

UNIVERSIDADE FEDERAL DE JUIZ DE FORA  
FACULDADE DE ENGENHARIA  
PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA  
ELÉTRICA

Iago Zanuti Biundini

Planejamento Online de Caminhos por Cobertura através de  
Meta-Heurística

Juiz de Fora

2022

**Iago Zanuti Biundini**

**Planejamento Online de Caminhos por Cobertura através de  
Meta-Heurística**

Tese apresentada ao Programa de Pós-graduação em Engenharia Elétrica da Universidade Federal de Juiz de Fora como requisito parcial à obtenção do título de Doutor em Engenharia Elétrica. Área de concentração: Energia

Orientador: Prof. André Luís Marques Marcato, D.Sc.

Coorientadora: Prof. Milena Faria Pinto, D.Sc.

Juiz de Fora

2022

Ficha catalográfica elaborada através do programa de geração automática da Biblioteca Universitária da UFJF, com os dados fornecidos pelo(a) autor(a)

Biundini, Iago Zanuti.

Planejamento Online de Caminhos por Cobertura através de Meta-Heurística /Iago Zanuti Biundini. -- 2022.

144 p. : il.

Orientador: André Luis Marques Marcato

Coorientadora: Milena Faria Pinto

Tese (doutorado) - Universidade Federal de Juiz de Fora, Faculdade de Engenharia. Programa de Pós-Graduação em Engenharia Elétrica, 2022.

1. Planejamento de Caminho por Cobertura. 2. Meta-heurística. 3. VANT. I. Marcato, André Luis Marques, orient. II. Pinto, Milena Faria, coorient. III. Título.

**Iago Zanuti Biundini**

**Planejamento Online de Caminhos por Cobertura através de Meta-Heurística**

Tese apresentada ao Programa de Pós-Graduação em Engenharia Elétrica da Universidade Federal de Juiz de Fora como requisito parcial à obtenção do título de Doutor em Engenharia Elétrica. Área de concentração: Sistemas de Energia Elétrica

Aprovada em 23 de fevereiro de 2022.

**BANCA EXAMINADORA**

**Prof. Dr. André Luis Marques Marcato** - Orientador

Universidade Federal de Juiz de Fora

**Profa. Dra. Milena Faria Pinto**

Centro Federal de Educação Tecnológica Celso Suckow da Fonseca

**Prof. Dr. Andre Gustavo Scolari Conceição**

Universidade Federal da Bahia

**Prof. Dr. Diego Barreto Haddad**

Centro Federal de Educação Tecnológica Celso Suckow da Fonseca

**Prof. Dr. Eduardo Pestana de Aguiar**

Universidade Federal de Juiz de Fora

**Prof. Dr. Ivo Chaves da Silva Junior**  
Universidade Federal de Juiz de Fora

Juiz de Fora, 10/02/2022.



Documento assinado eletronicamente por **Andre Luis Marques Marcato, Professor(a)**, em 23/02/2022, às 12:48, conforme horário oficial de Brasília, com fundamento no § 3º do art. 4º do [Decreto nº 10.543, de 13 de novembro de 2020](#).



Documento assinado eletronicamente por **Milena Faria Pinto, Usuário Externo**, em 23/02/2022, às 12:51, conforme horário oficial de Brasília, com fundamento no § 3º do art. 4º do [Decreto nº 10.543, de 13 de novembro de 2020](#).



Documento assinado eletronicamente por **Ivo Chaves da Silva Junior, Professor(a)**, em 23/02/2022, às 13:15, conforme horário oficial de Brasília, com fundamento no § 3º do art. 4º do [Decreto nº 10.543, de 13 de novembro de 2020](#).



Documento assinado eletronicamente por **Andre Gustavo Scolari Conceicao, Usuário Externo**, em 23/02/2022, às 14:03, conforme horário oficial de Brasília, com fundamento no § 3º do art. 4º do [Decreto nº 10.543, de 13 de novembro de 2020](#).



Documento assinado eletronicamente por **Eduardo Pestana de Aguiar, Professor(a)**, em 23/02/2022, às 15:15, conforme horário oficial de Brasília, com fundamento no § 3º do art. 4º do [Decreto nº 10.543, de 13 de novembro de 2020](#).



Documento assinado eletronicamente por **Diego Barreto Haddad, Usuário Externo**, em 11/03/2022, às 14:40, conforme horário oficial de Brasília, com fundamento no § 3º do art. 4º do [Decreto nº 10.543, de 13 de novembro de 2020](#).



A autenticidade deste documento pode ser conferida no Portal do SEI-Ufjf ([www2.ufjf.br/SEI](http://www2.ufjf.br/SEI)) através do ícone Conferência de Documentos, informando o código verificador **0674946** e o código CRC **FD205086**.

Dedico este trabalho, em primeiro lugar, a minha  
família e amigos!

## **AGRADECIMENTOS**

Gostaria de agradecer ao meu orientador, o Dr. André Luís Marques Marcato, pela paciência, trabalho e conhecimento compartilhado nesses anos de pesquisa. Além desse, a minha família e amigos pelo apoio, em especial a Dra. Milena Faria Pinto por toda a ajuda e apoio nessa pesquisa. Não deixo de agradecer também ao imprescindível apoio da CAPES (Coordenação de Aperfeiçoamento de Pessoal de Nível Superior), INESC TEC (Instituto de Engenharia de Sistemas e Computadores de Portugal) e da CPFL durante o doutorado.

## RESUMO

Diferentes aplicações práticas com Veículos Aéreos Não Tripuláveis (VANTs) surgiram nos últimos anos, exigindo inspeções periódicas e detalhadas para verificar possíveis alterações estruturais. As inspeções usando veículos aéreos não tripulados VANTs devem minimizar o tempo de voo devido às restrições de tempo da bateria e identificar as características topográficas do terreno ou estruturas. Nesse sentido, o Planejamento do Caminho de Cobertura (CPP) visa encontrar o melhor caminho para a cobertura de uma determinada área respeitando as restrições da operação. As informações fotométricas do terreno são usadas para criar rotas ou mesmo refinar caminhos já criados, além de possibilitar diversos tipos de análises de imagens. Portanto, a principal contribuição desta pesquisa é o desenvolvimento de uma metodologia que utiliza um algoritmo meta-heurístico para criar missões otimizadas que buscam equilibrar dois objetivos conflitantes: tempo da missão e qualidade das imagens visando reconstruções 3D. A técnica foi aplicada tanto em um cenário simulado quanto em um ambiente real para verificar sua eficácia, buscando a aplicação de diversas técnicas meta-heurísticas e uma análise estatística sobre os resultados encontrados. Além disso o algoritmo foi aplicado nos mais diversas estruturas, tanto em taludes como em regiões a serem investigadas, através de sensores ativos como Lasers e mapas fornecidos de maneira *offline* através de *point cloud* e *digital elevation model*. Os resultados mostraram que o algoritmo foi capaz de criar missões otimizadas, equidistante a superfície e com todos os critérios de CPP sendo atendidos com equilíbrio dos objetivos de tempo e fotometria em comparação com outros algoritmos. Além disso, existe a possibilidade de controlar os dois objetivos propostos, visando aumentar a qualidade da reconstruções tridimensional e o tempo da missão.

Palavras-chave: Planejamento de Caminho por Cobertura. Meta-heurística. VANT.



## ABSTRACT

Different practical applications for Unmanned Aerial Vehicles (UAVs) have emerged in recent years, requiring periodic and detailed inspections to verify possible structural changes. Inspections using UAVs should minimize flight time due to battery time constraints and identify topographical features of terrain or structures. In this sense, coverage path planning (CPP) aims to find the best path to cover a given area while respecting the restrictions of the operation. Photometric terrain information is used to create routes or even refine already created paths, in addition to enabling different types of image analysis. Therefore, the main contribution of this research is the development of a methodology that uses a meta-heuristic algorithm to create optimized missions that seek to balance two conflicting objectives: mission time and image quality at 3D reconstructions. The technique was applied both in a simulated scenario and in a real environment to verify its effectiveness, seeking the application of several meta-heuristic techniques and a statistical analysis of the results found. In addition, the algorithm was applied to the most diverse structures, both on slopes and in regions to be investigated, through active sensors such as lasers and maps provided offline through point cloud and digital elevation models. The results showed that the algorithm was able to create optimized missions, equidistant from the surface and with all CPP criteria met. In addition, there is the possibility of controlling the two proposed objectives, aiming to increase the quality of the three-dimensional reconstructions and the mission time.

Keywords: Coverage Path Planning. Metaheuristic. UAV.

## LISTA DE ILUSTRAÇÕES

Figura 1 – Um exemplo de mapa de grade (grid map). . . . .	29
Figura 2 – Um exemplo de mapa trapezoidal. . . . .	30
Figura 3 – Um exemplo da diferença entre trapezoidal e boustrophedon. . . . .	31
Figura 4 – Estrutura de População do Algoritmo Genético. . . . .	35
Figura 5 – Resumo da Metodologia. . . . .	48
Figura 6 – Metodologia do Algoritmo de Planejamento de Caminho por Cobertura. . . . .	49
Figura 7 – Funcionamento do <i>LaserScan</i> e orientação do VANT. . . . .	51
Figura 8 – Resultado Final do Procedimento de Captura de informações do Ambiente em Simulador Gazebo-ROS. . . . .	52
Figura 9 – Comparação entre algoritmo Concave hull, Convex Hull e pontos próximos. . . . .	53
Figura 10 – Metodologia de aquisição de DEM. . . . .	54
Figura 11 – Esquemático do Problema. . . . .	57
Figura 12 – Esquemático do Cálculo de Pontos Paralelos com Distância $Dist_H$ . . . . .	60
Figura 13 – Exemplos de Missões desenvolvidas para duas respostas da Meta-heurística. . . . .	61
Figura 14 – Teste para análise do Ganho de velocidade máxima. . . . .	63
Figura 15 – Identificação de Pontos de Interesse em mundo real. . . . .	66
Figura 16 – Identificação de Pontos de Interesse em Simulador. . . . .	67
Figura 17 – Criação de Pontos com Aproximação de 1m. . . . .	68
Figura 18 – Modelo utilizado no Gazebo. . . . .	71
Figura 19 – Exemplo de distribuições normal e não normal. . . . .	73
Figura 20 – Exemplo de distribuições entre as turmas. . . . .	75
Figura 21 – Exemplo de variância intergrupo e intragrupo. . . . .	76
Figura 22 – Exemplo de grupos iguais e grupos distintos. . . . .	77
Figura 23 – Exemplo de Boxplot. . . . .	78
Figura 24 – Talude inclinado para teste de meta-heurística. . . . .	79

