entretanto, se for possível definir subáreas especiais no planejamento da missão, a aeronave pode ser programada para realizar um voo mais baixo apenas nestas regiões que requerem mais dados. Dessa forma, otimizando o tempo total da missão de sobrevoo e consequentemente o consumo de energia da aeronave.

Este trabalho define e descreve a implementação de um algoritmo que atende essa solução em específico, utilizando como referência o algoritmo original de bustrofédon, descrito em Choset e Pignon (1998), e acrescentando dois tipos de regiões de voo: uma considerada normal e que deve ser usada por padrão, e outra denominada especial para o qual as características de voo devem possibilitar a coleta de mais dados relevantes por meio dos sensores. A principal característica de voo utilizada é o sobrevoo em menor altitude nas regiões especiais, refletindo em um GSD menor e na presença de um maior detalhamento nas imagens coletadas.

Para a implementação foram tomadas decisões de desenvolvimento tendo como alvo o VANT da DJI Mavic Pro (DJI, 2016), o qual oferece um conjunto de ferramentas de desenvolvimento para o ambiente Android/Java por meio de sua SDK (do inglês, *Software Development Kit*) (DJI, 2020). Entretanto, os algoritmos descritos a seguir são alheios a qualquer aeronave ou linguagem de programação, não tendo nenhuma dependência com bibliotecas ou tecnologias externas. A solução desenvolvida está disponível no repositório do GitHub em <https://github.com/Lechenko/h-cpp>.

3.1 ORGANIZAÇÃO DO ALGORITMO

Nessa seção são definidas as informações de entrada que devem ser passadas ao algoritmo, a estruturação das classes criadas como base para o mesmo e as etapas que constituem a solução proposta.

3.1.1 INFORMAÇÕES DE ENTRADA

Antes de detalhar o algoritmo em si, o usuário tem como responsabilidade fornecer as informações essenciais para que o mesmo possa ser executado corretamente. Tendo em vista o cenário no qual se espera que este algoritmo seja utilizado, sobre áreas rurais de diversos tamanhos e formatos, o usuário deve indicar para cada execução a área de interesse que será sobrevoada junto com as subáreas especiais que terão um sobrevoo mais próximo do solo. Tendo ainda que indicar a posição inicial da aeronave para ser possível iniciar o caminho planejado próximo ao local que a mesma se encontra.

Para isso, essas informações devem ser abstraídas para que possam ser utilizadas e processadas por um sistema digital. Portanto, ao utilizar de coordenadas do sistema de posicionamento global (GPS) obtém-se valores que indicam com precisão posições geográficas. Normalmente, é adequado fazer uma aproximação do formato da Terra para o de uma esfera perfeita, sem considerar outros fatores como mudanças de altitude e relevo da superfície (WILLIAMS, 2002), já que erros relacionados à curvatura dessa esfera só começam a surgir para distâncias maiores e não se tornam relevantes para a maioria das aplicações. Devido à pequena escala à qual será aplicada na superfície do planeta, e sendo visado a utilização em áreas de cultivo que tendem a ser planas e sem muitos obstáculos, a solução desenvolvida trata essas coordenadas como pertencentes a um plano cartesiano sem que se observe problemas de precisão relevantes nos cálculos posteriores. Dessa forma, as áreas e subáreas necessárias para a entrada podem ser representadas por meio de um polígono formado por uma lista de pontos que simbolizam as coordenadas de seus vértices seguindo o sentido horário.

Assim, como entrada o algoritmo recebe uma lista de subáreas que juntas representam uma única área de interesse. Essas subáreas são formadas por, além do polígono que a representa, uma propriedade de tipo que indica se esta é uma subárea normal ou especial. Devido à forma como a solução foi implementada, o usuário deve

se atentar em fornecer apenas uma única subárea com o tipo normal, e qualquer número de subáreas com o tipo especial, desde que estejam todas contidas dentro da subárea normal. Em conjunto com esta lista de subáreas, deve ser passado também uma única coordenada que indique a posição inicial em que o VANT se encontra. O Algoritmo 1 demonstra como tais valores de entrada podem ser estruturados em um arquivo JSON.

Algoritmo 1: Exemplo de dados de entrada em formato JSON

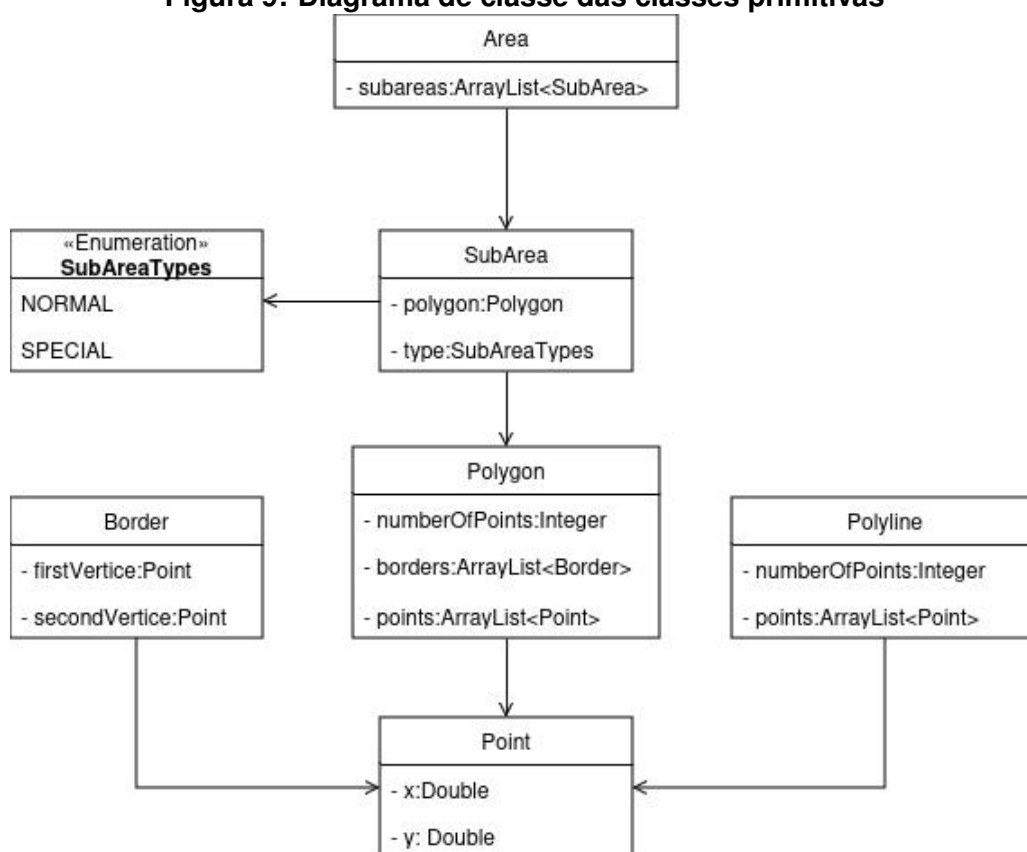
```
{ "startPosition": {  
  "latitude": -23.35465,  
  "longitude": -51.24353  
},  
"subareas": [  
  { "data": [  
    {  
      "latitude": -23.35629,  
      "longitude": -51.24368  
    },  
    {  
      "latitude": -23.3552,  
      "longitude": -51.2429  
    },  
    {  
      "latitude": -23.35511,  
      "longitude": -51.24148  
    },  
    {  
      "latitude": -23.36157,  
      "longitude": -51.24146  
    },  
    {  
      "latitude": -23.36223,  
      "longitude": -51.24367  
    }  
  ] },  
  "type": "normal"  
} ] }
```

Estas informações então são recebidas e abstraídas para um conjunto de classes primitivas que todo o algoritmo irá utilizar durante a sua execução.

3.1.2 CLASSES PRIMITIVAS

Decorrente da abstração mencionada anteriormente, um conjunto de classes primitivas foram projetadas seguindo os conceitos de programação orientada a objetos a fim de conter e organizar as informações da entrada, além de facilitar a modularização e o entendimento do código. A Figura 9 ilustra essas primitivas e como elas interagem entre si por meio de suas propriedades.

Figura 9: Diagrama de classe das classes primitivas



Fonte: Elaborado pelo autor

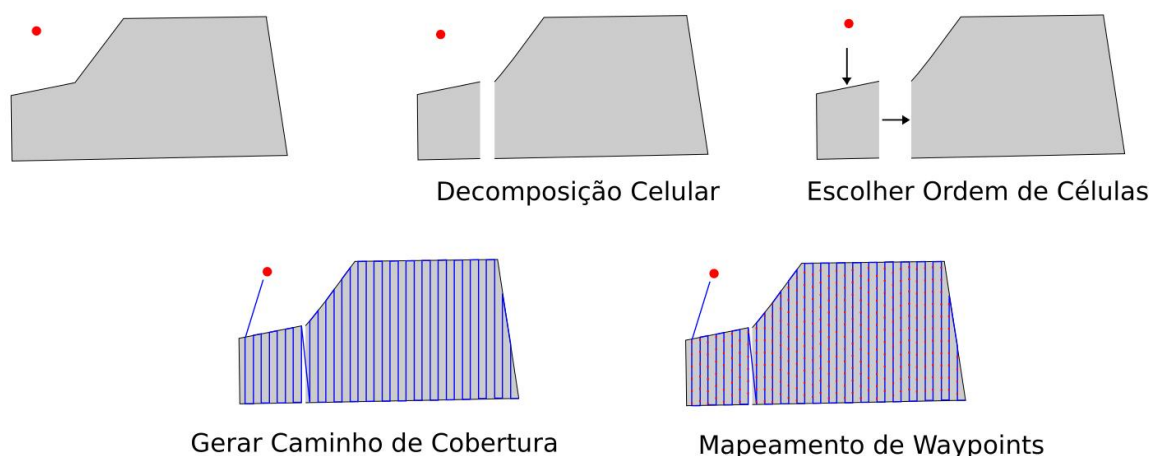
A primitiva base é representada pela classe `Point`, que armazena apenas as posições x e y do plano cartesiano. A partir desta classe, é possível construir `Border` para representar arestas e segmentos de reta, `Polygon` para representar as regiões e subregiões formadas por um grupo de pontos e arestas e a `Polyline` para uma linha formada por inúmeros pontos para formar o caminho final. Em uma terceira camada

de primitivas há a *SubArea* para acomodar o polígono e o tipo de subárea (normal ou especial) e *Area* para armazenar um conjunto de subáreas.

3.1.3 ALGORITMO DE PLANEJAMENTO DE CAMINHO DE COBERTURA HETEROGÊNEO

Uma vez com a entrada inserida corretamente e abstraída para objetos referentes às classes primitivas, o planejamento de caminho de cobertura pode ser executado sem mais intervenções do usuário. No decorrer das próximas seções será descrito de forma detalhada as etapas e decisões tomadas no desenvolvimento do algoritmo, o qual pode ser agrupado em 4 diferentes etapas (Figura 10): a decomposição celular para simplificar e organizar áreas de interesse com formatos complexos (Seção 3.2), a escolha do melhor caminho entre as células (Seção 3.3), a varredura dessas células utilizando um algoritmo de *back and forth* (Subseção 3.3.2) e o mapeamento de *waypoints* e coordenadas para repassar para o controlador de missão (Subseção 3.3.3).