Figura 25 – Foto retirada de um talude do vertedouro da Usina Hidrelétrica Barra Grande em Pinhal da Serra, Rio Grande do Sul, Brasil . . . . .	80
Figura 26 – Leituras após filtro do talude inclinado para teste de meta-heurística. . . . .	80
Figura 27 – Histograma e qq-plot dos dados de Tempo de Missão, Cobertura Horizontal e Cobertura Vertical. . . . .	83
Figura 28 – Boxplot entre tempo de missão e a população. . . . .	85
Figura 29 – Boxplot entre cobertura horizontal e a população. . . . .	86
Figura 30 – Boxplot entre cobertura horizontal e a população. . . . .	87
Figura 31 – Boxplot entre Tempo de Processamento em ciclos de CPU e todas as populações. . . . .	89
Figura 32 – Boxplot entre Tempo de Processamento em ciclos de CPU e as populações menores. . . . .	89
Figura 33 – Boxplot entre Tempo de Processamento em segundos e todas as populações. . . . .	90
Figura 34 – Boxplot entre Tempo de Processamento em segundos e as populações menores. . . . .	91
Figura 35 – Boxplot entre tempo de missão e as meta-heurísticas com 300 de população. . . . .	93
Figura 36 – Boxplot entre cobertura horizontal e as meta-heurísticas com 300 de população. . . . .	94
Figura 37 – Boxplot entre cobertura vertical e as meta-heurísticas com 300 de população. . . . .	95
Figura 38 – Boxplot entre Fitness e as meta-heurísticas com 300 de população. . . . .	96
Figura 39 – Imagem do Talude 2. . . . .	97
Figura 40 – Leituras do Laser para o Talude 2. . . . .	98
Figura 41 – Missão para o Talude 2. . . . .	98
Figura 42 – Imagem do Talude 3. . . . .	99
Figura 43 – Leituras do Laser para o Talude 3. . . . .	99
Figura 44 – Missão para o Talude 3. . . . .	100
Figura 45 – Imagem do Talude 4. . . . .	100

Figura 46 – Leituras do Laser para o Talude 4. . . . .	101
Figura 47 – Missão para o Talude 4. . . . .	101
Figura 48 – Região no Simulador Gazebo-ROS para missões com DEM. . . . .	102
Figura 49 – DEM da região simulada. Em destaque a região das casas. . . . .	103
Figura 50 – Missão criada para um mapa de DEM. . . . .	103
Figura 51 – Missão com identificação dinâmica executada no simulador Gazebo-ROS. . . . .	105
Figura 52 – Talude para teste variação do tempo máximo de missão. . . . .	107
Figura 53 – Talude para teste variação do tempo máximo de missão no simulador Gazebo-ROS. . . . .	108
Figura 54 – Leituras do sensor laser após a aplicação do filtro de pontos próximos. . . . .	109
Figura 55 – Missão de teste de tempo máximo igual a 1 minutos. . . . .	110
Figura 56 – Missão de teste de tempo máximo igual a 5 minutos. . . . .	111
Figura 57 – Missão de teste de tempo máximo igual a 10 minutos. . . . .	111
Figura 58 – Missão de teste de tempo máximo igual a 10 minutos com correções de segurança. . . . .	112
Figura 59 – Boxplot do tempo de missão com a variação da relação entre os ganhos. . . . .	113
Figura 60 – Boxplot da cobertura horizontal com a variação da relação entre os ganhos. . . . .	114
Figura 61 – Boxplot da cobertura vertical com a variação da relação entre os ganhos. . . . .	115
Figura 62 – Estrutura Real utilizada para testes. . . . .	116
Figura 63 – Análise de normalidade das variáveis de decisão em ambiente real. . . . .	117
Figura 64 – Boxplot da variação de $Dist_H$ . . . . .	118
Figura 65 – Boxplot da variação de $Num_{way}$ . . . . .	119
Figura 66 – Boxplot da variação de $D_{min}$ . . . . .	120
Figura 67 – Exemplo de missão criada utilizando <i>DroneDeploy</i> . . . . .	122
Figura 68 – Exemplo de missão criada utilizando <i>polygon coverage planning</i> . . . . .	123
Figura 69 – Exemplo de missão criada utilizando DARP. . . . .	124

Figura 70 – Comparação entre missão proposta e dos outros algoritmos. 127

## LISTA DE TABELAS

Tabela 1 – Informações dos Trabalhos Correlatos - On/Offline, Ambiente e decomposição. . . . .	44
Tabela 2 – Informações dos Trabalhos Correlatos - Objetivos: Tempo e Fotometria. . . . .	45
Tabela 3 – Exemplo de tabela ANOVA. . . . .	74
Tabela 4 – Hiper-parâmetros das metaheurísticas utilizadas. . . . .	82
Tabela 5 – Tabela ANOVA de $T_{Missão}$ com a variação da população	84
Tabela 6 – Tabela ANOVA de $Cobertura_{Horizontal}$ com a variação da população . . . . .	85
Tabela 7 – Tabela ANOVA de $Cobertura_{Vertical}$ com a variação da população . . . . .	86
Tabela 8 – Tabela ANOVA de tempo de processamento em ciclos de clock com a variação da população. . . . .	88
Tabela 9 – Tabela ANOVA de Tempo de Processamento em segundos com a variação da população. . . . .	90
Tabela 10 – Tabela ANOVA de tempo de missão com a variação da população reduzida. . . . .	92
Tabela 11 – Tabela Anova de cobertura horizontal com a variação da população reduzida. . . . .	92
Tabela 12 – Tabela Anova de cobertura vertical com a variação da população reduzida. . . . .	92
Tabela 13 – Tabela Anova de tempo de missão com a variação da meta-heurística. . . . .	93
Tabela 14 – Tabela Anova da cobertura horizontal com a variação da meta-heurística. . . . .	94
Tabela 15 – Tabela Anova da cobertura vertical com a variação da meta-heurística. . . . .	95
Tabela 16 – Tabela Anova de <i>fitness</i> com a variação da meta-heurística	96
Tabela 17 – Resultados da missão dinâmica com identificação de pontos de interesse. . . . .	106

Tabela 18 – Comparação da variação do tempo máximo em relação as missões. . . . .	110
Tabela 19 – Tabela Anova de Tempo de missão com a variação da relação entre os ganhos. . . . .	113
Tabela 20 – Tabela Anova de cobertura horizontal com a variação da relação entre os ganhos. . . . .	113
Tabela 21 – Tabela Anova de cobertura vertical com a variação da relação entre os ganhos. . . . .	114
Tabela 22 – Tabela Anova de pontos na reconstrução 3D com a variação da distância ao talude. . . . .	118
Tabela 23 – Tabela Anova de pontos na reconstrução 3D com a variação do número de translados horizontais. . . . .	119
Tabela 24 – Tabela Anova de pontos na reconstrução 3D com a variação da distância entre os pontos. . . . .	120
Tabela 25 – Resultados comparativos entre metodologias em mundo real.	125

## LISTA DE ABREVIATURAS E SIGLAS

ACO	<i>Ant Colony Optimization</i>
ANOVA	<i>Analysis of variance</i>
AUV	<i>veículos subaquáticos autônomos</i>
BA	<i>Bat Algorithm</i>
BSA	<i>Bird Swarm Algorithm</i>
CA	<i>Culture Algorithm</i>
CLONALG	<i>Algoritmo Imunológico Artificial</i>
CPP	<i>Coverage Path Planning</i>
CPU	<i>Unidade de Central de Processamento</i>
CRITT	<i>Centro Regional de Inovação e Transferência de Tecnologia</i>
CRO	<i>Chemical Reaction Optimization</i>
DARPA	<i>Defense Advanced Research Projects Agency</i>
DEM	<i>digital elevation model</i>
FPA	<i>Flower Pollination Algorithm</i>
GA	<i>Genetic Algorithm</i>
GBMO	<i>Gases Brownian Motion Optimization</i>
GPS	<i>Global Positioning System</i>
IMU	<i>Inertial Measurement Unit</i>
K-NN	<i>k-vizinhos mais próximos</i>
LCA	<i>League Championship Algorithm</i>
LRS	<i>linear rank selection</i>
MEALPY	<i>MEta-heuristics ALgorithms in PYthon</i>
MEOP	<i>Otimização Evolutiva multi-objetivo</i>
MST	<i>Minimum Spanning Tree</i>
PCP	<i>polygon coverage planning</i>
PSO	<i>Particle swarm optimization</i>
ROS	<i>Robot Operating System</i>
RWS	<i>roulette wheel selection</i>
SA	<i>Simulated Annealing</i>
STC	<i>Spanning Tree Coverage</i>
TLBO	<i>Teaching-Learning-Based Optimization</i>
TOS	<i>tournament selection</i>
VANT	<i>Veículo Aéreo Não Tripulado</i>



## LISTA DE SÍMBOLOS

$Cobertura_{hor}$	Taxa de Interseção entre Imagens no eixo Horizontal
$Cobertura_{vert}$	Taxa de Interseção entre Imagens no eixo Vertical
$D_{min}$	Distância mínima entre dois pontos no eixo horizontal
$Dist_h$	Distância em relação a Superfície
$Dist_{hor}$	Distância Horizontal da abertura da câmera em relação a posição do VANT
$Dist_{vert}$	Distância Vertical da abertura da câmera em relação a posição do VANT
$f_{min}$	Frequência Mínima do morcego no BA
$f_{max}$	Frequência Máxima do morcego no BA
$h_{min}$	Altura Mínima da Superfície
$h_{max}$	Altura Máxima da Superfície
$lat_i$	Posição no Eixo $x$ do VANT no tempo $i$ (Latitude)
$lon_i$	Posição no Eixo $y$ do VANT no tempo $i$ (Longitude)
$Num_{way}$	Número de waypoints verticais na missão
$T_{Missão}$	Tempo total da missão
$vel_{VANT}$	Velocidade média do VANT
$\theta_{hor}$	Abertura Horizontal da câmera
$\theta_{yaw}$	Angulo YAW no VANT
$\theta_{vert}$	Abertura Vertical da câmera

## SUMÁRIO

<b>1</b>	<b>Introdução . . . . .</b>	<b>18</b>
1.1	Principais Objetivos . . . . .	21
1.2	Contribuições . . . . .	21
1.3	Publicações Relacionadas . . . . .	23
1.4	Organização da Proposta da Tese . . . . .	24
<b>2</b>	<b>Referencial Teórico . . . . .</b>	<b>25</b>
2.1	Planejamento de Caminho por Cobertura . . . . .	25
2.2	Meta-heurística . . . . .	32
2.2.1	Métodos baseados em biologia . . . . .	34
2.2.2	Métodos baseados em física . . . . .	38
2.2.3	Métodos baseados em outras ciências . . . . .	39
2.3	Trabalhos Correlatos . . . . .	40
2.3.1	Conclusões Parciais . . . . .	46
<b>3</b>	<b>Metodologia Proposta . . . . .</b>	<b>47</b>
3.1	Captura de Informações do Ambiente . . . . .	49
3.2	Planejamento do Caminhos . . . . .	54
3.3	Identificação de Objetivos Dinâmicos . . . . .	65
3.4	Missões de Aproximação e Readequação da Missão Geral . . . . .	67
3.5	Conclusões Parciais . . . . .	68
<b>4</b>	<b>Resultados e Discussões . . . . .</b>	<b>70</b>
4.1	Testes em simulação . . . . .	70
4.1.1	Informações sobre simuladores, VANT virtual e metodologia de comparação . . . . .	70
4.1.2	Comparação entre meta-heurísticas . . . . .	78
4.1.3	Identificação de pontos de interesse e missões dinâmicas . . . . .	104
4.1.4	Comparação entre tempos máximos permitidos e ganhos de <i>fitness</i> <sup>107</sup>	
4.2	Testes em ambiente real . . . . .	115
4.3	Comparação entre Algoritmos . . . . .	121
4.4	Conclusões Parciais . . . . .	127
<b>5</b>	<b>Conclusões e Propostas Futuras . . . . .</b>	<b>129</b>
	<b>REFERÊNCIAS . . . . .</b>	<b>132</b>

## 1 Introdução

No ano de 2015, no município de Mariana, Minas Gerais, e 2019, em Brumadinho no mesmo estado [Freitas et al. 2019] [Globo 2021], aconteceram rompimentos de barragens de detritos que causaram dezenas de mortos além de impactos socioeconômicos e ambientais. Essas instalações, assim como reservatórios de usinas hidroelétricas, possuem estruturas com grandes dimensões tais como taludes, pontes e represas. Estas construções estão sujeitas a forças e desgaste, o que exige inspeção periódica para garantir segurança e evitar acidentes como os citados anteriormente.