Figura 10: Fluxo das etapas do algoritmo de planejamento de caminho de cobertura heterogêneo



Fonte: Elaborado pelo autor

3.2 DECOMPOSIÇÃO CELULAR

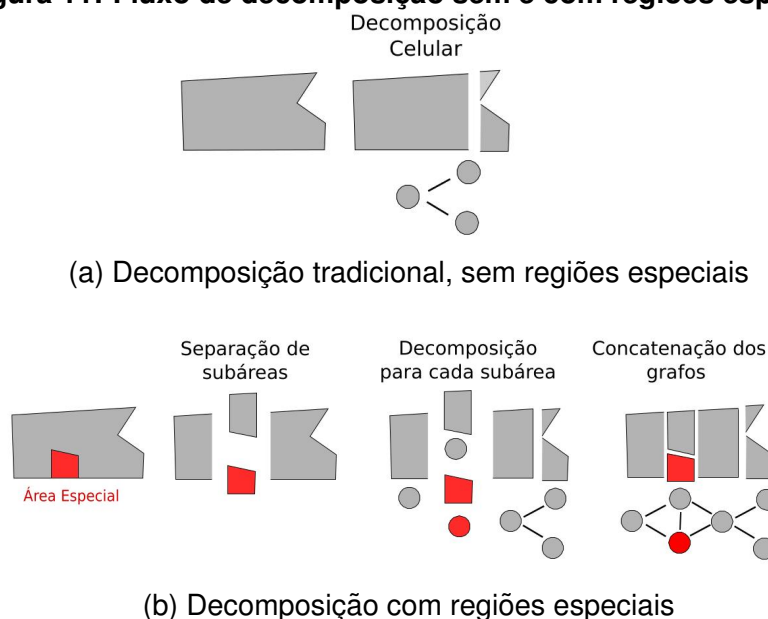
Como visto na Seção 2.2, existem diversas técnicas para realizar um planejamento de caminho de cobertura as quais são classificadas baseadas na forma como a área de interesse é subdividida em regiões menores. Essa estratégia de dividir e conquistar é crucial em um algoritmo de CPP, pois aqui ficam várias das limitações de cenários que o algoritmo estará apto a analisar e oferecer uma resposta. Este trabalho permite realizar o planejamento de áreas de interesses complexas, formadas por polígonos côncavos e convexos, podendo haver ou não subáreas especiais as quais se espera uma maior coleta de dados.

Para isso, foi desenvolvida uma técnica de Decomposição Celular Exata, similar a definida pelos algoritmos de decomposição trapezoidal e de bustrofédon, com a diferença de considerar essas subáreas especiais com diferentes propriedades na geração do caminho. Como o nome sugere, esta etapa irá recortar a área de interesse em múltiplas células para facilitar os processos posteriores. Estas células serão organizadas em um grafo representado por uma matriz de adjacência, que conecta cada uma das células com as suas respectivas vizinhas.

O procedimento e resultado esperado é ilustrado pela Figura 11, no qual o fluxo representado pela Figura 11a não difere muito do já encontrado na literatura, sendo utilizado no caso de não haver subáreas especiais. Enquanto que o fluxo da Figura 11b é acionado na presença de uma ou mais subáreas especiais, na qual é realizada uma primeira divisão das subáreas para garantir que não haja sobreposições entre elas. Cada nova subárea então é submetida ao algoritmo habitual de decomposição celular da Figura 11a, o qual retornará um grafo de células para cada uma delas. Por fim, estes grafos são concatenados entre si e formam uma representação única da área de interesse original.

3.2.1 SOBREPOSIÇÃO DE ÁREAS

Devido a como foi definida a abstração da entrada pelas classes da Subseção 3.1.2, no caso de múltiplas subáreas, é possível, e espera-se, que subáreas especiais sobreponham a mesma região de subáreas normais. Entretanto, esta sobreposição deve ser removida de imediato para evitar que o planejamento do caminho final percorra o mesmo trecho uma segunda vez.

Figura 11: Fluxo de decomposição sem e com regiões especiais

Fonte: Elaborado pelo autor

A remoção desta duplicidade ocorre em dois passos: dividir a área normal em múltiplos polígonos seguindo os vértices das subáreas, e recortar estes polígonos de forma que não se sobreponham com nenhuma outra subárea. O primeiro passo consiste em, para cada SubArea especial, compará-la com a SubArea normal preparando esta última para ser destacada em múltiplos polígonos. Para isso cada vértice da SubArea especial é analisado e projetado para as arestas da área de interesse, criando-se um novo ponto para que seja possível subdividir essa área sem perder a sua forma. Este processo ocorre varrendo os vértices da esquerda para a direita (eixo x) e os projetando ortogonalmente (eixo y), cuja decisão é arbitrária para diminuir a complexidade do desenvolvimento total.

O recorte que será realizado é baseado no algoritmo de Weiler-Antherton, descrito por Weiler e Atherton (1977), que é capaz de recortar a interseção entre dois polígonos de forma eficaz. O algoritmo recebe como entrada os pontos de dois polígonos, o polígono que se quer recortar e o polígono que forma o molde do recorte. O passo anterior consiste em garantir que as arestas dos polígonos analisados se encontrem de alguma forma, já que o algoritmo de Weiler-Antherton não considera polígonos que estejam completamente inseridos um no outro.

O algoritmo de Weiler-Antherton é descrito da seguinte maneira:

- Crie duas listas e coloque em cada uma os vértices de um dos polígonos

seguindo a ordem no sentido horário;

- Calcule os pontos de intersecção entre os dois polígonos e os classifique como ENTERING ou LEAVING, de acordo com se as arestas entram ou saem do polígono de molde quando listadas em um sentido horário;
- Adicione estes pontos nas duas listas em suas posições apropriadas, considerando a ordem dos pontos ao percorrer o perímetro do polígono em sentido horário;
- Crie uma nova lista para o polígono final e a popule com a função `mountResult` do Algoritmo 2.

Algoritmo 2: Funções para montar o resultado do algoritmo de Weiler-Antherton

```
mountResult(List pointsA, List pointsB) {
    result = List()
    start = firstIntersection(pointsB, IntersectionType)
    current = start
    do {
        current = getNextIntersection(current, pointsB, directionB, result)
        current = getNextIntersection(current, pointsA, directionA, result)
    } while(current != start)
    return result
}

getNextIntersection(Intersection current, List points, int direction,
List resultPoints) {
    if (!points.contains(current)) return current
    Intersection aux = current
    do {
        resultPoints.add(aux.point)
        aux = getNextPoint(aux, points, direction)
    } while (aux.status == IntersectionStatus.NONE)
    return aux
}
```

Com o retorno da função `mountResult` têm-se os pontos que formam o polígono de intersecção entre `pointsA` e `pointsB`. Sendo este encontrado com a

iteração dos elementos das duas listas, alternando de uma para a outra quando uma interseção é encontrada e até se encontrar o ponto inicial, fechando então o circuito. Entretanto, ao adicionar um pequeno ajuste no parâmetro *direction*, alterando a direção em que se buscará os pontos nas listas e qual o tipo de interseção inicial, é possível alterar o comportamento do algoritmo para realizar outras operações como a diferença e a junção dos dois polígonos. Para o caso específico deste trabalho, a diferença é mais atrativa para que se possa remover a duplicidade entre as subáreas. O recorte então é feito e resulta em um novo conjunto de *SubArea* sem qualquer duplicação (Figura 12). Cada uma destas segue para uma etapa de decomposição e criação das células.

Figura 12: Exemplo de recorte de polígono



(a) Polígono que será cortado e molde do corte

(b) Polígono cortado

Fonte: Elaborado pelo autor

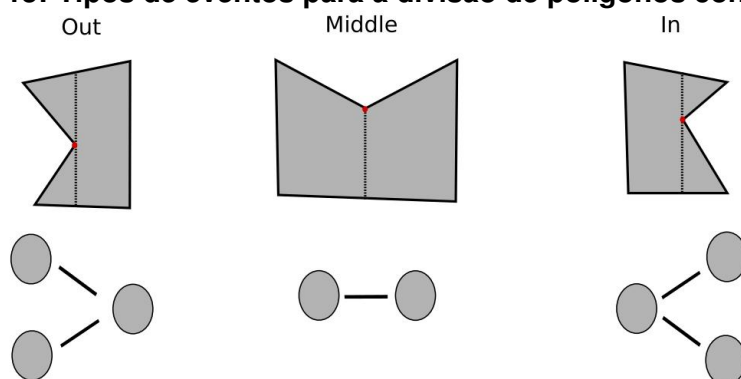
3.2.2 DECOMPOSIÇÃO CELULAR DAS SUBÁREAS

Para evitar o aumento da complexidade durante a geração de um caminho que realize a cobertura completa da área de interesse, uma divisão celular é necessária para evitar possíveis áreas delimitadas por polígonos côncavos. O algoritmo descrito a seguir é inspirado na decomposição trapezoidal e de bustrofédon com o intuito de dividir cada *SubArea* em regiões menos complexas se necessário, produzindo então um conjunto de *Cell* organizadas em uma estrutura de grafo.

Para dividir a região referente a uma *SubArea*, a qual é representada por um objeto da classe *Polygon*, no caso deste ser um polígono côncavo, mapeia-se cada vértice e verifica-se quais pontos são críticos para a redução desta complexidade. Em outras palavras, ao analisar a geometria do polígono em questão, é possível definir em quais vértices uma nova aresta deve ser traçada, separando-o em novos polígonos

convexos menores. Essa análise é armazenada em um objeto `CriticalPoint` para cada vértice, envolvendo o `Point` em questão junto com uma classificação de eventos cruciais para a divisão correta do polígono: IN, OUT, MIDDLE e NONE. Cada um desses eventos é ilustrado na Figura 13.

Figura 13: Tipos de eventos para a divisão de polígonos côncavos



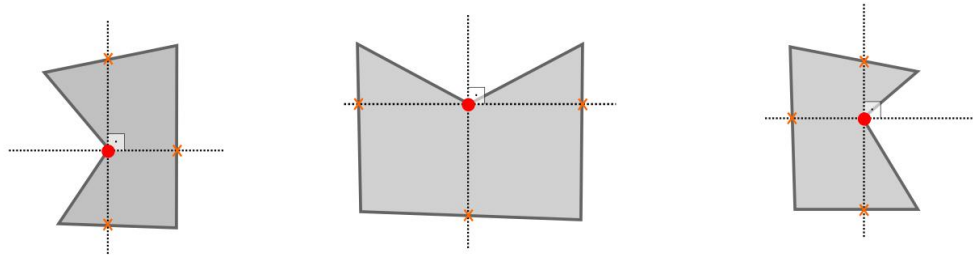
Fonte: Elaborado pelo autor

Estes eventos são atribuídos ao examinar o polígono tendo um ângulo de corte θ como referência. Aqui novamente optou-se em um primeiro momento por utilizar o ângulo fixo de $\theta = \pi/2$ para realizar os cortes, pois dessa forma pode-se validar a contribuição do trabalho em realizar planejamento heterogêneo e generalizar para outros cenários posteriormente. A partir de cada vértice é traçado dois segmentos de retas perpendiculares entre si, com ângulo de inclinação de $\alpha = \theta$ e $\beta = \theta + \pi/2$. A Figura 14 ilustra esse processo de preparação, na qual as linhas tracejadas representam os segmentos de reta perpendiculares, a partir dos quais é possível determinar as marcações em laranja, localizando os pontos de interseções entre os segmentos e as arestas do polígono.

Ao inspecionar qualquer polígono de entrada convexo dessa forma, constata-se como trivial que esta análise irá gerar apenas um único ponto de interseção para cada um dos segmentos de reta. Consequentemente, ao verificar a quantidade dos pontos de interseções pode-se aferir quais vértices estão provocando a concavidade e onde deve ser realizado a separação. Neste caso, um novo `CriticalPoint` é criado nos pontos de interseção obtidos pelo segmento de reta em α para facilitar o aparo do polígono naquela posição.

Uma vez identificado o vértice que provoca a concavidade, é necessário determinar se este apresenta um evento de IN, OUT ou MIDDLE. O que é possível ao analisar, no caso do MIDDLE, em qual segmento de reta foram encontradas as

Figura 14: Segmentos de retas perpendiculares e interseções a fim de encontrar concavidades



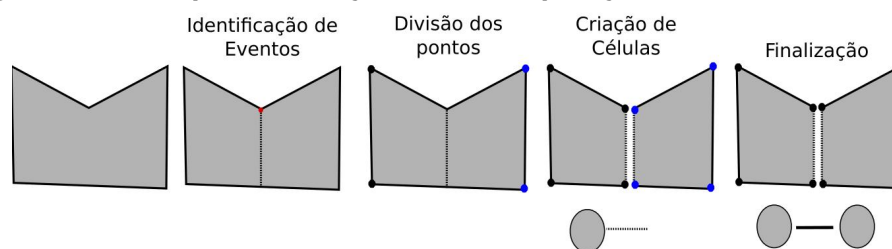
Fonte: Elaborado pelo autor

interseções, pois só se encontrará uma interseção no segmento de reta com ângulo α e múltiplas no segmento com ângulo β . Para os casos IN e OUT é necessário avaliar o ângulo entre as arestas que estão conectando este vértice, a fim de determinar se serão geradas duas novas células a partir daquele ponto ou se compreende o encontro de duas células em uma única nova (Figura 13).

A verificação de cada vértice é realizada ordenando os mesmos por sua posição referente ao ângulo β . Assim, para um ponto p_i , o qual foi definido como um ponto crítico e de ruptura, um novo polígono pode ser formado pelos vértices que cumprirem a condição $p_n \cos(\beta) \leq p_i \cos(\beta)$. Contudo, um algoritmo que caminhe entre as arestas também é verificado para evitar a criação de polígonos incompletos e que possam contradizer a área original. Uma vez gerado, uma nova *SubArea* é gerada e adicionada dentro de uma célula *Cell*, que por sua vez é adicionada na matriz de adjacência *AdjacencyMatrix* em um nó conectado às outras células que compartilham uma aresta em comum. Por fim, os pontos nos quais $p_n \cos(\beta) < p_i \cos(\beta)$ são removidos da iteração para evitar a geração de duplicidade e a execução continua até que todos os pontos sejam verificados com sucesso. Note que para os pontos cuja a condição $p_n \cos(\beta) = p_i \cos(\beta)$ é verdadeira não são removidos pois estes formam a aresta que conecta esta célula com a que será gerada na próxima iteração, sendo mantidos dessa forma para que esta adjacência permaneça. A Figura 15 demonstra cada etapa do processo de decomposição celular e montagem do polígono descrita.

A saída final desta etapa está totalmente contida no grafo representado pela matriz de adjacência gerada. No caso do processo anterior descrito pela Subseção 3.2.1 ter fornecido múltiplas *SubArea*, a decomposição ocorre isoladamente, gerando um grafo para cada uma delas. Posteriormente, estes são concatenados

Figura 15: Exemplo de iteração da decomposição de um evento MIDDLE



Fonte: Elaborado pelo autor

em uma única *AdjacencyMatrix* com todas as células produzidas para que se possa definir a disposição destas durante a elaboração do caminho de cobertura.

3.3 GERAÇÃO DO MELHOR CAMINHO

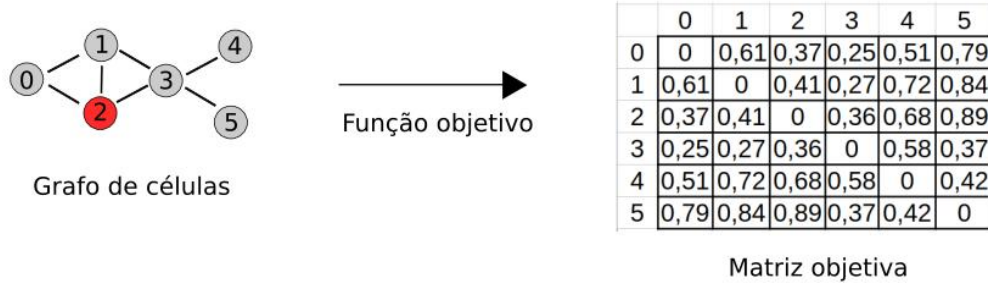
Com a decomposição celular finalizada obtém-se uma *AdjacencyMatrix* contendo as *Cell* que juntas formam a área de interesse fornecida inicialmente. Pode-se então iniciar a geração do caminho de cobertura final. Entretanto, antes deve-se definir a ordem a qual as células serão visitadas pelo VANT de forma a evitar transições muito longas entre diferentes células que estão distantes uma da outra.

3.3.1 ORDEM DE VISITA DAS CÉLULAS

A escolha da ordem de visita das células não é trivial, visto que pode ser impactada por fatores como o número de células vizinhas, o tamanho de cada célula e a distância entre elas. Portanto, para se tomar essa decisão deve-se utilizar as informações processadas até aqui, por meio de uma função objetivo para encontrar o custo de uma célula para outra. Essa função leva em consideração cada par de células calculando um valor para cada um destes, os quais são armazenados no formato de uma matriz denominada matriz objetiva (Figura 16), a qual contém todos os possíveis custos de transição entre cada célula.

Para se adquirir esses valores, foi elaborada uma função objetivo simples que busca quantificar a distância entre os centros das células, sugerindo que a escolha seja feita dando prioridades para as células vizinhas e menores. Essa função foi escolhida de forma arbitrária para que se possa demonstrar a metodologia definida, sendo de interesse de trabalhos futuros explorar outras formas de quantificar

Figura 16: Aplicação de uma função objetiva para todos os pares de células presentes em um grafo, gerando uma matriz objetiva



Fonte: Elaborado pelo autor

a transição entre os objetos obtidos pela decomposição celular. Dessa forma, ao considerar duas células C_1 e C_2 , cada uma sendo delimitada por um polígono formado por um conjunto de coordenadas $P = \{(x_1, y_1), (x_2, y_2), \dots, (x_n, y_n)\}$, pode-se calcular o ponto médio central desses polígonos pela Equação (3).

$$\begin{aligned} x_{cm} &= \frac{1}{N} \sum_{k=1}^N x_k, \\ y_{cm} &= \frac{1}{N} \sum_{k=1}^N y_k \end{aligned} \quad (3)$$

Uma vez obtidos, calcula-se a distância euclidiana entre esses dois pontos médios. A Equação (4) descreve a função objetivo utilizada para calcular a matriz de custos, considerando que P_1 e P_2 sejam os polígonos das células C_1 e C_2 , com ponto médio centrais definidos por $(x_{cm,1}, y_{cm,1})$ e $(x_{cm,2}, y_{cm,2})$, respectivamente.

$$F_{obj}(P_1, P_2) = \sqrt{(x_{cm,1} - x_{cm,2})^2 + (y_{cm,1} - y_{cm,2})^2} \quad (4)$$

Essa nova matriz de custos é inserida em um algoritmo para definir a ordem final das células. Partindo da célula mais próxima da posição inicial, o índice da próxima célula é escolhido considerando o menor custo disponível na matriz, desconsiderando as células que já foram selecionadas em iterações anteriores a fim de evitar que uma célula seja visitada múltiplas vezes. Essa é uma estratégia gulosa para se encontrar uma sequência subótima da passagem entre as células, sendo esta a última etapa necessária antes de se calcular o caminho de cobertura.

3.3.2 CAMINHO DE COBERTURA

Com a decomposição celular em polígonos convexos (Seção 3.2), a remoção das áreas sobrepostas (Subseção 3.2.1), e a ordem de visita das células determinada (Seção 3.3), pode-se finalmente construir o caminho de cobertura com um algoritmo de *back and forth*. Este algoritmo varre cada polígono de uma extremidade à outra com um espaçamento fixo entre as trilhas. Este espaçamento leva em consideração o cálculo do GSD que afere o comprimento referente a cada pixel da imagem naquela altitude e o número de pixels que a câmera é capaz de capturar. Combinando essas duas informações, se tem a área de cobertura, em metros, que é possível cobrir pela câmera nessa altitude e, portanto, o espaçamento máximo entre as trilhas para não haver lacunas entre as imagens coletadas. Como a altitude alvo varia entre os tipos de subáreas, a distância entre as trilhas também irá variar entre as células que representam uma região normal e as que representam uma região especial.

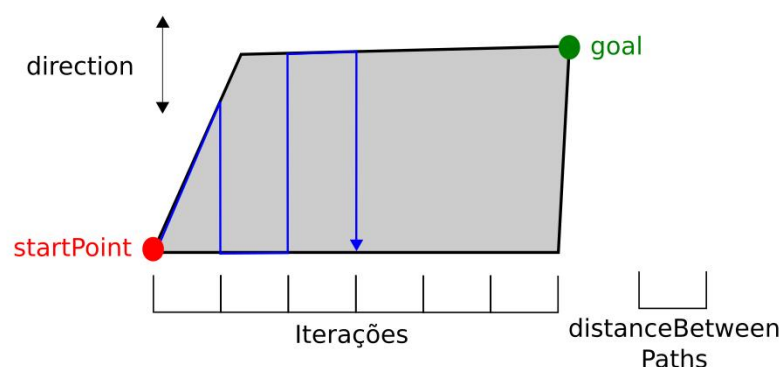
Seguindo a ordem de visita das células pela lista ordenada resultante da Subseção 3.3.1, para cada célula, é criada uma lista de pontos denominada `path` que representa o caminho sendo gerado. O primeiro ponto é chamado de `initialPoint`, estando fora da célula, sendo no caso da primeira célula o ponto inicial definido pela entrada e para as demais o último ponto do `path` da célula anterior. A partir desse primeiro ponto, os próximos pontos que geram o caminho de cobertura de cada célula são encontrados em sequência.

Para isso, primeiro, deve-se definir os seguintes parâmetros, que também são exemplificados na Figura 17:

- Define o espaço entre as trilhas em `distanceBetweenPaths`;
- Define o ângulo de direção em que a varredura ocorrerá como `direction`;
- Selecione as arestas que não seguem essa direção e a defina em uma lista como `walls`;
- Encontre o vértice mais próximo do `initialPoint` e defina como `startPoint`;
- Define o ponto atual com o valor de `startPoint` e o armazena em `currentPoint`;
- Encontre o vértice mais distante do `startPoint` possível e o defina como `goal`;

- Calcule o número de iterações necessárias para varrer o polígono todo de `startPoint` até `goal`, considerando `distanceBetweenPaths` e o ângulo ortogonal a `direction`, e defina como `numberOfIterations`.

Figura 17: Ilustração dos parâmetros utilizados durante a geração de caminho



Fonte: Elaborado pelo autor

Uma vez com todos os parâmetros definidos, a geração do caminho é realizada por iterações que encontram cada uma das trilhas entre o `startPoint` e o `goal` seguindo um padrão de *back and forth* com o ângulo definido em `direction`. Cada iteração se concentra nas seguintes etapas:

- Valide se `currentPoint` existe e se é diferente do último ponto de `path`;
- Adicione `currentPoint` ao `path`;
- Valide se `currentPoint` está contido dentro das arestas do polígono;
- Calcule uma interseção válida com uma aresta presente em `walls` ao traçar uma linha seguindo `direction`;
- Defina o ponto encontrado como `currentPoint`;
- Defina a aresta em que o ponto foi encontrado como `currentWall`;
- Adicione `currentPoint` ao `path`;
- Calcule o ponto em direção à próxima trilha com `distanceBetweenPaths`, seguindo a direção do ângulo de `currentWall` no sentido que se aproxima de `goal`;

- Define o ponto encontrado como `currentPoint`;

O caminho de cada célula é abstraído em uma *Polyline* que contém uma lista de *Point* como atributo. Por fim, a transição entre células é dada pela conexão entre o último ponto do *path* da célula anterior com o primeiro ponto do *path* da célula seguinte.

3.3.3 MAPEAMENTO DE WAYPOINTS

Com todas as células visitadas pela etapa de geração de caminho, pode-se repassar para a aeronave uma lista de coordenadas nas quais cada imagem deverá ser capturada. O espaçamento entre as fotografias considera novamente o valor de GSD da altitude definidos para cada subárea, similar ao cálculo feito para encontrar o espaçamento entre as trilhas, para que a câmera faça a cobertura total da área de interesse sem que existam lacunas ou sobreposições entre as imagens. Dessa forma, subáreas especiais em um sobrevoo mais baixo conseguem projetar o sensor da câmera em uma área menor e são necessários mais *waypoints* e fotografias coletadas.