A inspeção dessas estruturas extensas e complexas é um desafio para a engenharia, executada tradicionalmente por humanos usando técnicas de escalada e descida usando cordas ou até mesmo equipamentos de alto custo como voos de helicópteros. Nesse sentido, nos últimos anos, devido a melhora das capacidades tanto de *hardware* quanto de *software*, tais como bateria, sistema de processamento embarcado, entre outros, o Veículo Aéreo Não Tripulado (VANT) passou a ser uma opção para investigações dessas regiões, o que aumenta a possibilidade de vistorias regulares. Como exemplo, Khaloo et al. [Khaloo et al. 2018] e Buffi et al. [Buffi et al. 2017] apresentaram um aplicativo VANT para realizar a avaliação e inspeção de destas áreas. As imagens adquiridas pelo VANT foram georreferenciadas por marcadores instalados na superfície inspecionada para aumentar a confiabilidade da vistoria. Um ponto crítico dos trabalhos citados é o planejamento adequado do caminho para fornecer informações confiáveis. O planejamento do caminho deve ser eficiente e também fornecer informações suficientes para as análises necessárias.

O campo da robótica obteve muitos avanços devido ao progresso dos *hardwares* e dos *softwares* envolvidos. Na inteligência artificial, o planejamento originalmente significava uma busca por uma sequência de operadores ou ações lógicas que transformam um estado inicial em um estado de objetivo desejado. Na teoria do controle, o foco está nos algoritmos que calculam trajetórias viáveis para sistemas com objetivos adicionais, tais como evitar obstáculos, por exemplo, e otimização [Aggarwal e Kumar 2020].

Uma necessidade fundamental da robótica é ter algoritmos que convertam especificações de alto nível de tarefas criadas por humanos, em descrições de baixo

nível, tal como o equipamento se moverá. Os termos “planejamento de movimento” e “planejamento de trajetória” são usados para esses tipos de problemas. Uma versão clássica do planejamento de movimento é conhecida como: “Problema de Mover um Piano” [Galceran e Carreras 2013]. Imagine dar um modelo de uma casa e um piano como entrada para um algoritmo. O algoritmo deve determinar como movê-lo de uma sala para outra na casa sem colidir em nada. Sendo um desafio clássico, pode-se adapta-lo para realizar um caminho evitando obstáculos. O planejamento de movimento (*Motion planning*) ignora a dinâmica e outras restrições diferenciais e se concentra nas translações e rotações necessárias. Já o planejamento de trajetória (*Trajectory planning*) inclui o planejamento do caminho com a velocidade, o tempo e a cinemática do movimento. O presente trabalho se insere no planejamento de caminho, já que além das questões de rotação e translação, o planejamento de caminho considera a velocidade e o tempo do VANT para a criação das missões, mas sem se preocupar com a cinemática do VANT, diferente do planejamento de trajetória.

O planejamento de caminhos é um tópico ativo da robótica e refere-se a encontrar uma rota ideal de um objeto em movimento desde o início até o ponto final sob certas restrições [Bo et al. 2015]. O CPP (Coverage Path Planning, ou Planejamento de Caminho por Cobertura em tradução literal) é a tarefa de determinar um caminho que passa por todos os pontos de uma área objetivo. Várias aplicações no campo da robótica são variações do problema de encontrar caminhos ideais. Além disso, os VANTs se posicionaram na vanguarda de diferentes campos de aplicação, como inspeção de infraestrutura [Besada et al. 2018, Biundini et al. 2020], busca e resgate [Pinto et al. 2019, Pinto et al. 2020], entrega de objetos [Murray e Raj 2020, Madridano et al. 2020], entre outros.

Além de encontrar uma solução, outros requisitos também são cruciais para o CPP. Em um dos primeiros trabalhos sobre CPP encontrados na literatura, Cao et al. [Cao e Huang 1988] definiu os requisitos que um determinado robô deve atender para uma operação de cobertura, os quais são:

- O robô deve percorrer todos os pontos na área de destino, cobrindo-o completamente;

- O robô deve preencher a região sem sobrepor caminhos;
- Operação contínua e sequencial sem repetição de caminhos é necessária;
- O robô deve evitar todos os obstáculos;
- Trajetórias de movimento simples (por exemplo, linhas retas ou círculos) devem ser usadas para simplificar o controle.

Essas condições demonstram que o robô deve cobrir uma região inteira sem sobrepor caminhos, manter a segurança de operação e evitar desperdícios de rota. Nem sempre todos esses requisitos podem ser satisfeitos, pois o ambiente onde o está ocorrendo o planejamento pode não ser conhecido. Dessa forma, utilizam-se essas considerações como plano inicial do planejamento e, caso não seja possível atendê-las em completo, permanecer com a de maior prioridade de operação, ou seja, segurança.

Em alguns casos, a minimização da distância, o uso de energia, o tempo da missão ou outros parâmetros são essenciais. Por exemplo, o trabalho de [Franco e Buttazzo 2015] propôs um algoritmo de CPP que gera subcaminhos ótimos considerando os requisitos fotogramétricos e estima a viabilidade das restrições de energia do VANT. Em [Hayat et al. 2017], Hayat et al. propõe uma abordagem diferente para otimização de dois objetivos. O primeiro objetivo é encontrar um determinado alvo em um problema de cobertura e o segundo estabelece um link de comunicação. Apesar dos bons resultados, ambos os trabalhos se limitam à superfície 2D.

No caso de reconstrução de ambientes, existe a necessidade de que as fotos da superfície tenham sobreposição para a realização da reconstrução, coincidindo pontos equivalentes em mais de uma imagem. Para o planejamento de caminho, é necessário adicionar esta restrição aos requisitos do planejamento para que o mesmo tenha resultados satisfatórios. Esta condição, como exemplo, pode ser visto no “problema da galeria de arte” [Galceran e Carreras 2013]. Este problema consiste em alocar um número mínimo de seguranças necessários para vigiar em uma galeria poligonal, de modo que cada ponto do local fique visível para pelo

menos um guarda. No caso, quanto maior a área de sobreposição entre as imagens, maior a chance de coincidência entre as partes.

## 1.1 Principais Objetivos

O principal objetivo desta pesquisa é um método para otimizar caminhos no problema de CPP. Um VANT realizará esses caminhos, o qual será responsável por capturar as imagens usadas posteriormente para a reconstrução 3D. A finalidade é produzir reconstruções que possam ter qualidade de identificação de pontos e até mesmo comparações temporais entre as mesmas. Portanto, é necessário que o VANT seja capaz de obter fotos suficientes com uma sobreposição adequada para gerar os dados necessários da nuvem de pontos densos. Somados aos parâmetros fotométricos, é indispensável a otimização de parâmetros de voo, como uso de energia, segurança de manobras e tempo de voo. Desta forma, é proposta uma metodologia de planejamento de caminho por cobertura utilizando otimização meta-heurística e identificação de objetivos dinâmicos. Em resumo os objetivos iniciais da pesquisa são:

- Otimizar parâmetros de missão para CPP com objetivos de tempo e fotometria.
- Possibilitar a criação de missão equidistantes a superfícies ou terrenos com informações tridimensionais das áreas a serem investigadas;
- Criação de uma biblioteca de CPP para investigação de regiões tridimensionais aplicadas a robôs simulados e reais utilizando o Robot Operation System (ROS).
- Identificação de objetivos dinâmicos e adaptação dos caminhos gerados de forma *online* para aumentar a qualidade das reconstruções 3D nessas regiões.

## 1.2 Contribuições

A partir dos objetivos propostos na Seção 1.1, as principais contribuições desta tese de doutorado foram:

- **Algoritmo *online* de planejamento de caminho de cobertura utilizando meta-heurística para captura de imagens para reconstrução 3D com objetivos conflitantes;**

A presente metodologia apresenta um algoritmo de planejamento de caminho de cobertura *online* que tem como principal objetivo a captura de imagens para a reconstrução 3D de superfícies. É um algoritmo que tem capacidade de obter informações de superfícies, realizar o planejamento e a execução do mesmo considerando dois objetivos conflitantes: tempo de voo e interseção entre imagens, principal contribuição da tese. Para isso, o método de otimização são as meta-heurísticas por suas capacidades de representar o problema de forma mais simplificada e o controle do tempo da resposta, já que a preparação é realizada enquanto o VANT sobrevoa a região. Como adicional do método proposto tem a menor variação da densidade de pontos presente nas reconstruções 3D com as imagens otimizadas pelo método, além do ajuste de ângulos inerentes à qualidade das imagens obtidas.

- **Identificação de objetivos dinâmicos e adequação do planejamento a esses objetivos;**

O algoritmo proposto tem capacidade de identificar pontos de interesse durante o voo, ocasionando pequenas missões locais de aproximação para a captura de imagens mais próximas. Assim, existe um equilíbrio entre os objetivos globais de reconstrução de toda a superfície, sem perder a qualidade em pontos de interesse.

- **Adaptação do algoritmo de planejamento para modificações de trajetória em tempo real.**

Durante o processo de execução da missão, atualiza-se o caminho considerando a área ainda a ser coberta e o tempo disponível. O objetivo é corrigir possíveis atrasos da missões por questões não planejadas ou até mesmo pelas submissões de identificação de pontos de interesse.

### 1.3 Publicações Relacionadas

Este trabalho de pesquisa resultou até o presente momento em dois capítulos de livro internacional e um artigo de revista, listados abaixo:

- "Coverage Path Planning Optimization for Slopes and Dams Inspection", Biundini, I. Z., Melo, A. G., Pinto, M. F., Marins, G. M., Marcato, A. L., Honorio, L. M., Iberian Robotics conference. Springer, Cham, 2019. [Biundini et al. 2019].