3.4 EXPERIMENTOS

Para facilitar simulações e a visualização de experimentos ao utilizar a solução proposta, foi construída uma aplicação em um ambiente Android/Java com uma interface integrada ao serviço de GPS oferecido pela SDK do *Google Maps* (GOOGLE, 2023). Essa interface ilustra as áreas de interesse fornecidas como entrada do algoritmo, sua decomposição celular, e o caminho de cobertura encontrado ao final da execução.

A visualização das etapas e resultados é importante tanto durante o desenvolvimento da aplicação na identificação de problemas de comportamento quanto na validação da saída e do caminho gerado mesmo antes de realizar um único voo com a aeronave. Dessa forma, as imagens obtidas por essa aplicação serão utilizadas durante o Capítulo 4 para auxiliar na descrição dos resultados e se estes alcançaram os comportamentos esperados.

Além de uma validação qualitativa da solução de planejamento de caminho de cobertura por meio destas imagens, também foi realizada uma validação quantitativa de métricas referente à execução da aplicação para possibilitar uma comparação em

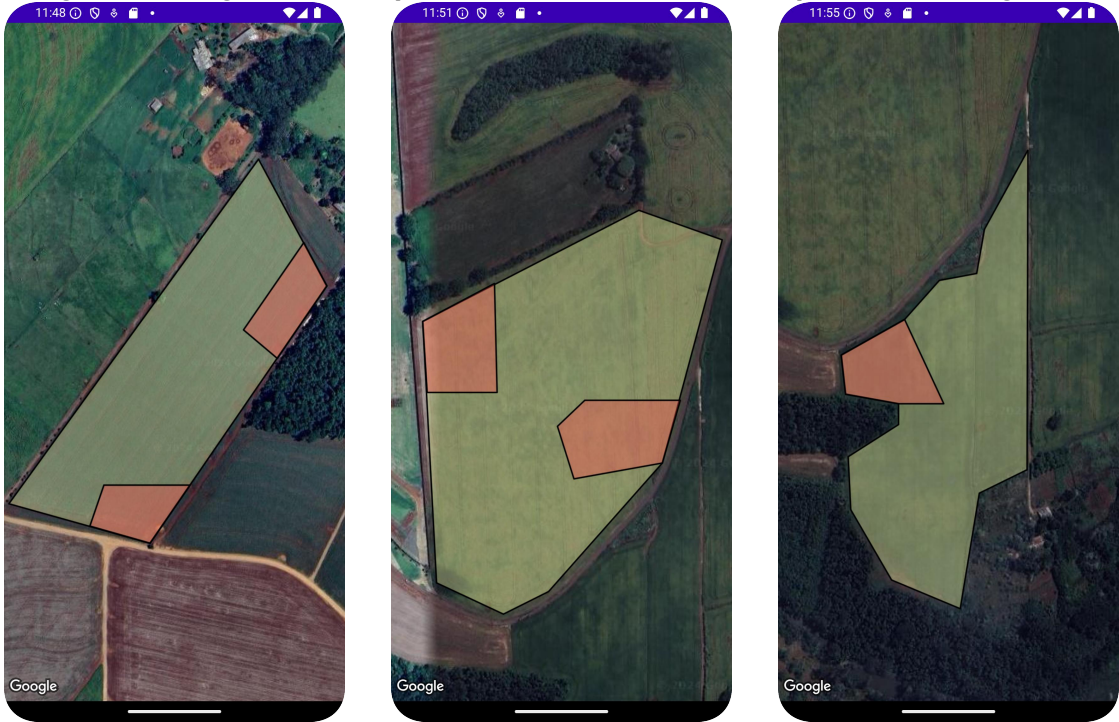
diferentes cenários. São duas as principais métricas: tempo de execução do algoritmo, analisando como o planejamento de caminho de cobertura heterogêneo impacta o tempo de preparação de uma missão da aeronave; e o comprimento total do caminho gerado para que se possa comparar execuções entre a solução clássica homogênea e a heterogênea proposta. Estas métricas serão coletadas ao executar o código referente ao método descrito neste trabalho em uma JVM (*Java Virtual Machine*) em um sistema operacional Linux. Devido à preemptividade intrínseca dos sistemas operacionais modernos, aferir o tempo de execução de um algoritmo de forma isolada é arquiteturalmente impossível. Portanto, no caso da métrica temporal, os números serão obtidos por meio da média e desvio padrão de 30 execuções para cada um dos cenários.

São ao todo 9 experimentos distribuídos entre as 3 regiões presentes na Figura 18, sendo para cada uma das áreas delimitadas 3 cenários: primeiro a região inteira em uma altitude alta (40 m), depois novamente a região como um todo, mas com um sobrevoo mais baixo (20 m), e por fim um cenário híbrido, com toda a região em uma altitude alta (40 m) exceto algumas subáreas especiais no qual será planejado um voo em altitude menor (20 m). A Figura 19 ilustra esse último cenário para cada uma das regiões, com as subáreas especiais preenchidas pela cor vermelha.



Fonte: Elaborado pelo autor

Figura 19: Regiões dos experimentos com subáreas especiais, heterogêneas



Fonte: Elaborado pelo autor

A Tabela 2 sintetiza todas as informações sobre os experimentos e seus cenários. Os resultados e discussões sobre cada um deles estão descritos no Capítulo 4.

Tabela 2: Cenários dos experimentos

Região	Cenário	Altitude (m)	Número de subáreas especiais	Métricas qualitativas	Métricas quantitativas
1	Alta altitude	40	0	Imagens com resultado final	Comprimento do caminho e tempo de execução
1	Baixa altitude	20	0		
1	Híbrido	40/20	2		
2	Alta altitude	40	0		
2	Baixa altitude	20	0		
2	Híbrido	40/20	2		
3	Alta altitude	40	0		
3	Baixa altitude	20	0		
3	Híbrido	40/20	1		

Fonte: Elaborado pelo autor

4 RESULTADOS

Uma vez implementada a solução descrita no Capítulo 3 e executados os experimentos da Seção 3.4, é analisado neste capítulo os resultados de tais cenários a fim de validar se o objetivo proposto pelo trabalho foi alcançado e compreender as vantagens e desvantagens do método descrito.

4.1 REGIÃO 1

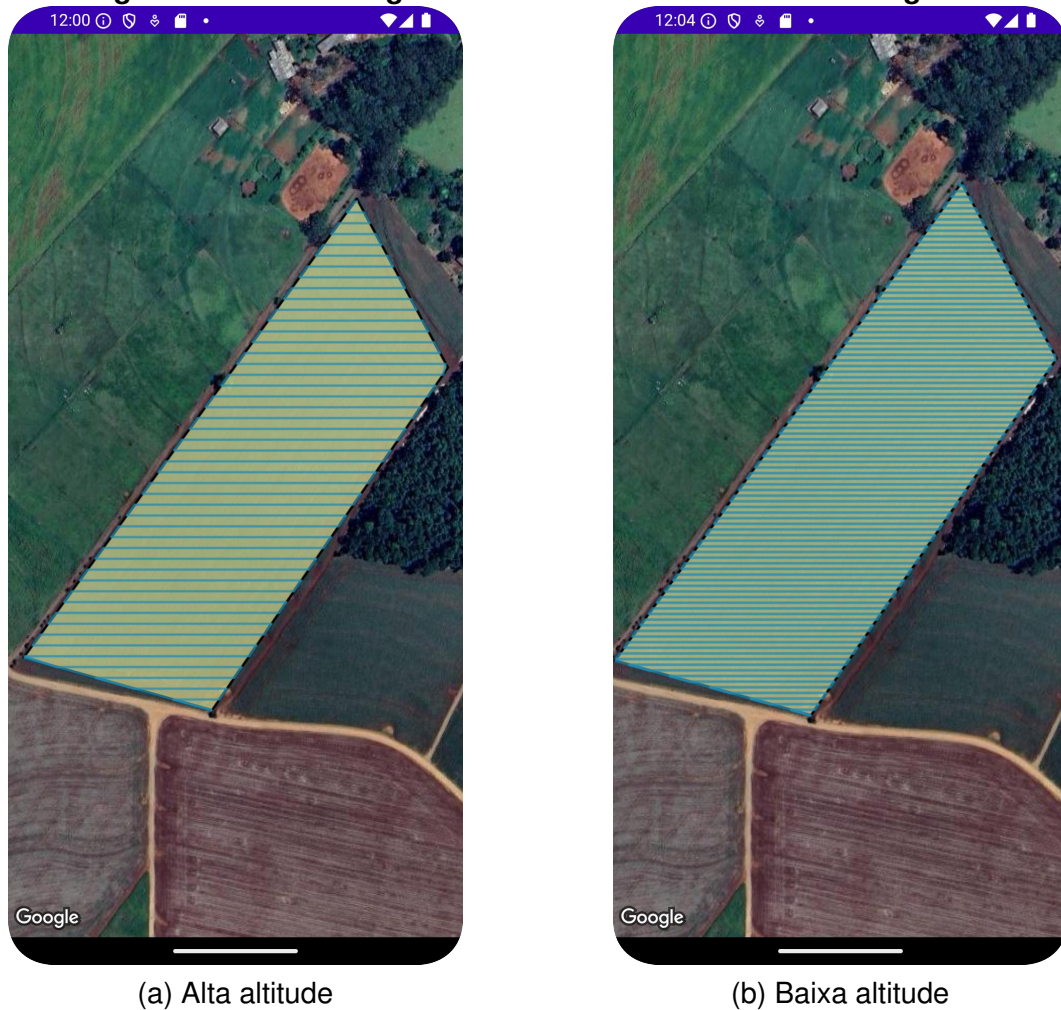
A primeira região consiste em uma superfície delimitada por um polígono de formato trapezoidal e tem como objetivo verificar o comportamento básico do algoritmo em um contexto simples. Dessa forma, para essa entrada serão executados 3 cenários distintos: o planejamento de caminho em alta altitude (40 m), o planejamento de caminho em baixa altitude (20 m) e um planejamento de caminho híbrido, sendo em alta altitude com apenas algumas áreas especiais em baixa altitude.

Como se pode perceber pela Figura 20, o algoritmo teve o resultado esperado ao gerar caminhos que cobrem completamente a superfície delimitada, com a única diferença na altitude esperada. Dessa forma, é fácil de se observar que em um sobrevoo mais alto as trilhas se encontram mais distantes uma das outras do que em um sobrevoo mais próximo do solo, indicados pela Figura 20a e Figura 20b respectivamente.

A mudança desse parâmetro impacta diretamente nas métricas quantitativas que foram observadas e anotadas na Tabela 3, praticamente dobrando o comprimento do caminho final gerado e aumentando também a duração média da execução. O comportamento do comprimento do caminho final teve o resultado esperado, pois, ao diminuir pela metade a altitude e aproximando as trilhas entre si, se vê necessário um número consideravelmente maior de trilhas para possibilitar a cobertura da mesma superfície. Estendendo a missão da aeronave e necessitando de uma capacidade consideravelmente maior de energia e armazenamento para ser concluída.

Entretanto, é esperado também um tempo de execução levemente maior para

Figura 20: Caminhos gerados em alta e baixa altitude na região 1



(a) Alta altitude

(b) Baixa altitude

Fonte: Elaborado pelo autor

Tabela 3: Métricas dos experimentos da região 1

Cenário	Comprimento do caminho final (m)	Duração total (ms)
Alta altitude	9702,161	$0,485 \pm 0,587$
Baixa altitude	18448,537	$1,363 \pm 5,553$
Híbrido	11920,543	$2,646 \pm 1,475$

Fonte: Elaborado pelo autor

esse cenário em baixa altitude devido ao aumento no número de iterações para resultar no caminho final. Na Tabela 3 nota-se um aumento de 181% ao comparar com o cenário anterior em alta altitude, demonstrando que a mudança de altitude também tem impacto na etapa de planejamento da missão.

Da mesma forma, percebe-se um alto valor do desvio padrão encontrado

na segunda linha da Tabela 3, o qual pode ser justificado devido à preempção do sistema operacional ter ocasionado algumas execuções com valores discrepantes devido à execução de inúmeros processos de forma concorrente. Esta situação será observada novamente em outros experimentos, e uma discussão mais detalhada será apresentada no final deste capítulo.

O último cenário para a região 1 consiste em adicionar duas subáreas especiais a fim de possibilitar um sobrevoo em altitude menor apenas nestas subáreas predefinidas e economizar recursos durante a missão da aeronave. A Figura 21 expõe os resultados em duas etapas distintas, sendo a Figura 21a uma organização das células geradas pela remoção de sobreposições entre subáreas (Subseção 3.2.1) e pela decomposição celular de cada nova subárea (Subseção 3.2.2), e a Figura 21b o caminho final gerado visitando e cobrindo todas as células.

Figura 21: Caminho gerado com subáreas especiais na região 1



(a) Decomposição celular



(b) Resultado final

Fonte: Elaborado pelo autor

Ao analisar a Figura 21a nota-se que o método para remoção de sobreposições ocorreu sem falhas, evitando a possibilidade da aeronave cobrir a mesma área múltiplas vezes. A decomposição celular das subáreas também não resultou em nenhum problema, apesar que as 3 células do topo esquerdo da região possam ser combinadas em uma única célula sem impactar o formato ou complexidade do polígono para etapas futuras. Essa é uma das principais contribuições encontradas no algoritmo de bustrofédon de Choset e Pignon (1998) e vantagens sobre o algoritmo anterior de decomposição trapezoidal, mas que não foi priorizada durante o desenvolvimento deste trabalho por aumentar sua complexidade sem impactar diretamente no contexto proposto de planejamento de cobertura heterogêneo, restando então como melhoria para futuros trabalhos.

O caminho gerado e traçado na Figura 21b mostra a cobertura de todas as células com exceção de um único detalhe na célula especial do topo direito. Esse erro se deve a uma falha na tomada de decisão do algoritmo de geração de caminho ao mudar entre duas arestas do mesmo polígono, pulando um espaço a mais antes de transitar para o lado oposto. O erro ocorre em uma situação específica, já que formas similares podem ser encontradas por toda a imagem em que o problema não ocorreu e, agora que identificado, pode ser depurado e resolvido em versões futuras.

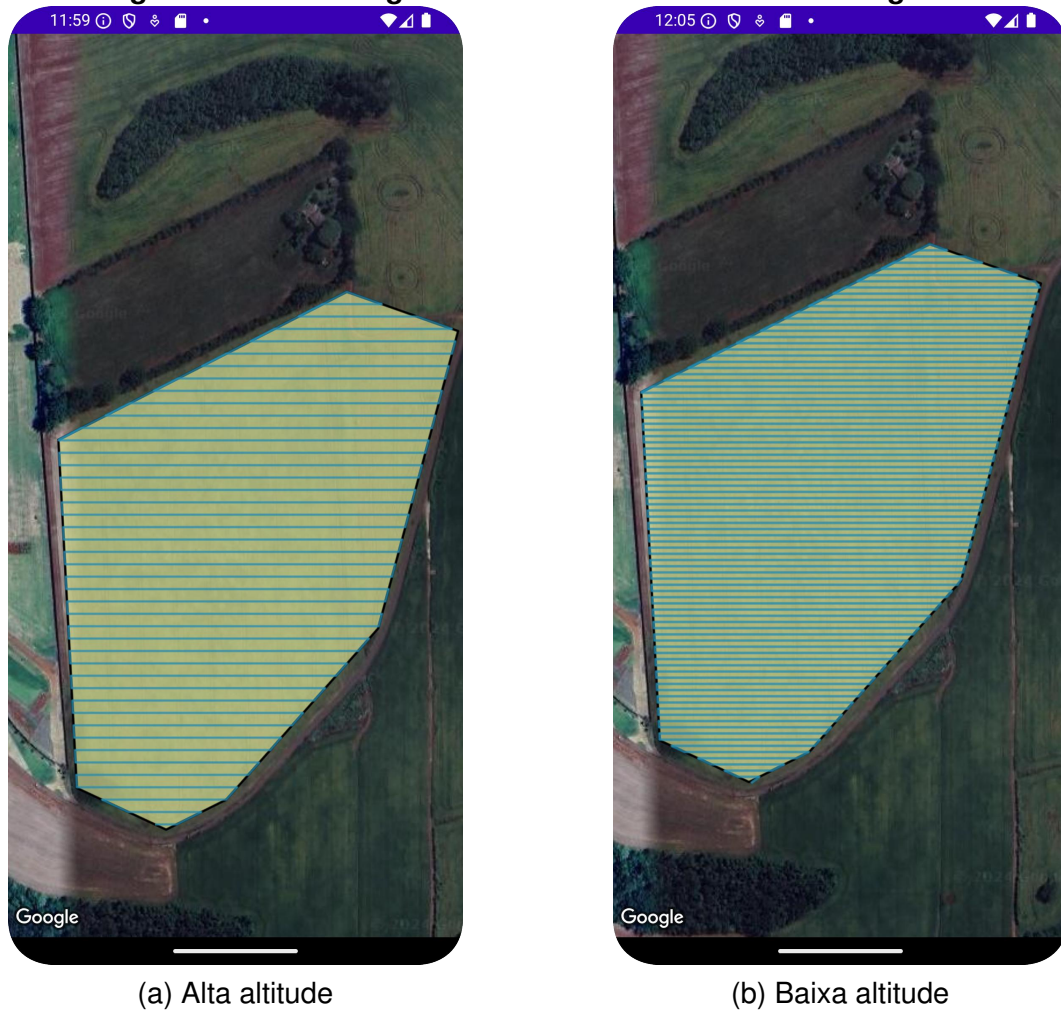
Outro ponto a se salientar, agora que existem células variadas em um mesmo planejamento, é o cálculo da ordem entre as células por meio da função objetivo (Seção 3.3). No caso da região Figura 21b, dificilmente se nota a transição do caminho de uma célula para a outra e não se encontra saltos entre células que não estão próximas, podendo inferir que a estratégia escolhida previamente para avaliar e definir as ordens das células foi bem sucedida neste cenário.

Na Tabela 3 para o cenário híbrido, há um comprimento de caminho intermediário, sendo apenas 22,8% maior que o do cenário de alta altitude em comparação com os 90,1% do cenário de baixa altitude, salientando a vantagem de caminhos de cobertura heterogêneos em uma coleta de dados com diferentes resoluções a fim de otimizar os recursos. O tempo de execução acarretou um tempo consideravelmente maior que os outros dois cenários, mesmo com um desvio padrão relativamente menor, indicando que as etapas adicionadas no pré-processamento das subáreas e a iteração entre um número maior de células impacta na execução total. Entretanto, esse impacto já era esperado devido ao alto número de laços de repetição e verificações presentes nessas etapas.

4.2 REGIÃO 2

A superfície da região 2 tem um formato menos convencional, mas que ainda é um polígono convexo e, portanto, seus resultados devem ser similares aos da região anterior. Como pode se observar pelas Figura 22a e Figura 22b, novamente não houve subdivisões entre as células e a geração do caminho de cobertura ocorreu sem nenhum problema.

Figura 22: Caminho gerado em alta e baixa altitude na região 2



Fonte: Elaborado pelo autor

A Tabela 4 apresenta as métricas dos cenários para essa região. Nela é possível notar um aumento no comprimento do caminho final de aproximadamente 91% do primeiro para o segundo cenário, similar à proporção encontrada na região 1 de 90,1%. Também há uma duração total média para o cenário de alta altitude levemente maior em consideração a da Tabela 3 devido, possivelmente, ao maior

número de vértices e a uma maior área para ser coberta. Entretanto, a comparação com os valores de tempo de baixa altitude mostra que os aferidos na região 2 (0,817) são menores do que no polígono mais simples da região 1 (1,363). Este é mais um indício para a interferência da preempção do sistema operacional durante a execução dos experimentos.

Tabela 4: Métricas dos experimentos da região 2

Cenário	Comprimento do caminho final (m)	Duração total (ms)
Alta altitude	11623,801	$0,654 \pm 2,762$
Baixa altitude	22245,648	$0,817 \pm 3,948$
Híbrido	15379,271	$3,866 \pm 7,403$

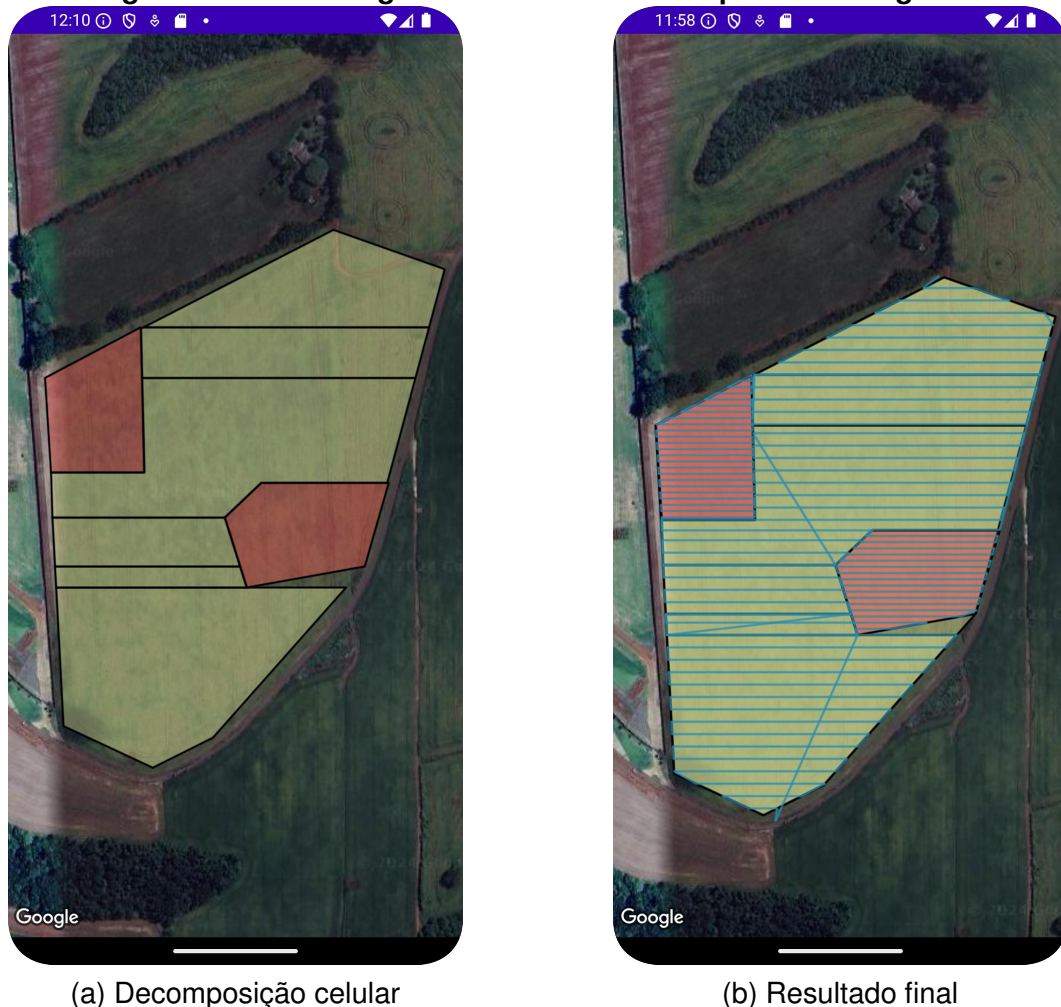
Fonte: Elaborado pelo autor

Para o cenário híbrido foram adicionadas duas subáreas especiais em extremidades opostas da região. Nota-se pela Figura 23a que as células resultantes da decomposição celular apresentam uma falha logo abaixo da subárea especial localizada à direita da imagem. Essa falha ocorreu durante a divisão da área de interesse em múltiplas subáreas normais a fim de remover as sobreposições com áreas especiais (Subseção 3.2.1), pois o algoritmo de Weiler-Antherton não prevê como resultado múltiplos polígonos ou polígonos com autoinserções. Portanto, a implementação do mesmo deve ser adaptada futuramente para evitar a perda de área de cobertura. Se observa que novamente algumas células podem ser combinadas, reduzindo o número de iterações e transições durante a geração de caminho.