O capítulo de livro apresenta o algoritmo de otimização baseado em morcegos para a otimização de trajetórias geradas manualmente por um operador. A partir do resultado manual, o algoritmo passa por um filtro de convexidade para eliminar pontos desnecessários e otimiza para os critérios de tempo e cobertura de um talude. Os resultados foram feitos em simulador Gazebo-ROS.

- "Coverage Path Planning Optimization Based on Point Cloud for Structural Inspection", Biundini, I. Z., Melo, A. G., Pinto, M. F. , Marcato, A. L. M., Honorio, L. M., Frontiers in Nature-Inspired Industrial Optimization. Springer, Singapore, 2022, 141-156.

O capítulo de livro apresenta o algoritmo de otimização baseado em morcegos para a otimização de trajetórias a partir de *point cloud*. A partir dos pontos, o algoritmo passa por um filtro de convexidade para eliminar pontos desnecessários e otimiza para os critérios de tempo e cobertura de um talude. Os resultados foram feitos em simulador Gazebo-ROS.

- "A Framework for Coverage Path Planning Optimization Based on Point Cloud for Structural Inspection", Biundini, I. Z., Melo, A. G., Pinto, M. F. , Marcato, A. L. M., Honorio, L. M., Sensors, Cham, 2021.

O artigo apresenta uma metodologia estrutural para otimização de trajetórias por cobertura baseado no algoritmo de morcegos a partir de *point cloud*. O artigo apresenta comparações de estratégias para realização do planejamento de trajetória, além da introdução de novos critérios fotométricos. Os resultados foram feitos em simulador Gazebo-ROS e no Phantom 4 Pro.



Além disso, foi submetido até o presente momento mais um artigo que está no processo de revisão principal pelos autores, listado abaixo:

- "Experimentation and simulation with autonomous coverage path planning for UAVs", Biundini, I. Z., Melo, A. G., Coelho, F. O., Marcato, A. L. M., Honorio, L. M., Pinto, M. F. *Journal of Intelligent Robotic Systems*, 2022.

O artigo explora a otimização baseada em meta-heurística presente nessa tese como um gerador de pontos de navegação que serão otimizados por métodos clássicos e propostos para geração dos melhores caminhos em relação a distância e custo computacional.

#### 1.4 Organização da Proposta da Tese

Além deste capítulo introdutório, essa tese de doutorado é dividida em mais quatro capítulos. No Capítulo 2, tem-se a revisão teórica das técnicas utilizadas neste trabalho. A Seção 2.3 apresenta uma breve revisão dos trabalhos relacionados, destacando o estado da arte em algoritmos de planejamento de cobertura.

O Capítulo 3 apresenta a metodologia proposta do algoritmo de planejamento de caminho por cobertura e suas fundamentações matemáticas.

No Capítulo 4 foram demonstrados os atuais resultados obtidos do algoritmo utilizando o simulador Gazebo-ROS, demonstrando a eficácia da metodologia proposta.

E por fim, no Capítulo 5, são realizadas as conclusões. Além disso, este capítulo apresenta as propostas futuras para a pesquisa.

## 2 Referencial Teórico

O capítulo 2 relata as definições necessárias para a implementação da metodologia proposta, sendo fundamentais para o entendimento da mesma. Na Seção 2.1 será esclarecida a base teórica do planejamento de caminho por cobertura. A Seção 2.2 apresentará as informações sobre as meta-heurísticas utilizadas. A Seção 2.3 apresentará alguns dos trabalhos correlacionados que abordam o problema de CPP.

### 2.1 Planejamento de Caminho por Cobertura

O CPP (Coverage Path Planning ou Planejamento de Caminho por Cobertura) pode ser resumidamente definido como um problema de determinar o melhor caminho que cubra todos os pontos de uma determinada área ou volume de interesse, evitando obstáculos. É importante ressaltar que existem diferenciações de algoritmos CPP para ambientes 2D que os impedem ser utilizados em ambientes 3D [Kapoutsis e Chatzichristofis 2017]. Contudo, diferentes estratégias foram propostas para lidar de forma adaptativa com as especificações referentes aos ambientes 2D e 3D. Como exemplo, a navegação autônoma teve um grande incentivo neste tipo de pesquisa [Buehler e Iagnemma 2009], onde foram apresentados artigos para o planejamento de trajetória destes veículos. Baras et al. [Baras e Dasygenis 2019] propõem um algoritmo *offline*, isto é, com informações completas do local de varredura, para resolver o desafio do CPP em ambientes 3D. Esse algoritmo é capaz de resolver o problema de CPP de vários robôs em ambientes bidimensionais e tridimensionais, adaptando o caminho de cada robô com base em suas especificações técnicas, como tipo e velocidade.

O planejamento do caminho de um local A a um local B, prevenção e reação a obstáculos, e às mudanças do ambiente são tarefas simples para os seres humanos devido ao seu alto grau de cognição. Já para robôs estas tarefas requerem algoritmos complexos e muitas vezes têm na absorção de informações alto grau de incerteza. Um robô usa sensores para perceber o ambiente e para construir ou atualizar seu mapa. Para determinar as ações de movimento apropriadas que levam ao local desejado, o mesmo pode usar diferentes algoritmos de decisão e

planejamento [Klancar et al. 2017]. Sendo assim, um correto planejamento de trajetória é de extrema importância para sistemas autônomos.

O planejamento de caminhos é usado para resolver problemas que variam desde o planejamento simples de rotas espaciais até a seleção de uma sequência de ações para atingir um determinado objetivo. Existem duas classificações para o ambiente das missões: dinâmico ou estático. O dinâmico apresenta mudanças em tempo real, o que inclui obstáculos em movimento, mudança de cores e iluminação, enquanto o estático não muda suas características com o tempo. Além disso, o ambiente pode ser classificado em relação ao conhecimento de seus obstáculos, chamado de conhecido ou nebuloso quando as informações são completas ou desconhecidas [Klancar et al. 2017]. O presente trabalho utiliza ambientes estáticos e nebulosos, já que ao inicializar o voo o ambiente não está totalmente identificado. Mesmo quando o ambiente é fornecido anteriormente as informações de pontos de interesse são desconhecidas, sendo identificadas enquanto ocorre a missão.

Geralmente, existem vários caminhos viáveis como solução do problema (ou seja, caminhos que não colidem com obstáculos). Para restringir a escolha, exigências ou critérios adicionais são introduzidos para definir a otimização desejada:

- o comprimento do caminho deve ser o menor;
- o caminho adequado é aquele em que o robô pode passar no menor tempo possível;
- o caminho deve estar o mais longe possível dos obstáculos;
- o caminho não deve possuir curvas fechadas;
- o caminho deve considerar restrições de movimento (por exemplo, onde no momento atual nem todas as direções são possíveis).

O planejamento de caminhos de obstáculos dinâmicos é um problema desafiador devido à dimensão adicional de tempo no espaço de pesquisa. Em abordagens que ignoram a dimensão do tempo e tratam as obstruções dinâmicas como estáticas, o replanejamento frequente é inevitável à medida que os obstáculos se movem. Neste caso, as soluções geralmente são sub-ideais e podem ser incompletas.

Como resultado, as trajetórias previstas estão sujeitas às alterações frequentes devido às previsões incorretas, o que torna necessário gerar novos planos em tempo hábil [Vemula e Muelling 2016].

Alguns algoritmos foram utilizados para a resolução de planejamento de caminhos, tais como Campos Potenciais [Orozco-Rosas e Montiel 2019], Amostragem Aleatória [LaValle 1998] [Lee e Pippin 2008], e métodos heurísticos como “D\*” [Koenig e Likhachev 2002] e “A\*” [Dai et al. 2019].

O problema do CPP está relacionado ao problema do caixeiro viajante de cobertura (do inglês, *covering salesman problem*), uma variante do caixeiro viajante [Pandiri e Singh 2020]. No entanto, neste problema, o agente deve cobrir todos os pontos na área de destino, visitando toda a região e capturando imagens de toda a superfície ao contrário de somente visitar uma posição em todos os bairros. Como o problema do caixeiro viajante é NP-hard, o tempo computacional necessário para solucioná-lo aumenta drasticamente quando a dimensão aumenta [Arkin e Fekete 2000]. Observe que o problema do caixeiro viajante não conta com obstáculos. De fato, mesmo o básico de planejamento de caminhos, conhecido como “problema do movedor de piano”, é demonstradamente NP-hard [LaValle 2006].

Uma solução de baixo custo computacional, e aplicável em certos cenários, é permitir que o robô se movimente de maneira aleatória pelo ambiente, abordagem na qual alguns robôs de limpeza de piso se baseiam. Essa solução mostra que se o chão for varrido de forma aleatória por tempo suficiente, ele deverá ser limpo. Exemplos de robôs de limpeza comerciais baseados totalmente ou em parte nessa estratégia são o RC3000 da Karcher, o Trilobite da Electrolux e o Roomba da iRobot [Palacín et al. 2005]. No entanto, para cobrir vastas áreas, e para operações de robótica aérea que lidam com um espaço tridimensional, é difícil pensar que um algoritmo ‘aleatório’ possa ser útil, onde o custo de operação do veículo (energia e tempo) seria inacessível. De forma a solucionar esse problema, vários métodos foram propostos.

A literatura de CPP propõe diversos algoritmos. As principais diferenciações do mesmo são: (i) Natureza da informação dos Algoritmos (*Online* ou *Offline*); (ii) Tipo de Divisão em que a superfície é submetida para a vistoria (Grade, Decomposição trapezoidal, Decomposição de Morse, entre outros); (iii) Tipo de

terreno a ser vistoriado (2D ou 3D); e (iv) Quantidade de robôs envolvidos, ou seja, presença de um ou mais robôs.

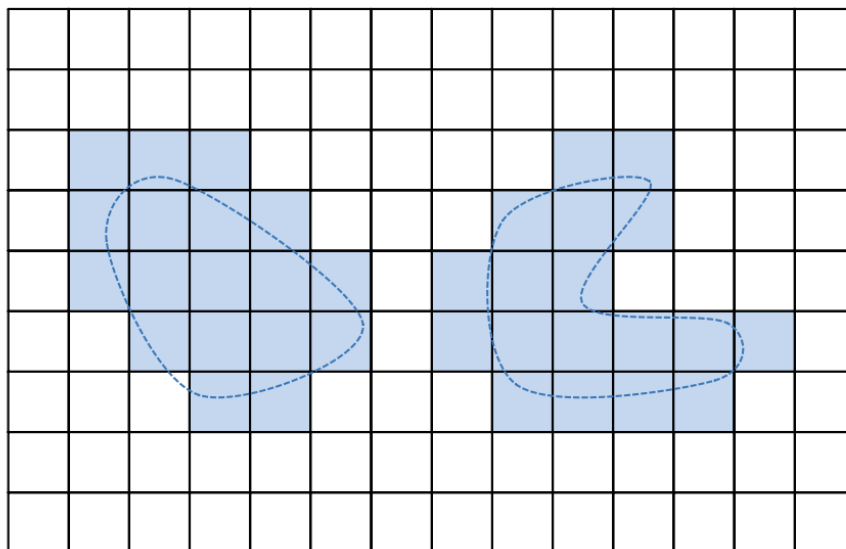
Algoritmos *offline* consideram que todo o terreno é previamente conhecido pelo robô e não mudará enquanto executa a missão. Porém, assumir que o conhecimento completo do ambiente não é possível para todos os cenários, principalmente em ambientes dinâmicos ou com a presença de mais equipamentos móveis na missão. Já os algoritmos *online* não assumem conhecimento completo do ambiente antecipadamente, permitindo que as mudanças do mesmo sejam detectadas durante o percurso do VANT, onde utilizam sensores em tempo real para capturar informações do ambiente e criar mapas próprios [Choset 2001]. Algoritmos *Online* não têm garantia de otimalidade global, utilizando das informações do ambiente em tempo real para projetar o caminho enquanto o executa. Desta forma, como não tem todas as características do mapa, o CPP *online* não pode garantir que o caminho gerado inicialmente será o melhor antes de observar todo o ambiente. Já algoritmos *offline* podem ou não ter otimalidade global, já que todas as informações são disponíveis, dependendo somente dos métodos utilizados para desenvolver o caminho.