O caminho gerado percorre todas as células delimitadas pela Figura 23a e pode ser visualizado na Figura 23b. É possível notar a cobertura total de todas as células, junto com as transições entre as células, onde o último ponto no caminho dentro de uma célula e o ponto inicial da próxima célula se encontram em sentidos opostos. Essas conexões quando não ideais podem acabar aumentando o custo final da missão, com seu resultado podendo ser melhorado por meio de outras funções objetivos que levem em consideração outros parâmetros a fim de otimizar o caminho entre as células.

Para este cenário com subáreas especiais, a Tabela 4 indica um comportamento similar ao encontrado na primeira área de interesse. Um comprimento do caminho final com um valor intermediário ao comparado com os outros dois cenários e a duração total da execução consideravelmente maior por conta dos

Figura 23: Caminho gerado com subáreas especiais na região 2



(a) Decomposição celular

(b) Resultado final

Fonte: Elaborado pelo autor

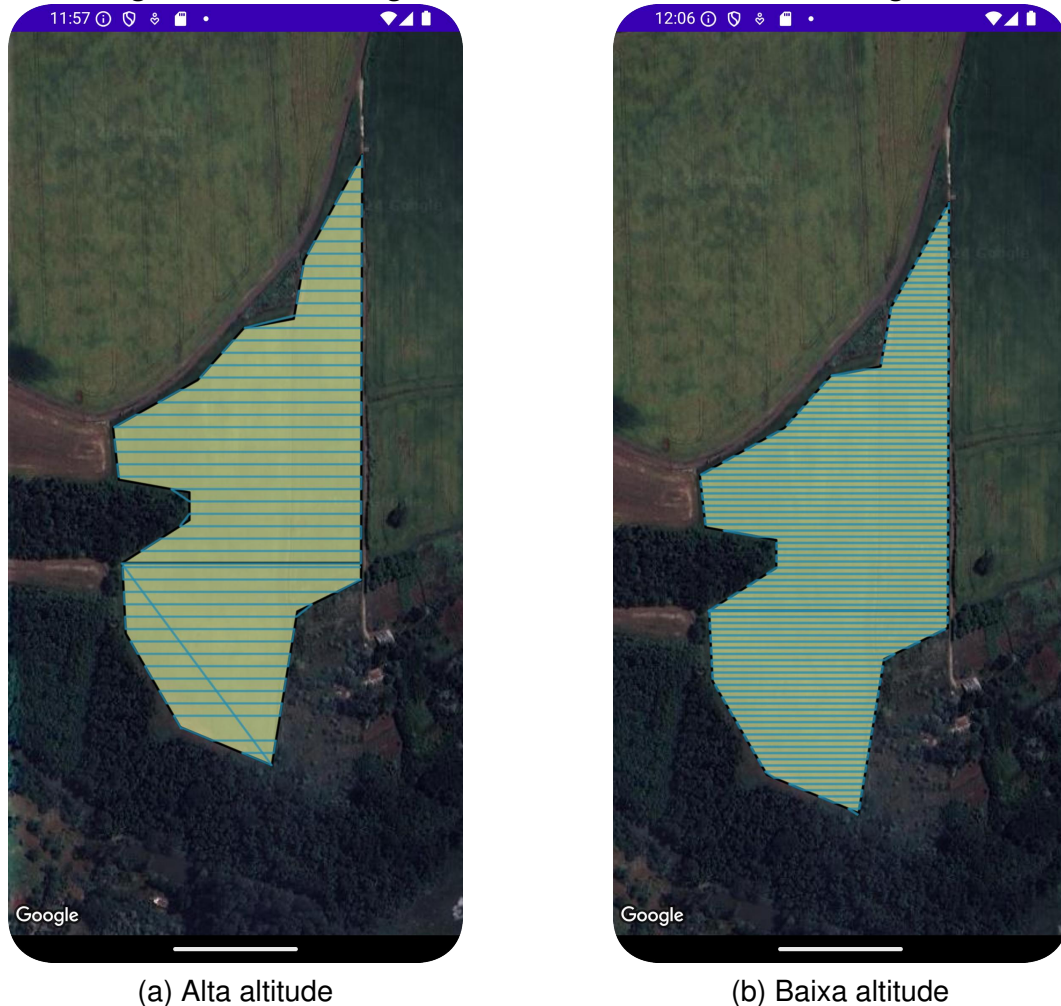
múltiplos processos e iterações adicionadas.

4.3 REGIÃO 3

A área de interesse da região 3 é a de um polígono côncavo, sendo assim um contexto mais complexo para validar o método proposto. Apesar dos caminhos de cobertura demonstrado na Figura 24a e na Figura 24b estarem corretos, a decomposição celular resultou em apenas duas células as quais não removem todas as concavidades presentes na região. O que levanta duas questões, que a implementação atual de identificação de eventos e pontos críticos deve ser revisitada para identificar formas de aperfeiçoar esse reconhecimento; e se de fato a decomposição deve ocorrer com a finalidade de remover toda e qualquer concavidade,

mesmo que estas não afetem a legitimidade do resultado ou apenas aquelas que causem transtornos durante os processos seguintes, e como diferenciar estes dois casos.

Figura 24: Caminho gerado em alta e baixa altitude na região 3

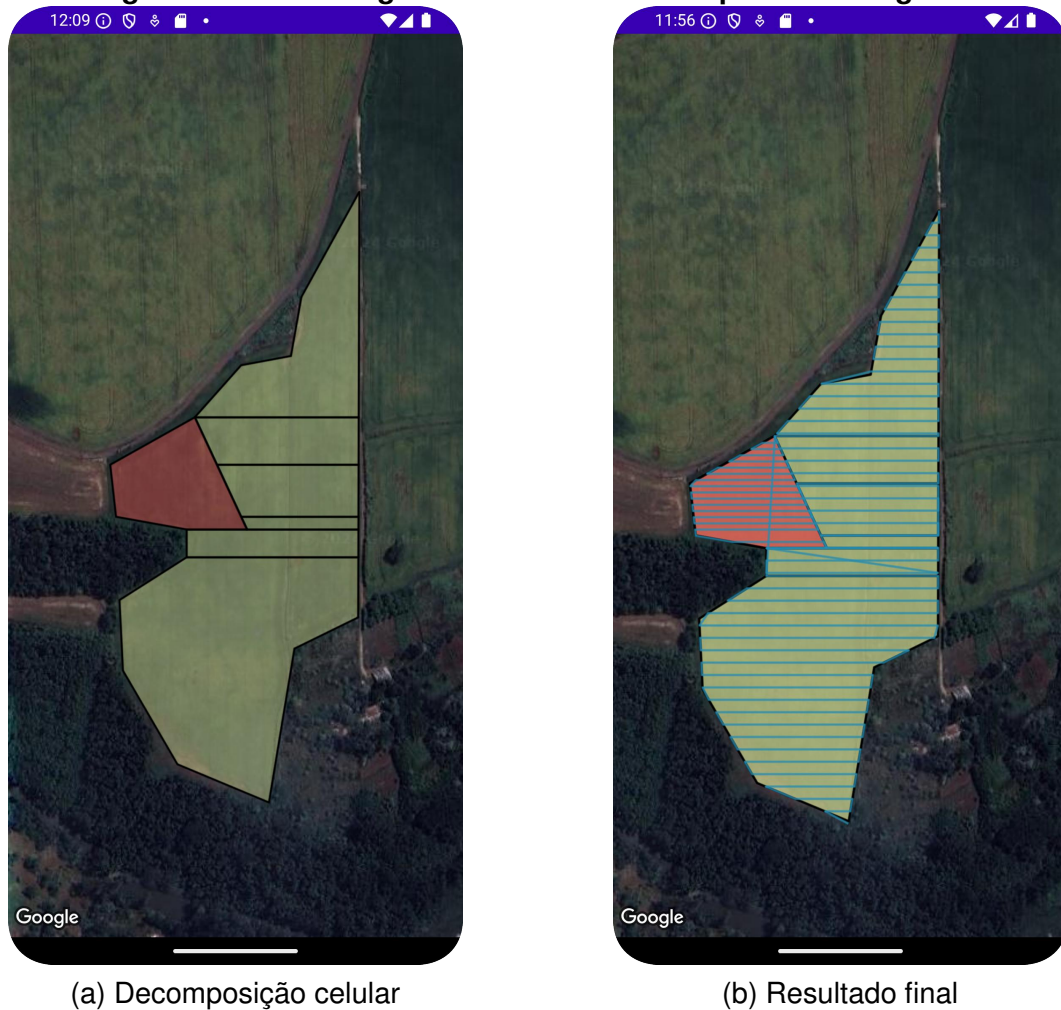


Fonte: Elaborado pelo autor

Para o cenário heterogêneo, foi adicionado apenas uma subárea especial à esquerda da área de interesse. A Figura 25a expressa o resultado da decomposição celular nessa região e as observações dos experimentos anteriores também estão presentes aqui: a remoção das sobreposições entre subáreas normais e especiais, a permanência de algumas concavidades e a possível combinação de células vizinhas. O caminho de cobertura obtido (Figura 25b) possibilita coletar informações de toda a superfície de entrada, apresentando novamente uma pequena falha na troca de arestas dentro da área especial, como também foi comentado durante a Seção 4.1.

Ao analisar os dados presentes na Tabela 5, pode-se constatar que o caminho

Figura 25: Caminho gerado com subáreas especiais na região 3



Fonte: Elaborado pelo autor

obtido pelos cenários da região 3 reforçam os pontos sugeridos pela Tabela 3 e pela Tabela 4. O comprimento final intermediário aferido no cenário com mudanças de altitude se vê mais atrativo em termos de recursos se comparado a um sobrevoo todo mais próximo do solo, trazendo um equilíbrio entre os recursos gastos e os dados obtidos a cada sobrevoo. A relação entre os tempos de execução também seguem o mesmo comportamento das tabelas anteriores, deixando evidente o aumento da complexidade do algoritmo no terceiro cenário, mesmo com múltiplas execuções, com valores discrepantes e um alto desvio padrão.

Tabela 5: Métricas dos experimentos da região 3

Cenário	Comprimento do caminho final (m)	Duração total (ms)
Alta altitude	7332,052	$1,644 \pm 2,969$
Baixa altitude	13549,383	$4,569 \pm 17,859$
Híbrido	8969,057	$7,109 \pm 19,439$

Fonte: Elaborado pelo autor

4.4 COMPARAÇÃO DOS EXPERIMENTOS

Ao executar o algoritmo detalhado neste trabalho em diversos cenários, evidenciam-se as vantagens e aplicações para este como também as limitações presentes na implementação atual. Com a última se tornando explícita na Figura 23a que, por conta de um cenário não previsto, teve como resultado a perda de parte da área de interesse durante a decomposição celular. O que demonstra a alta dificuldade encontrada em problemas que envolvem operações com formas geométricas complexas e o cuidado que se deve ter ao entender e sugerir soluções para estes.

Apesar dessas falhas pontuais, os resultados foram valiosos para apontar como um algoritmo de planejamento de cobertura de caminho heterogêneo pode contribuir para a coleta de dados com um melhor custo benefício. A Tabela 6 concatena todos os resultados quantitativos em uma única tabela, facilitando a observação de padrões encontrados durante os diversos experimentos. Nela fica evidente o ganho dos cenários híbridos ao planejar um caminho de cobertura consideravelmente menor que os cenários totalmente em baixa altitude. Portanto, permitindo-se atribuir um sobrevoo mais próximo da vegetação para as subregiões nas quais se necessita de uma resolução maior de dados, enquanto que, economiza recursos ao sobrevoar em uma altitude maior as regiões que não se beneficiam desse nível de detalhamento”.

Os resultados discutidos no Capítulo 4 demonstram a capacidade do algoritmo em lidar com entradas diversas, polígonos simples e complexos, com nenhuma ou múltiplas subáreas especiais. Demonstra também, apesar da alta complexidade, uma performance eficiente e satisfatória para a aplicação proposta, levando em seu pior cenário cerca de 26 ms para retornar um caminho de cobertura válido para a entrada mais complexa citada.

Tabela 6: Métricas de todos os experimentos

Região	Cenário	Células geradas	Comprimento do caminho final (m)	Duração total (ms)
1	Alta altitude	1	9702,161	$0,485 \pm 0,587$
1	Baixa altitude	1	18448,537	$1,363 \pm 5,553$
1	Híbrido	7	11920,543	$2,646 \pm 1,475$
2	Alta altitude	1	11623,801	$0,654 \pm 2,762$
2	Baixa altitude	1	22245,648	$0,817 \pm 3,948$
2	Híbrido	8	15379,271	$3,866 \pm 7,403$
3	Alta altitude	2	7332,052	$1,644 \pm 2,969$
3	Baixa altitude	2	13549,383	$4,569 \pm 17,859$
3	Híbrido	7	8969,057	$7,109 \pm 19,439$

Fonte: Elaborado pelo autor

Entretanto, o cálculo e a coleta da duração total dos experimentos deixa nítida uma dificuldade em se obter o tempo de execução exato de um programa em um sistema preemptivo. Mesmo ao tomar o cuidado para aferir os resultados utilizando o tempo médio de 30 execuções, percebe-se um valor de desvio padrão muito elevado, partindo de 56% até 483% do valor médio calculado. Como o algoritmo descrito e implementado durante este trabalho não apresenta nenhum fator de aleatoriedade, tomando exatamente as mesmas decisões todas as vezes em que é executado, ou quaisquer dependências externas que passe por uma rede de comunicação ou outro sistema com alguma latência a ser considerada. Entende-se como principal causador destas anormalidades a preempção inerente em sistemas operacionais modernos, que se torna explícita em curtos períodos de tempo, já que o cenário que resulta na maior duração total se encontra na ordem dos milissegundos.

Para uma análise mais profunda do tempo de execução das diferentes etapas do método descrito neste trabalho, o Apêndice A contém os valores da duração de cada uma dessas etapas para os 9 cenários discutidos no decorrer deste capítulo.

Por fim, entende-se os resultados dos experimentos aqui descritos como altamente proveitosos para validar e contrapor o método proposto, no qual o caminho de cobertura final para uma área de interesse com subáreas de diferentes parâmetros foi gerado com sucesso e, simultaneamente, observam-se pontos de oportunidade para se realizar ajustes e outras melhorias na implementação aqui detalhada.

5 CONCLUSÃO

Para o contexto de agricultura de precisão, ser capaz de coletar dados de qualidade e acurados é fundamental para possibilitar a realização de análises do estado atual da lavoura, compreendendo quais seções necessitam de intervenção e quais ações devem ser tomadas para se ter o melhor resultado. Para isso, é necessário ter a disposição métodos eficientes e baratos que possibilitem a coleta desses dados de forma recorrente. Atualmente, uma das formas mais acessíveis para coletar imagens aéreas de uma localização é por meio do sobrevoo de aeronaves não tripuláveis.

Dessa forma, para que se possa aumentar a flexibilidade das missões de VANTs em regiões agrícolas ao considerar múltiplas altitudes para a coleta dos dados em determinadas subregiões, foi descrito neste trabalho um novo método de planejamento de caminho de cobertura. Tal abordagem se aproveita da mudança na elevação da câmera para registrar imagens com resoluções e níveis de detalhes distintas, na tentativa de otimizar a relação entre os dados que são de fato relevantes para uma análise posterior e os recursos gastos durante a execução da missão.

Os resultados obtidos foram favoráveis para o algoritmo concebido que se mostrou hábil em reduzir a complexidade do problema ao subdividir a área de interesse em subáreas e conectar estas por meio de um único caminho de cobertura final. O algoritmo foi validado tanto para casos mais simples em uma área de interesse uniforme em formato de trapézio quanto para casos mais complexos em uma área de interesse com múltiplas concavidades e subáreas especiais. Em todos os cenários foi gerado um caminho confiável e sem a ocorrência de sobreposições ou a duplicação de uma mesma localidade que poderia gerar a cobertura de um mesmo ponto por duas vezes ou mais. A sugestão de um caminho heterogêneo com subáreas especiais que necessitam de imagens em resoluções maiores também se mostrou benéfico ao resultar em um comprimento de caminho notavelmente menor do que nos testes com o sobrevoo realizado todo em uma altitude mais baixa, apresentando um comprimento até 35,4% menor para o caso híbrido em comparação com o caso em baixa altitude da mesma região.

Outro ponto a se salientar dos resultados é o tempo de execução que, mesmo com os experimentos simulando regiões agrícolas e valores de entrada similares às situações reais para as quais o algoritmo foi proposto, foi capaz de oferecer um resultado satisfatório para esses experimentos em poucos milissegundos. Sendo o pior cenário observado a partir dos dados coletados sendo de cerca de 26 ms para o planejamento do caminho de cobertura completo com diferentes altitudes em uma área de interesse total de 7,38 ha.

Apesar de falhas estarem presentes durante a execução dos experimentos, estas se manifestam não dos métodos propostos, mas sim de casos isolados gerados pelo formato e interação de estruturas geométricas, exigindo um polimento de etapas específicas implementadas para que estes casos sejam corrigidos. Dessa forma, a aplicação e aprimoramento deste algoritmo se vê como uma alternativa capaz de suprir necessidades específicas na coleta de dados e sobrevoos de cobertura.

5.1 TRABALHOS FUTUROS

Tendo em vista os objetivos apresentados neste trabalho, os resultados coletados e discutidos, diversas são as contribuições futuras que podem ser feitas a fim de continuar e aprimorar esta pesquisa.

Projetar e desenvolver uma interface entre o algoritmo de planejamento de caminho de cobertura heterogêneo e o usuário final, para que este último possa interagir na etapa de delimitação de área de interesse e suas subáreas de forma mais intuitiva, permitindo também a escolha de parâmetros como altitude e GSD. Essa interface poderia fazer a comunicação com o veículo aéreo, enviando o caminho de cobertura direto para a aeronave conectada, possibilitando a realização de um sobrevoos com uma aeronave real para validar todos os pontos aqui apresentados e, tornando assim, o fluxo da aplicação completo.

Aprimorar a solução já implementada, como comentado durante todo o texto, decisões foram tomadas durante o desenvolvimento para agilizar o mesmo, com o custo do algoritmo atender a um conjunto restritivo de entradas. Ao retornar a estas decisões, aumenta-se a flexibilidade do algoritmo diante as mais diversas formas e entradas. Como a definição dos eixos de cortes durante o processo de decomposição celular, que podem ser escolhidos em tempo de execução para utilizarem um ângulo que possa minimizar o número de células no resultado final. Para isso será necessário

desenvolver uma forma confiável e geral que consiga definir estes ângulos em um tempo de execução viável. Áreas de interesse com buracos e subáreas especiais que não compartilham uma aresta com outra subárea, estando totalmente dentro ou até parcialmente fora de uma subárea normal, foram cenários não abordados devido às limitações no tempo de desenvolvimento para testar e validar regras e processos gerais que pudessem decompor estes casos. Foi mencionado também que ao final da decomposição pode-se concatenar células vizinhas que apresentam propriedades geométricas parecidas e que quando unidas não geram nenhuma complicação para as etapas posteriores, diminuindo o número de decisões que devem ser tomadas durante as etapas posteriores do algoritmo e se assemelhando mais ao método de bustrofédon citado durante o trabalho.

Já durante a etapa de escolha da ordem das células existem dois caminhos para otimizações. A definição e execução com novas funções objetivas que podem levar em considerações outras informações se as duas células são vizinhas, ou o número de outras células próximas para priorizar a célula que fica na extremidade do grafo, além de adicionar um peso distinto referente a troca de altitude que a aeronave deverá fazer para células de tipos diferentes. Cada uma dessas decisões alteram o comportamento da escolha da ordem das células, podendo ser benéficas em casos específicos. Outra melhoria desta etapa está justamente no algoritmo que buscará uma ordem subótima para percorrer o grafo tendo em base os custos presentes na matriz objetivo. A implantação atual utiliza de uma estratégia gulosa para se obter esta resposta, entretanto, diversos são os trabalhos da literatura que proveem alternativas mais eficientes para visitar uma única vez todos os nós de um grafo com pesos, também conhecido como problema do caixeiro viajante. Ao procurar e entender estas opções pode-se encontrar um método mais compatível para se encontrar a melhor ordem das células.

A geração do caminho de cobertura de cada célula ocorre na mesma direção para qualquer formato representado pelo polígono desta, porém, é possível definir um ângulo de direção dinâmico que leve em consideração a aresta mais longa do polígono. Desta forma, se minimiza o número de iterações do algoritmo e curvas necessárias para cobrir toda a área de interesse. Mas, uma vez que este ângulo possa diferir do utilizado durante a decomposição celular, deve-se tomar o cuidado para que as decisões realizadas em uma das etapas não prejudiquem a execução da outra ao gerar células que aumentem o número de iterações necessárias ou que tenham concavidades prejudiciais no ângulo da maior aresta.

Algumas irregularidades foram observadas durante a análise dos resultados, como falhas na remoção de sobreposições da região 2 devido à geração de duas subáreas durante o algoritmo de corte, falhas na remoção de concavidades da região 3 e lacunas presentes no caminho gerado de alguns cenários devido à troca da aresta de referência. Estas irregularidades devem ser investigadas mais a fundo e corrigidas visando uma maior consistência e confiabilidade do caminho gerado. Além disso, existe um amplo espaço para otimizações do código, reduções do número de laços de repetições, refatorações mais eficientes e um melhor uso da memória, o que pode reduzir o salto no tempo de execução do algoritmo.

REFERÊNCIAS

- ACAR, E. U. et al. Morse Decompositions for Coverage Tasks. **The International Journal of Robotics Research**, v. 21, n. 4, p. 331–344, abr. 2002. ISSN 0278-3649, 1741-3176. Disponível em: <http://journals.sagepub.com/doi/10.1177/027836402320556359>.
- BAHNEMANN, R. et al. Revisiting Boustrophedon Coverage Path Planning as a Generalized Traveling Salesman Problem. In: ISHIGAMI, G.; YOSHIDA, K. (Ed.). **Field and Service Robotics**. Singapore: Springer Singapore, 2021. v. 16, p. 277–290. ISBN 9789811594595 9789811594601. Series Title: Springer Proceedings in Advanced Robotics. Disponível em: http://link.springer.com/10.1007/978-981-15-9460-1_20.
- CABREIRA, T. M.; BRISOLARA, L. B.; FERREIRA JR, P. R. Survey on coverage path planning with unmanned aerial vehicles. **Drones**, MDPI, v. 3, n. 1, p. 4, 2019.
- CHOSSET, H. Coverage for robotics – A survey of recent results. **Annals of Mathematics and Artificial Intelligence**, v. 31, p. 113–126, 2001. ISSN 1573-7470.
- CHOSSET, H.; PIGNON, P. Coverage path planning: The boustrophedon cellular decomposition. In: ZELINSKY, A. (Ed.). **Field and service robotics**. London: Springer London, 1998. p. 203–209. ISBN 978-1-4471-1273-0.
- COOMBES, M.; CHEN, W.-H.; LIU, C. Fixed wing uav survey coverage path planning in wind for improving existing ground control station software. In: IEEE. **2018 37th Chinese Control Conference (CCC)**. Wuhan, 2018. p. 9820–9825.
- DJI. **Mavic Pro - DJI**. 2016. Acessado em 15 de Março de 2023. Disponível em: <https://www.dji.com/br/mavic>.
- DJI. **Mobile SDK V4: Documentation Introduction**. 2020. Acessado em 6 de Março de 2023. Disponível em: <https://developer.dji.com/document/544659e8-9dab-4ad8-9414-a31e1c9b89b1>.
- EMBRAPA. Tecnologia em mecanizacao no brasil: equipamentos e sistemas para o futuro. **Seminário Temático da EMBRAPA. Centro Nacional de Pesquisa de Milho e Sorgo**, EMBRAPA-CNPMS, Sete Lagoas, n. Documento 10, p. 35, 1997.
- EMBRAPA. **VII Plano Diretor da Embrapa 2020-2030**. Brasília, DF, 2020. 31p p. Disponível em: <https://ainfo.cnptia.embrapa.br/digital/bitstream/item/217274/1/VII-PDE-2020.pdf>.
- FELIPE-GARCIA, B.; HERNANDEZ-LOPEZ, D.; LERMA, J. L. Analysis of the ground sample distance on large photogrammetric surveys. **Applied Geomatics**, v. 4, n. 4, p. 231–244, dez. 2012. ISSN 1866-928X. Disponível em: <https://doi.org/10.1007/s12518-012-0084-2>.