Os métodos exatos de decomposição celular quebram o espaço livre em regiões simples e não sobrepostas chamadas de células. A união de todas as células preenchem exatamente o espaço livre. Essas regiões, onde não existem obstáculos, são fáceis de cobrir e podem ser varridas pelo robô usando movimentos simples, tais como “zigue-zague” [Galceran e Carreras 2013].

Os métodos baseados em grade usam uma representação do ambiente decomposto em uma coleção de divisões uniformes. Moravec e Elfes [Moravec e Elfes 1985] propuseram esse conceito para mapear um ambiente interno com um anel de sonar montado em um robô móvel. Nesta representação, cada célula possui um valor associado informando se um obstáculo está presente ou se é um espaço livre. O valor pode ser binário ou uma probabilidade. Normalmente, cada célula é um quadrado ou diferentes formas, como triângulos. Como resultado, estas técnicas são de “resolução completa”, ou seja, sua integridade depende da resolução do mapa. É fácil criar um mapa de grade, pois pode ser representado como uma matriz onde cada elemento contém informações de ocupação associadas a uma célula.

Como resultado, as representações baseadas em grade são as mais amplamente utilizadas para algoritmos de cobertura. No entanto, os mapas de grade sofrem com o crescimento exponencial do uso de memória, independentemente da complexidade do ambiente [Thrun 1998]. A Figura 1 apresenta um exemplo de mapa em grade (*grid map*).

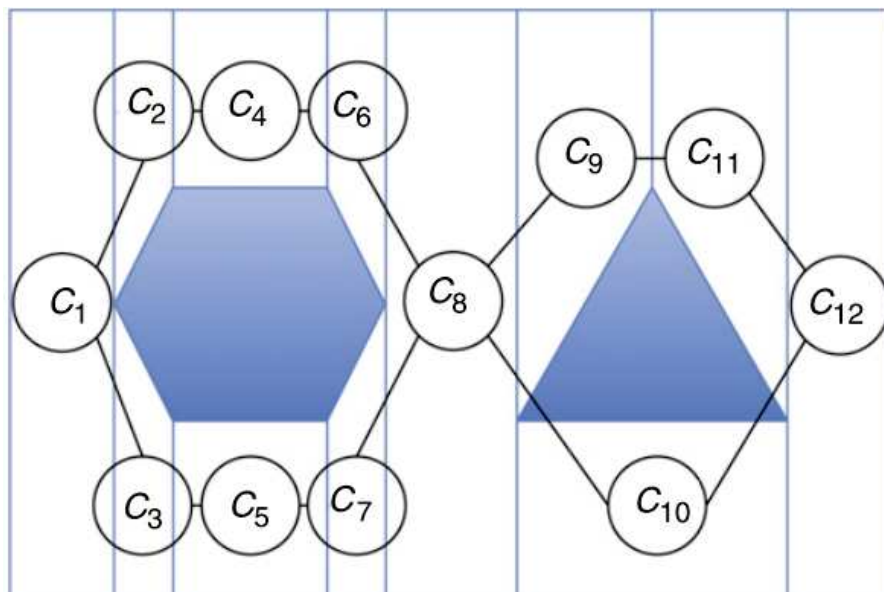
Figura 1 – Um exemplo de mapa de grade (grid map).



Fonte: Retirado de [Galceran e Carreras 2013].

Uma das técnicas mais simples de decomposição celular exata é a trapezoidal [Latombe 1990] [Choset et al. 2005]. Na decomposição trapezoidal, cada célula é um trapézio. Desse modo, utilizam-se movimentos simples de vai e vem para cobrir toda a região. A cobertura completa é garantida encontrando um passeio exaustivo pelo grafo de adjacência associado. Uma desvantagem é que ela gera muitas estruturas que podem ser fundidas para formar maiores. Isso é inconveniente, pois quanto mais células estarem presentes, mais longo o caminho de cobertura final se tornará. Isso acontece porque o método cria apenas células convexas. A Figura 2 apresenta um exemplo de mapa com decomposição trapezoidal.

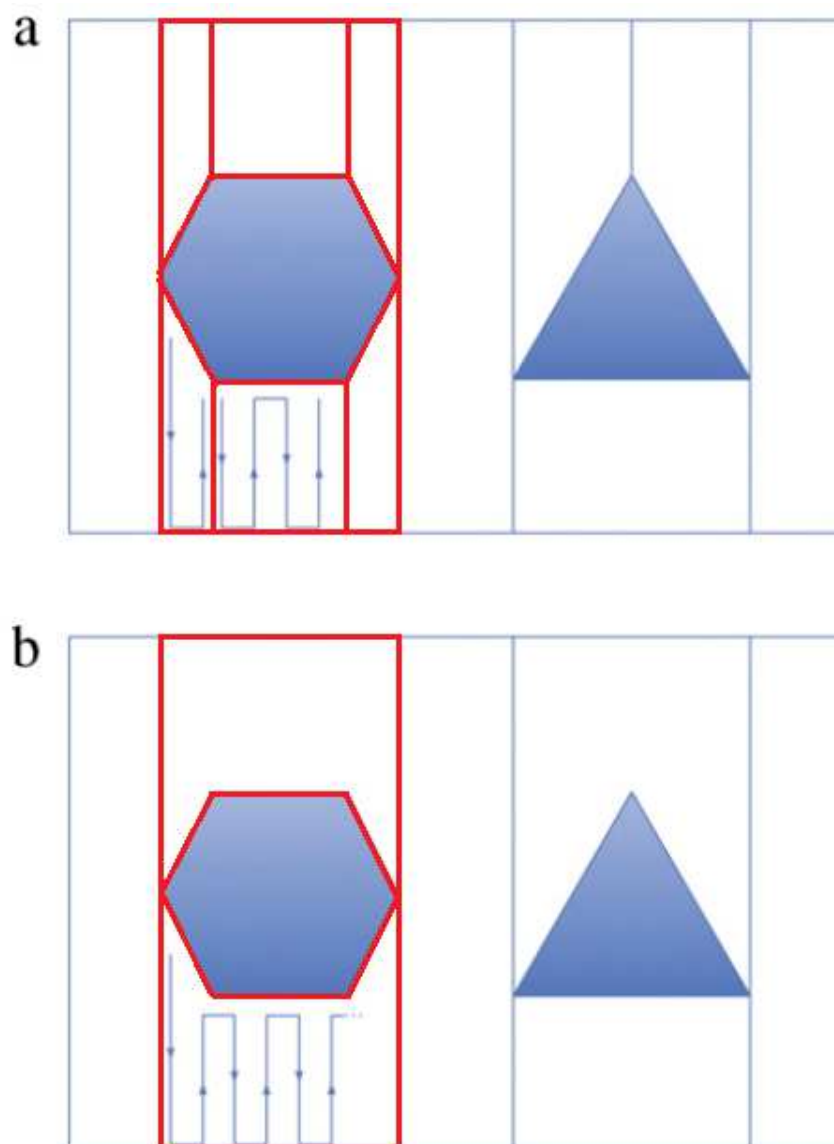
Figura 2 – Um exemplo de mapa trapezoidal.



Fonte: Retirado de [Galceran e Carreras 2013].

Choset e Pignon propuseram a decomposição celular do boustrophedon [Choset e Pignon 1998] [Choset et al. 2000]. O método é semelhante ao trapezoidal, mas considera vértices onde um segmento vertical pode ser estendido acima e abaixo. Os vértices onde isso ocorre são os pontos críticos. Ao aderir a essa estratégia, é possível reduzir o número de células criadas e obter caminhos mais curtos, resolvendo o problema da decomposição trapezoidal de criar células em excesso. Observe que, como a decomposição trapezoidal, esse método assume obstáculos poligonais e o terreno a ser conhecido *a priori* e, portanto, é classificado como *offline*. A palavra “boustrophedon” vem do grego antigo, e literalmente significa “o caminho do boi” significando o padrão em que um boi arrasta um arado para frente e para trás. A Figura 3 mostra a diferença de segmentação do trapezoidal (a) com a boustrophedon (b), destacada em vermelho. Percebe-se que a fundição de células pelo método de boustrophedon gera caminhos mais curtos. O método de boustrophedon produz 2 células destacadas em vermelho enquanto na mesma região o método trapezoidal produz 6 células.

Figura 3 – Um exemplo da diferença entre trapeizodal e boustrophedon.



Fonte: Editado de [Galceran e Carreras 2013].

Outros métodos baseados na divisão por grade ou boustrophedon foram desenvolvidos, tais como o método Morse [Acar et al. 2002] [Gao e Ding 2020] ou em pontos de referência [Wong e MacDonald 2004].

A maioria dos métodos de planejamento do caminho de cobertura pressupõe



que o ambiente possa ser modelado como uma superfície plana simples. Essa suposição é válida para limpeza de pisos, detecção de minas terrestres, roçadeira de grama, entre outros. No entanto, algumas regiões são tridimensionais e, em vez disso, é necessário um planejamento tridimensional da cobertura para cobrir essas superfícies. É o caso de um veículo subaquático autônomo de cobertura do mar ou de veículos de pintura por spray, por exemplo [Galceran e Carreras 2013].

Hert e Tiwari [Hert e Tiwari 1996] apresentaram um algoritmo de cobertura 3D baseado em um espaço planar bidimensional. Sua aplicação alvo é um veículo subaquático autônomo que imagina o fundo do mar. Sua solução é utilizada em um ambiente planar 3D projetivamente, aplicando o algoritmo de cobertura nos planos horizontais sucessivos, que se encontram em diferentes profundidades.

O trabalho proposto consiste em utilizar os conhecimentos de CPP para realizar missões com o intuito de captura de imagens para reconstrução 3D. Os requisitos definidos por Choset [Choset e Pignon 1998] são bases para o planejamento, considerando que são ambientes tridimensionais. Foi utilizada a divisão da superfície utilizando mapas de grade (*grid map*), onde cada célula é definida pelo tamanho do campo de visão da câmera.

## 2.2 Meta-heurística

Com a evolução tecnológica que impulsiona a tecnologia da informação, um grande número de problemas de otimização surge em vários campos, como engenharia, bioinformática, pesquisa operacional e geofísica, por exemplo. A maioria dos problemas de otimização foram classificados como NP-hard que tem o tempo de resolução exponencial em relação à grandeza do problema [Basset e Abdel 2018]. Assim, apenas um pequeno conjunto de problemas podem ser tratados usando métodos matemáticos exatos. Com isto, os pesquisadores pensaram em usar possíveis soluções alternativas (ou seja, métodos de aproximação) que podem encontrar uma solução boa o suficiente em um tempo razoável. Com tais métodos podendo ser categorizados em heurísticas e meta-heurísticas.

A diferença significativa entre as duas é que as heurísticas são mais dependentes do problema do que as meta-heurísticas. Em outras palavras, as heurísticas

podem ser aplicadas de forma eficiente a um problema específico, enquanto se tornam insuficientes para outros problemas. Por outro lado, a meta-heurística é um *framework* de algoritmo genérico ou um otimizador de caixa preta que pode ser aplicado a quase todos os problemas de otimização [Basset e Abdel 2018].

Fred Glover cunhou a palavra “meta-heurística” em 1986 [Glover 1986] para ilustrar o método heurístico sem característica específica do problema. A razão por trás do robusto mecanismo de busca da meta-heurística é a harmonização entre dois esquemas de busca: exploração global (diversificação) e exploração local (intensificação) [Blum e Roli 2003]. O primeiro é responsável pela exploração do espaço de busca, tentando descobrir novas regiões onde as respostas podem ser encontradas, enquanto o segundo busca a melhor solução nas áreas circundantes. Apesar capacidade da meta-heurística em resolver um determinado problema, ela não pode resolver todos os problemas de otimização, e todas as meta-heurísticas têm o mesmo desempenho em média ( Teorema do “sem almoço grátis” ou *no free lunch* (NFL) [Wolpert e Macready 1997] [Kuyu e Vatansever 2021]). Este foi um ímpeto para o desenvolvimento de um grande número de meta-heurísticas para adaptação a vários tipos de problemas, tais como contínuos, discretos, irrestritos e multi-objetivos.

Além disso, as meta-heurísticas têm características fundamentais que podem ser resumidas como:

- As meta-heurísticas não são para um problema específico;
- Meta-heurísticas geralmente são aproximadas;
- A exploração da meta-heurísticas sobre o espaço busca encontrar uma solução “boa” ou “suficiente”;
- Meta-heurísticas essencialmente podem ser descritas por nível de abstração;
- Meta-heurísticas geralmente permitem uma implementação paralela fácil;
- As meta-heurísticas vão desde a pesquisa local básica até técnicas de aprendizado avançadas;

- Meta-heurísticas podem incorporar vários mecanismos para evitar convergência prematura.

O Algoritmo 1 mostra uma estrutura abstrata para as meta-heurísticas.

---

**Algorithm 1:** Pseudocódigo de uma meta-heurística.

---

```

input   : Criar uma ou mais soluções iniciais;
Output : Melhor Indivíduo Encontrado  $x_*$ 

while Critério de parada não for satisfeito do
  if Exploração Local then
    | Cria uma nova solução pela exploração local;
  else
    | Cria uma nova solução pela exploração global;
    | Atualiza a melhor solução encontrada.

```

---

Fonte: Elaborado pelo autor, tendo como base [Basset e Abdel 2018]

Uma possível classificação das meta-heurísticas é sobre qual área da natureza ela é inspirada. A Seção 2.2.1 apresenta os algoritmos baseados em biologia, que exploram a movimentação de animais em procura de abrigo ou alimento e na evolução das espécies. A Seção 2.2.2 apresenta os algoritmos baseados em comportamento físico, tais como recozimento de metais.

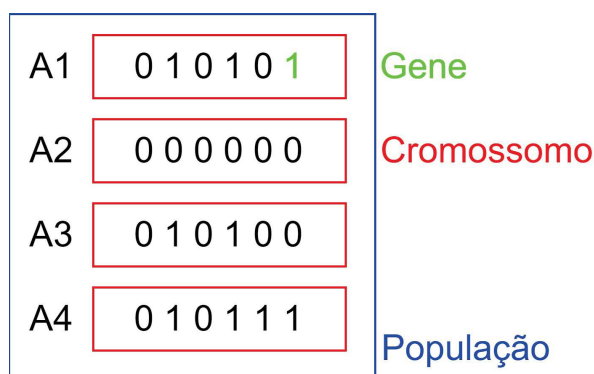
### 2.2.1 Métodos baseados em biologia

A maioria das meta-heurísticas é baseada em princípios de evolução biológica. Em particular, elas estão preocupadas em simular várias metáforas biológicas que diferem na natureza dos esquemas de representação de cada indivíduo e de como ocorre a interação com o meio. Existem três paradigmas principais: sistemas evolucionários, enxame e imunológico.

Algoritmos evolutivos simulam a progressão biológica da evolução no nível celular, empregando operadores de seleção, cruzamento e mutação para gerar soluções candidatas cada vez melhores (cromossomos). Para a computação evolutiva, existem quatro paradigmas históricos: programação evolutiva [Fogel 1998], estratégias evolutivas [Rechenberg 1973], algoritmos genéticos [John 1992] e programação genética [Koza 1990].

O princípio de Darwin “Sobrevivência do mais apto” é o ponto de partida na introdução do mecanismo da evolução biológica. Os algoritmos evolutivos emulam o processo de evolução biológica dos cromossomos, definindo os seguintes operadores: seleção, cruzamento e mutação. Os cromossomos, como mostrado na Figura 4, são tratados como soluções candidatas para um determinado problema e são avaliados de acordo com sua adequação ao problema. A seleção de pais para reprodução é um processo influente para a geração de novas soluções. Existem diferentes esquemas de seleção, como a seleção da roleta (*roulette wheel selection* ou RWS), a seleção de torneio (*tournament selection* ou TOS) e a seleção de classificação linear (*linear rank selection* ou LRS), entre outros. A Figura 4 também mostra as definições de genes, que são partes características do cromossomo e a definição de população, que é um conjunto de indivíduos definidos cada um por seu cromossomo.

Figura 4 – Estrutura de População do Algoritmo Genético.



Fonte: Elaborado pelo autor.

Durante a fase de *crossover*, algumas partes de dois cromossomos selecionados são trocadas. Os genes podem ser trocados de maneiras diferentes, como *crossover* uniforme, *crossover* de dois pontos, *crossover* aritmético, entre outros [Katoch e Chauhan 2021]. Na Mutação, algumas partes dos cromossomos são alteradas aleatoriamente para escapar dos ótimos locais. A escolha adequada do método de *crossover* e mutação depende da representação do indivíduo. O Algoritmo 2 mostra o pseudocódigo de um algoritmo evolutivo.

---

**Algorithm 2:** Pseudocódigo do Algoritmo Evolutivo.
 

---

**input** :

- Função de *Fitness*:  $f(x)$ ,  $x = (x_1, x_2, \dots, x_n)$ ;
- População inicial de indivíduos  $x_i$  ( $i = 1, 2, \dots, n$ ) = Pop

**Output** : Melhor Indivíduo Encontrado  $x_*$

**while** *Critério de parada não for satisfeito do*

**for**  $i \leftarrow 1$  **to**  $n$  **do**

└ Calcule o *fitness* de  $x_i$

**while**  $len(Selecionados) < m$  **do**

└ Selecione um indivíduo (Guloso ou Proporcional ao *fitness*);

**while** (*Filhos*)  $< n$  **do**

Seleciona dois ou mais pais entre os indivíduos selecionados anteriormente;

**if**  $rand < C$  **then**

└ Realiza o *crossover* entre eles gerando um Filho;

**if**  $rand < M$  **then**

└ Realiza a Mutação do Filho Criado;

└ Transforma Filhos em População;

---

Fonte: Elaborado pelo autor, tendo como base [Basset e Abdel 2018]

Além dos algoritmos clássicos, outros algoritmos podem ser classificados como evolucionários, tais como: *Flower Pollination Algorithm* [Yang 2012] e *Coral Reefs Optimization* [Nguyen et al. 2019].

A Otimização do Enxame de Partículas (PSO) foi inspirada no comportamento dos animais na sociedade, como cardume de peixes ou bando de pássaros. Pode ser considerado como um algoritmo de inteligência semi-evolutivo, ou seja, como um algoritmo evolucionário. No PSO, uma população de soluções é amostrada aleatoriamente e avaliada para determinar a melhor solução e repetir os mesmos processos por um número predefinido de vezes. Ao contrário de um algoritmo evolucionário, cada solução candidata possui uma velocidade e uma memória de suas posições, podendo cada uma se locomover em direção a um objetivo mais

próximo ou a um líder, que é classificado como melhor indivíduo do grupo. Dessa forma, cada indivíduo é um animal num espaço  $n$ -dimensional que está à procura de sobreviver no meio, tendo sua posição e velocidade atualizadas.

Entre estes algoritmos estão o *Bees Algorithm* [Pham 2015], *Cat Swarm Optimization* [Chu e Tsai 2006], *Ant Colony Optimization* [Dorigo e Birattari 2006], *Firefly Algorithm* [Łukasik e Żak 2009], *Bat Algorithm* [Yang e Karamanoglu 2012] [Yang e Gandomi 2012], *Whale Optimization Algorithm* [Mirjalili e Lewis 2016], *Bird Swarm Algorithm* [Meng et al. 2016], *Hunger Games Search* [Yang et al. 2021], entre outros.

O Algoritmo 3 mostra um pseudoalgoritmo genérico para os algoritmos baseados em população.

---

**Algorithm 3:** Pseudocódigo do Algoritmo populacional.

---

**input** :

- Função de *Fitness*:  $f(x)$ ,  $x = (x_1, x_2, \dots, x_n)$ ;
- Posição inicial de indivíduos  $x_i$  ( $i = 1, 2, \dots, n$ ) = Pop
- Velocidade inicial dos indivíduos  $v_i$  ( $i = 1, 2, \dots, n$ ).

**Output** : Melhor Indivíduo Encontrado  $x_*$

**while** *Critério de parada não for satisfeito* **do**

**for** *Cada indivíduo* **do**

└ Calcule o *fitness* de  $x_i$

**if** *O valor de fitness é melhor que o do melhor indivíduo* **then**

└ Atualiza o melhor indivíduo para o de maior *fitness*.

**for** *Cada indivíduo* **do**

└ Calcula a velocidade do indivíduo e atualiza a posição.

---

Fonte: Elaborado pelo autor, tendo como base [Basset e Abdel 2018]

Bean [Bean 1960] introduziu pela primeira vez a teoria da Seleção Clonal em 1960 para discutir a resposta básica do sistema imune adaptativo (linfócitos) ao estímulo antigênico. O autor confirmou que apenas as células capazes de identificar um antígeno proliferariam, enquanto aquelas que não identificam um

antígeno seriam selecionadas e eliminadas. CLONALG é a versão de otimização dos paradigmas de Seleção Clonal, pois o sistema imunológico natural pode ter objetivos múltiplos ou contraditórios e não tem razão para desenvolver uma resposta ótima. Em seguida, os anticorpos de maior afinidade são clonados para gerar mais anticorpos contra o antígeno. Esses anticorpos não clonados são substituídos por novos. Para adquirir imunidade, as melhores soluções são guardadas em uma célula de memória.