FRANCO, C. D.; BUTTAZZO, G. Energy-aware coverage path planning of uavs. In: IEEE. **2015 IEEE international conference on autonomous robot systems and competitions**. Vila Real, 2015. p. 111–117.

GALCERAN, E.; CARRERAS, M. A survey on coverage path planning for robotics. **Robotics and Autonomous Systems**, v. 61, n. 12, p. 1258–1276, dez. 2013. ISSN 09218890. Disponível em: [〈https://linkinghub.elsevier.com/retrieve/pii/S092188901300167X〉](https://linkinghub.elsevier.com/retrieve/pii/S092188901300167X).

GOOGLE. **Maps SDK for Android overview**. 2023. Acessado em 6 de Março de 2023. Disponível em: [〈https://developers.google.com/maps/documentation/android-sdk/overview〉](https://developers.google.com/maps/documentation/android-sdk/overview).

GOV. **Vendas externas do agronegócio somam US\$ 96,8 bilhões em 2019: milho, carnes e algodão foram destaque nas exportações**. 2020. Acessado em 29 de Setembro de 2022. Disponível em: [〈https://www.gov.br/pt-br/noticias/agricultura-e-pecuaria/2020/01/vendas-externas-do-agronegocio-somam-us-96-8-bilhoes-em-2019#:~:text=Em%202018%2C%20a%20participa%C3%A7%C3%A3o%20do,43%2C25%20milh%C3%B5es%20de%20toneladas〉](https://www.gov.br/pt-br/noticias/agricultura-e-pecuaria/2020/01/vendas-externas-do-agronegocio-somam-us-96-8-bilhoes-em-2019#:~:text=Em%202018%2C%20a%20participa%C3%A7%C3%A3o%20do,43%2C25%20milh%C3%B5es%20de%20toneladas).

HERT, S.; TIWARI, S. A terrain-covering algorithm for an AUV. **Autonomous Robots**, v. 3, n. 2, p. 91–119, 1996. ISSN 1573-7527.

HRISTOV, G. et al. An overview of the use of unmanned aerial vehicles for precision agriculture. In: IEEE. **2020 International Conference on Biomedical Innovations and Applications (BIA)**. Varna, 2020. p. 137–140.

LATOMBE, J.-C. Exact Cell Decomposition. In: **Robot Motion Planning**. Boston, MA: Springer US, 1991. p. 200–247. ISBN 978-0-7923-9206-4 978-1-4615-4022-9. Disponível em: [〈http://link.springer.com/10.1007/978-1-4615-4022-9_5〉](http://link.springer.com/10.1007/978-1-4615-4022-9_5).

LEI, T. et al. Deep learning-based complete coverage path planning with re-joint and obstacle fusion paradigm. **Frontiers in Robotics and AI**, Frontiers Media SA, v. 9, 2022.

LIU, Y.; LI, H.; LIU, Z. Real-Time Coverage Path Planning of a UAV with Threat and Value Zone Constraints. In: JIA, Y.; ZHANG, W.; FU, Y. (Ed.). **Proceedings of 2020 Chinese Intelligent Systems Conference**. Singapore: Springer Singapore, 2021. v. 705, p. 840–848. ISBN 9789811584497 9789811584503. Series Title: Lecture Notes in Electrical Engineering. Disponível em: [〈http://link.springer.com/10.1007/978-981-15-8450-3_87〉](http://link.springer.com/10.1007/978-981-15-8450-3_87).

LUCCHI, B. B.; SILVA, M.; BARROS, G. S. d. C. **PIB do agronegócio 2022**. Piracicaba, 2023. Disponível em: [〈https://www.cepea.esalq.usp.br/upload/kceditor/files/PIB-DO-AGRONEGOCIO-2022.17MAR2023.pdf〉](https://www.cepea.esalq.usp.br/upload/kceditor/files/PIB-DO-AGRONEGOCIO-2022.17MAR2023.pdf).

MOGILI, U. R.; DEEPAK, B. B. V. L. Review on Application of Drone Systems in Precision Agriculture. **Procedia Computer Science**, v. 133, p. 502–509, 2018. ISSN 18770509. Disponível em: [〈https://linkinghub.elsevier.com/retrieve/pii/S1877050918310081〉](https://linkinghub.elsevier.com/retrieve/pii/S1877050918310081).

MORAVEC, H.; ELFES, A. High resolution maps from wide angle sonar. In: **Proceedings. 1985 IEEE International Conference on Robotics and Automation**. St. Louis, MO, USA: Institute of Electrical and Electronics Engineers, 1985. v. 2, p. 116–121. Disponível em: <http://ieeexplore.ieee.org/document/1087316/>.

OLIVEIRA, A. J. de et al. Potencialidades da utilização de drones na agricultura de precisão. **Brazilian Journal of Development**, v. 6, n. 9, p. 64140–64149, 2020.

RABELLO, A. et al. Mobile system for optimized planning to drone flight applied to the precision agriculture. In: IEEE. **2020 3rd International Conference on Information and Computer Technologies (ICICT)**. San Jose, 2020. p. 12–16.

SHARMA, M.; HEMA, N. Comparison of agricultural drones and challenges in implementation: A review. In: IEEE. **2021 7th International Conference on Signal Processing and Communication (ICSC)**. Noida, 2021. p. 26–30.

SHIRATSUCHI, L. S. O avanço dos drones. **Revista DBO**, v. 33, n. 403, p. 20–25, 2014. Disponível em: <http://www.infoteca.cnptia.embrapa.br/infoteca/handle/doc/1003261>.

SILVA, N. B. F. d. **Development of an autonomous unmanned aerial vehicle specification of a fixed-wing vertical takeoff and landing aircraft**. Tese (Doutorado Direto em Ciências de Computação e Matemática Computacional) — Universidade de São Paulo, São Carlos, out. 2018. Disponível em: <http://www.teses.usp.br/teses/disponiveis/55/55134/tde-16102018-100220/>.

SONG, Z. et al. Unmanned aerial vehicle coverage path planning algorithm based on cellular automata. In: IEEE. **2019 15th International Conference on Computational Intelligence and Security (CIS)**. Changchun, 2019. p. 123–126.

STEFANOPOULOU, A. et al. Improving time and energy efficiency in multi-UAV coverage operations by optimizing the UAVs' initial positions. **Int. J. Intell. Robot. Appl.**, Springer Science and Business Media LLC, v. 8, n. 3, p. 629–647, 2024.

THRUN, S. Learning metric-topological maps for indoor mobile robot navigation. **Artificial Intelligence**, v. 99, n. 1, p. 21–71, fev. 1998. ISSN 00043702. Disponível em: <https://linkinghub.elsevier.com/retrieve/pii/S0004370297000787>.

TSCHIEDEL, M.; FERREIRA, M. F. Introdução à agricultura de precisão: conceitos e vantagens. **Ciência Rural**, SciELO Brasil, v. 32, p. 159–163, 2002.

VALAVANIS, K. **Advances in Unmanned Aerial Vehicles: State of the Art and the Road to Autonomy**. Berlim: Springer Dordrecht, 2007. 544 p. ISBN 978-1-4020-6113-4.

VALAVANIS, K. P.; VACHTSEVANOS, G. J. (Ed.). **Handbook of Unmanned Aerial Vehicles**. Piscataway: Springer Netherlands, 2015. ISBN 978-90-481-9706-4 978-90-481-9707-1. Disponível em: <http://link.springer.com/10.1007/978-90-481-9707-1>.

WAWRZYN, D. **What is Ground Sample Distance and How Does it Affect Your Drone Data?** 2023. Disponível em: <https://www.propelleraero.com/blog/ground-sample-distance-gsd-calculate-drone-data/>.

WEILER, K.; ATHERTON, P. Hidden surface removal using polygon area sorting. **ACM SIGGRAPH computer graphics**, ACM New York, NY, USA, v. 11, n. 2, p. 214–222, 1977. Disponível em: <https://dl.acm.org/doi/pdf/10.1145/563858.563896>.

WILLIAMS, E. Navigation on the spheroidal earth. **Journal of Navigation. Cambridge**, 2002. Disponível em: <http://edwilliams.org/ellipsoid/ellipsoid.pdf>.

WU, W. et al. Evaluation and enhancement of resolution-aware coverage path planning method for surface inspection using unmanned aerial vehicles. **IEEE Access**, IEEE, 2024.

XIE, J.; CARRILLO, L. R. G.; JIN, L. Path planning for uav to cover multiple separated convex polygonal regions. **IEEE Access**, IEEE, v. 8, p. 51770–51785, 2020.

YU, J. et al. Dynamic coverage path planning of energy optimization in uav-enabled edge computing networks. In: IEEE. **2021 IEEE Wireless Communications and Networking Conference Workshops (WCNCW)**. [S.l.], 2021. p. 1–6.

ZELENKA, J.; KASANICKÝ, T. Insect Pheromone Strategy for the Robots Coordination. **Applied Mechanics and Materials**, v. 613, p. 163–171, ago. 2014. ISSN 1662-7482. Disponível em: <https://www.scientific.net/AMM.613.163>.

ZHANG, N.; WANG, M.; WANG, N. Precision agriculture – a worldwide overview. **Computers and Electronics in Agriculture**, v. 36, n. 2-3, p. 113–132, nov. 2002. ISSN 01681699. Disponível em: <https://linkinghub.elsevier.com/retrieve/pii/S0168169902000960>.

APÊNDICE A – TEMPO DE EXECUÇÃO DE CADA ETAPA DO ALGORITMO PARA OS EXPERIMENTOS

A Tabela 7 apresenta a média e o desvio padrão dos tempos de execução de cada etapa do algoritmo. É possível perceber que a etapa mais custosa é a de decomposição celular, principalmente para o cenário híbrido.

Tabela 7: Duração de cada etapa do algoritmo para todos os experimentos

Região	Cenário	Duração da etapa (ms)			
		Decomposi- ção	Função objetivo	Ordem das células	Geração de caminho
1	Alta altitude	0,233 ±	0,015 ±	0,016 ±	0,220 ±
		0,414	0,084	0,070	0,335
1	Baixa altitude	0,739 ±	0,014 ±	0,014 ±	0,596 ±
		5,233	0,030	0,022	2,153
1	Híbrido	1,979 ±	0,166 ±	0,058 ±	0,444 ±
		1,404	0,104	0,106	0,221
2	Alta altitude	0,406 ±	0,012 ±	0,014 ±	0,222 ±
		2,700	0,017	0,018	0,592
2	Baixa altitude	0,295 ±	0,014 ±	0,015 ±	0,494 ±
		0,231	0,038	0,025	3,961
2	Híbrido	2,851 ±	0,398 ±	0,121 ±	0,496 ±
		4,408	5,523	0,904	0,407
3	Alta altitude	1,311 ±	0,035 ±	0,035 ±	0,263 ±
		2,965	0,079	0,165	0,301
3	Baixa altitude	3,156 ±	0,163 ±	0,052 ±	1,198 ±
		15,829	1,174	0,098	4,589
3	Híbrido	5,754 ±	0,337 ±	0,126 ±	0,892 ±
		17,138	0,921	0,252	4,985

Fonte: Elaborado pelo autor