O Algoritmo 4 mostra um pseudoalgoritmo genérico para os algoritmos baseados em população.

---

**Algorithm 4:** Pseudocódigo do Algoritmo Imunológico.

---

**input** :

- Função de *Fitness*:  $f(x)$ ,  $x = (x_1, x_2, \dots, x_n)$ ;
- População de indivíduos  $x_i$  ( $i = 1, 2, \dots, n$ ) = Pop

**Output** : Melhor Indivíduo Encontrado  $x_*$

**while** *Critério de parada não for satisfeito* **do**

Avalia cada indivíduo; Seleciona os melhores anticorpos; Clona os melhores; Aplica mutação nos melhores; Substitui os anticorpos que não foram clonados por novos;

---

Fonte: Elaborado pelo autor, tendo como base [Basset e Abdel 2018]

### 2.2.2 Métodos baseados em física

Simulated Annealing (SA) foi proposto por Metropolis [Metropolis et al. 1953] para simular o resfriamento do material em um banho de calor. Depois disso, Kirkpatrick et al. [Kirkpatrick e Gelatt 1983] aplicaram a ideia aos problemas de otimização, apesar de a palavra “meta-heurística” não ser conhecida naquela época. SA derivou seu nome do processo de recozimento físico. Assim, como o estado inicial do recozimento, o algoritmo é tolerante e pode mudar para uma solução pior. Em cada iteração, o algoritmo se torna mais rigoroso para obter uma solução melhor em cada etapa, ou seja, SA começa com uma solução inicial aleatória de alta temperatura. Então, outra solução é criada aleatoriamente perto da solução

inicial e se calcula a diferença entre as aptidões da solução inicial e da solução criada. Se a energia for reduzida, a nova solução é criada e o processo de busca continua, buscando um ponto onde a energia é menor. Caso a energia aumente, o ponto é aceito com certa probabilidade.

Outros métodos foram baseados em relações físicas, tais como *Gravitational Search Algorithm* [Rashedi e Nezamabadi-Pour 2009], inspirado nas leis de Newton de gravidade e movimento. *Electromagnetic Field Optimization* [Abedinpourshotorban et al. 2016] baseou nas relações de campos magnéticos, enquanto *Nuclear Reaction Optimization* [Wei et al. 2019] se baseou no processo de reações nucleares para o processo de otimização.

### 2.2.3 Métodos baseados em outras ciências

Além da biologia e da física, as meta-heurísticas foram baseadas nas mais diversas ciências da natureza, sociais e até nas artes.

Lam propôs *Chemical Reaction Optimization* (CRO) [Lam e Li 2009], um algoritmo que simula o comportamento de interações de moléculas em uma reação química para atingir um estado estável de baixa energia. No CRO, a molécula representa uma solução candidata e cada molécula possui dois tipos de energia: Energia Potencial que representa a função objetivo e Energia Cinética que representa a tolerância de mudar para uma pior estrutura. Baseados em princípios químicos tem-se a *Gases Brownian Motion Optimization* (GBMO) [Abdechiri e Meybodi 2013] retirado do movimento de partículas de fluidos em suspensão. No GBMO, cada solução candidata é apresentada por uma molécula que possui quatro especificações: posição, massa, velocidade e raio de turbulência, onde as moléculas se movimentam em direção ao objetivo respeitando o movimento Browniano e avaliado de acordo com sua posição.

Em relação às ciências sociais, existem as otimizações baseadas em iterações esportivas e culturais do ser humano. *Teaching-Learning-Based Optimization* (TLBO) [95] simula a influência dos professores sobre os alunos em um processo de aprendizagem da escola clássica, ou seja, o professor (a melhor solução) compartilha seu conhecimento com os alunos (população de soluções) e seu efeito de qualidade de ensino nas notas do aluno (valores de aptidão). Outro método é o *League*



*Championship Algorithm* (LCA), que emula um campeonato esportivo em que várias equipes jogam em uma liga por várias temporadas, ou seja, como um campeonato esportivo real, várias equipes (soluções) participam de uma liga (população de soluções) e competem em pares. Em seguida, a correspondência é analisada por análises de forças, fraquezas, oportunidades e ameaças. A equipe vencedora é determinada de acordo com a boa formulação dos jogadores da equipe e a força de jogo (valor de *fitness*). Com isso, cada equipe melhorará seu desempenho, mudando seu estilo de jogo e reformulando seus jogadores.

Sobre as artes têm-se métodos baseados na criação de músicas [Mora-Gutiérrez e Ramírez-Rodríguez 2014] e na procura de melhores harmônicos musicais [Geem e Kim 2001].

Outros métodos não bio-inspirados também são classificados como meta-heurísticas. Como exemplo é a busca Tabu [Glover e McMillan 1986], onde a ideia principal é a proibição (ou “tabu”) que uma região já visitada seja avaliada novamente com o objetivo de aumentar a diversificação, ou seja, a fim de explorar o espaço de solução e evitar ficar preso em ótimos locais.

Percebe-se que a pesquisa sobre meta-heurísticas abrange diversos campos de investigação, sendo cada método baseado em um comportamento tendo melhores resultados em alguns problemas, mesmo que o teorema do “sem almoço grátis” mostre que todas têm o mesmo desempenho em média. Dessa forma, na Seção 4.1.2, utilizando a biblioteca em Python *The state-of-the-art MEta-heuristics ALgorithms in PYthon* (MEALPY) [Thieu 2020], que tem diversas meta-heurísticas já implementadas e disponíveis com licença MIT, iremos comparar alguns métodos para o problema de CPP utilizando os critérios dos objetivos de tempo e fotometria, além do custo computacional.

### 2.3 Trabalhos Correlatos

Zelinsky et al. [Zelinsky et al. 1993] apresentou o primeiro algoritmo baseado em grade para CPP. Note que a decomposição em grade é bem simples para se aplicar, sendo rápida e eficiente. Em regiões onde não existem obstáculos côncavos ou com tamanhos proporcionais à dimensão da divisão, a decomposição

tem resultados significativos [Galceran e Carreras 2013].

Hert et al. [Hert e Tiwari 1996] apresentou um método de cobertura 3D baseado em um algoritmo planar bidimensional. Sua aplicação alvo é um veículo subaquático autônomo que imagina o fundo do mar. Sua solução se aplica a um ambiente 3D, aplicando o algoritmo de cobertura de terreno planar nos planos horizontais sucessivos, que se encontram em diferentes profundidades.

Butler et al. [Butler e Rizzi 1999] propôs um algoritmo exato de decomposição celular para robôs com sensor de contato (isto é, robôs sem capacidade de detecção de alcance) cobrindo ambientes retilíneos desconhecidos *online*. Sua aplicação motivadora para a cobertura de ambientes retilíneos é a calibração de um sistema de montagem automatizado no qual motores lineares planares operam em superfícies em forma de mesa para transferir produtos através de uma fábrica. Dessa forma, o algoritmo continua sendo *online*, porém com o local sendo identificado somente em pequenas distâncias do robô.

Atkar et al. [Atkar et al. 2001] considerou a geração de trajetória para robôs de pintura. Em seus primeiros trabalhos, os autores propuseram um CPP online, tridimensional para superfícies fechadas. O método estendeu as decomposições de Morse a espaços não-planos. No entanto, obstáculos no ambiente do alvo não foram considerados neste trabalho.

Acar et al. [Acar et al. 2002] generalizou a decomposição dos boustrophedon propondo uma nova abordagem baseada em pontos críticos das funções de Morse. A decomposição baseada em Morse tem a vantagem de lidar também com obstáculos não poligonais.

Gabriely e Rimon [Gabriely e Rimon 2002] propuseram o algoritmo Spiral-STC (Spanning Tree Coverage), uma abordagem *online* que consiste em subdividir a área em um mapa de grade e seguir um caminho espiral sistemático. Esse caminho espiral é gerado seguindo uma árvore de abrangência que o robô constrói usando seus sensores internos. O robô é capaz de cobrir todas as células da grade uma vez e percorrer uma trajetória completa de cobertura.

Wong e MacDonald [Wong e MacDonald 2003] apresentaram um algoritmo de cobertura topológica *online* para robôs móveis com base na detecção de referên-

cias naturais. Este trabalho é destinado a ambientes planares simples. A introdução de métodos online tem como objetivo analisar todo o ambiente enquanto gera os caminhos.

Galceran e Carreras [Galceran e Carreras 2012] apresentaram um método de CPP baseado na decomposição de Morse para minimizar a cobertura redundante de veículos subaquáticos autônomos.

Kapoutsis et al. [Kapoutsis e Chatzichristofis 2017] dividiu o terreno 2D em retângulos de tamanho fixo para o planejamento do caminho. O objetivo era minimizar a comprimento do trajeto e o tempo de vistoria dos veículos terrestres. O algoritmo permite a cobertura de áreas utilizando um único robô, assim como por múltiplos, e usa a *Minimum Spanning Tree* (MST) como técnica de otimização e algoritmo offline.

Ellefsen et al. [Ellefsen e Lepikson 2017] propôs um algoritmo CPP para a vigilância de estruturas subaquáticas tridimensionais, como poços de petróleo. A técnica de otimização é uma Otimização Evolutiva multi-objetivo (MEOP), com o objetivo de maximizar a cobertura de veículos subaquáticos autônomos (AUV) e minimizar o número de colisões. É uma técnica que necessita da informação completa do ambiente (offline).

Em Azpurua et al. [Azpurua et al. 2018], os autores propuseram um método para analisar ambientes de exploração mineral, geologia, arqueologia e oceanografia. O autor usou planos de cobertura de terrenos bidimensionais para otimizar o número de robôs e ângulos para aumentar a qualidade das reconstruções. A técnica consiste em segregar o ambiente em células hexagonais, que são agrupadas e alocadas a diferentes equipes de robôs.

Vamprala et al. [Vamprala e Saripalli 2018] realizou a geração de mapas melhorados a partir de informações parciais de mapas para localização de robôs. Sua proposta traz um algoritmo que recebe mapas tridimensionais incompletos ou desatualizados e os atualiza utilizando a próxima melhor visualização que maximiza as informações previstas. Dessa forma, o objetivo é melhorar a densidade de informações dos mapas, além da acurácia da localização dos robôs nos mapas gerados.

Em Theile et al. [Theile et al. 2020], os autores usaram o aprendizado por reforço para otimizar o processo de planejamento do caminho a fim de reduzir os custos de energia observando a vida útil da bateria e o aumento da área de cobertura. O método usa canais semelhantes a mapas para alimentar informações espaciais em camadas de rede convolucionais e aprender uma política de controle que generaliza as posições iniciais e as restrições de energia e decide entre várias posições de pouso.

Lakshmanan et al. [Lakshmanan et al. 2020] propôs um CPP para robôs de revestimento usando uma Rede Neural Convolucional (CNN) para atravessar um ambiente bidimensional que pode determinar uma trajetória eficiente em tempo real. Apesar de ser um algoritmo de tempo real, é necessário que o mapa seja previamente conhecido, sendo classificado como *offline*. O objetivo é minimizar os custos do robô, reduzindo o tempo e a energia consumida por ele.

Choi et al. [Choi e Lee 2021] apresentou um algoritmo para mapeamento tridimensional em uma área usando vetores de superfície. Este método usa  $k$ -vizinhos mais próximos ( $k$ -NN) e um mapa de ocupação para cobrir toda a região e apresentar um modelo do ambiente tridimensional. É um algoritmo em que as informações do ambiente são adquiridas por meio de sensores enquanto o VANT voa, sendo um algoritmo online.

As Tabelas 1 e 2 apresenta o resumo das informações dos trabalhos correlatos descritas anteriormente, sendo a primeira sobre as informações de informações do ambiente e decomposição, enquanto a segunda se refere aos objetivos explorados pelas metodologias. De acordo com a tabela, a metodologia proposta é classificada como *online*, pois o ambiente e os objetivos são identificados por sensores do mesmo para ambientes 3D, e usando decomposição por grade. Note que a principal diferença entre a proposta deste trabalho e os trabalhos correlatos se encontra no fato da aplicação de multi-objetivos conflitantes, tempo e fotometria, no planejamento de caminhos, adicionando replanejamento e missões locais para melhorar a qualidade da reconstrução 3D. Além disso, a captura de informações do ambiente pode ser a partir de sensores ou de *point clouds* previamente criados.

Tabela 1 – Informações dos Trabalhos Correlatos - On/Offline, Ambiente e decomposição.

<b>Referência</b>	<b>On/ Offline</b>	<b>Amb.</b>	<b>Decomposição</b>
Zelinsky et al. [Zelinsky et al. 1993]	Offline	2D	Grade
Hert et al. [Hert e Tiwari 1996]	Online	2.5D	Grade
Choset et al. [Choset e Pignon 1998]	Offline	2D	Boustrophedon
Butler et al. [Butler e Rizzi 1999]	Online	2D	Grade
Choset et al. [Choset et al. 2000]	Offline	2D	Boustrophedon
Atkar et al. [Atkar et al. 2001]	Online	3D	Grade
Acar et al. [Acar et al. 2002]	Online	2D	Decomposição de Morse
Gabriely et al. [Gabriely e Rimon 2002]	Online	2D	Grade
Wong et al. [Wong e MacDonald 2003]	Online	2D	Obstáculos Planares
Choset et al. [Choset et al. 2005]	Offline	2D	Trapezoidal
Galceran et al. [Galceran e Carreras 2012]	Online	3D	Decomposição de Morse
Ellefsen et al. [Ellefsen e Lepikson 2017]	Offline	3D	Decomposição em planos circulares
Azpúrua et al. [Azpúrua et al. 2018]	Offline	2D	Grade
Vamprala et al. [Vemprala e Saripalli 2018]	Online	3D	Decomposição por Elipsoides
Baras et al. [Baras e Dasygenis 2019]	Online	2D/3D	Grade
Song et al. [Song et al. 2019]	Online	3D	Grade
Jinh et al. [Jing et al. 2019]	Online	3D	Grade
Theile et al. [Theile et al. 2020]	Offline	3D	Grade
Lakshmanan et al. [Lakshmanan et al. 2020]	Offline	2D	Grade
Choi et al. [Choi e Lee 2021]	Online	3D	Grade
Metodologia Proposta	Online	3D	Grade

Tabela 2 – Informações dos Trabalhos Correlatos - Objetivos: Tempo e Fotometria.

<b>Referência</b>	<b>Tempo</b>	<b>Fotometria</b>
Zelinsky et al. [Zelinsky et al. 1993]	Sim	Não
Hert et al. [Hert e Tiwari 1996]	Não	Não
Choset et al. [Choset e Pignon 1998]	Não	Não
Butler et al. [Butler e Rizzi 1999]	Não	Não
Choset et al. [Choset et al. 2000]	Não	Não
Atkar et al. [Atkar et al. 2001]	Não	Não
Acar et al. [Acar et al. 2002]	Não	Não
Gabriely et al. [Gabriely e Rimon 2002]	Não	Não
Wong et al. [Wong e MacDonald 2003]	Sim	Sim
Choset et al. [Choset et al. 2005]	Não	Não
Galceran et al. [Galceran e Carreras 2012]	Não	Não
Ellefsen et al. [Ellefsen e Lepikson 2017]	Sim	Não
Azpúrua et al. [Azpúrua et al. 2018]	Não	Sim
Vamprala et al. [Vemprala e Saripalli 2018]	Sim	Não
Baras et al. [Baras e Dasygenis 2019]	Sim	Não
Song et al. [Song et al. 2019]	Não	Não
Jinh et al. [Jing et al. 2019]	Sim	Não
Theile et al. [Theile et al. 2020]	Não	Não
Lakshmanan et al. [Lakshmanan et al. 2020]	Sim	Não
Choi et al. [Choi e Lee 2021]	Não	Não
Metodologia Proposta	Sim	Sim

### 2.3.1 Conclusões Parciais

O presente capítulo teve como objetivo demonstrar o referencial teórico que será utilizado na metodologia. A Seção 2.1 trouxe uma definição para o objetivo deste documento sobre CPP, a escolha por uma decomposição em grade e as qualificações que o método proposto tem que atender. A Seção 2.2 apresenta diversas meta-heurísticas utilizadas nesse documento e suas respectivas características. Posteriormente, serão comparadas para o problema de CPP. Por fim, na Seção 2.3, adicionou uma amostragem dos algoritmos de CPP, para demonstrar os métodos de decomposição, os ambientes que estão sendo desenvolvido e a natureza (*online* ou *offline*) de cada algoritmo, colocando a metodologia proposta dentro dessas classificações.

Algoritmos de CPP foram escolhidos por suas características de cobertura por completo de uma superfície, com requisitos definidos e uma literatura abrangente para criar uma base teórica para o planejamento. A utilização das Meta-heurísticas é justificada pela necessidade de abrangência aos mapas propostos, que mudam, dependendo da superfície de exploração, além capacidade de abranger aspectos do CPP e de segurança da missão junto com os critérios fotométricos que são essenciais para boas reconstruções tridimensionais.

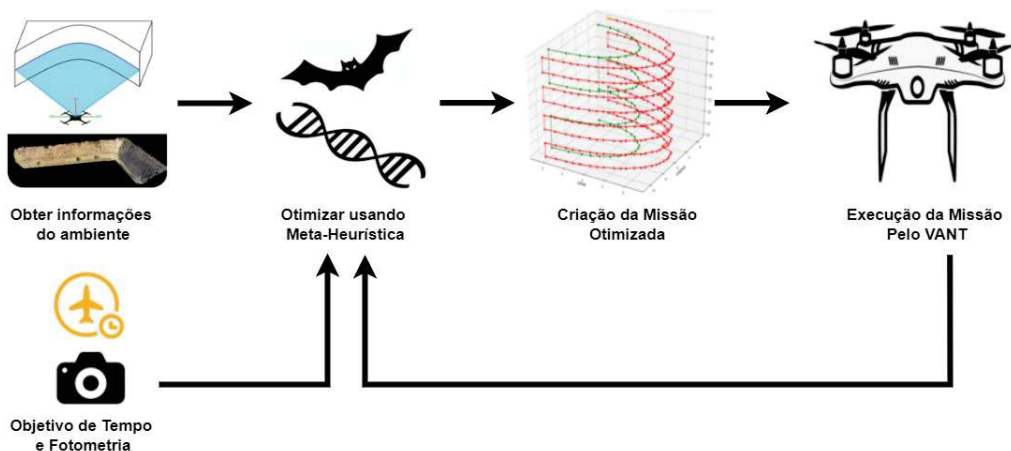
De acordo com a amostragem dos algoritmos de CPP a expectativa é que esse trabalho adicione numa literatura já vasta, que se preocupa com a cobertura completa da superfície e a locomoção em ambientes com muitos obstáculos, a necessidade de aspectos fotométricos para reconstruções tridimensionais, principalmente de grandes superfícies. A disputa entre os objetivos de tempo e de imagem gera caminhos que visam ambos os objetivos e a presença de mini missões com identificação de pontos de interesse ajudam a varrer grandes superfícies, sem a perda de qualidade em locais importantes.

### 3 Metodologia Proposta

O principal objetivo desta pesquisa é fornecer um método para otimizar trajetórias utilizando conceitos de CPP. Essas trajetórias são realizadas por VANTs que visam capturar imagens usadas posteriormente para uma reconstrução 3D. Portanto, é necessário que o mesmo seja capaz de obter fotos suficientes com uma sobreposição adequada para gerar os dados para a composição da nuvem de pontos densas [Dai e Feng 2014]. Neste processo devem ser atendidos alguns requisitos como otimizar parâmetros de voo, uso de energia, segurança de manobras e tempo de voo. A Figura 5 apresenta uma visão geral das etapas da metodologia proposta. A primeira etapa é a captura de informações do ambiente, podendo ser através de sensores no VANT, reconstruções 3D previamente realizadas, ou a partir de mapas de elevação digitais ( *digital elevation model* ou DEM). A partir dessas informações, tem-se a otimização utilizando meta-heurística tentando equilibrar os objetivos de tempo de missão e fotometria. A otimização cria uma missão com pontos equidistantes à superfície, para manter a qualidade em toda a reconstrução 3D, que é executada pelo VANT. Durante a execução da missão tem-se a realização da otimização em certos períodos de tempo para readequar a missão a problemas que podem acontecer durante a execução, tais como atrasos de captura na imagem, variação da velocidade média, desvio de obstáculos não planejados e as missões locais de aproximação, este último problema destacado na Seção 3.4.



Figura 5 – Resumo da Metodologia.



Fonte: Elaborado pelo autor.

Inicialmente, existe a captura das informações da superfície a ser analisada através de sensores no VANT, como o *LaserScan*, com objetivo de conhecer a região a ser vistoriada (Figura 6 - Etapa 1). Essa etapa também pode receber informações da região através de nuvem de pontos ou de DEM evitando a utilizando do *LaserScan*. Em seguida, os dados com a posição da superfície são processados em um algoritmo de otimização meta-heurístico que determinará o caminho que cobre todos os locais que precisarão ser inspecionados. Esse algoritmo otimizará uma função multi-objetivo envolvendo variáveis como tempo de inspeção, área de cobertura, entre outras (Figura 6 - Etapa 2).

A saída é uma missão com um conjunto de *waypoints* e orientações que serão executados para realizar inspeções, dividido em segmentos horizontais que seguem o formato da superfície com alturas diferentes. Com os dados fornecidos pelo CPP, é realizado o início da missão pelo VANT onde o mesmo se locomoverá para o primeiro *waypoint* fornecido (Figura 6 - Etapa 3).

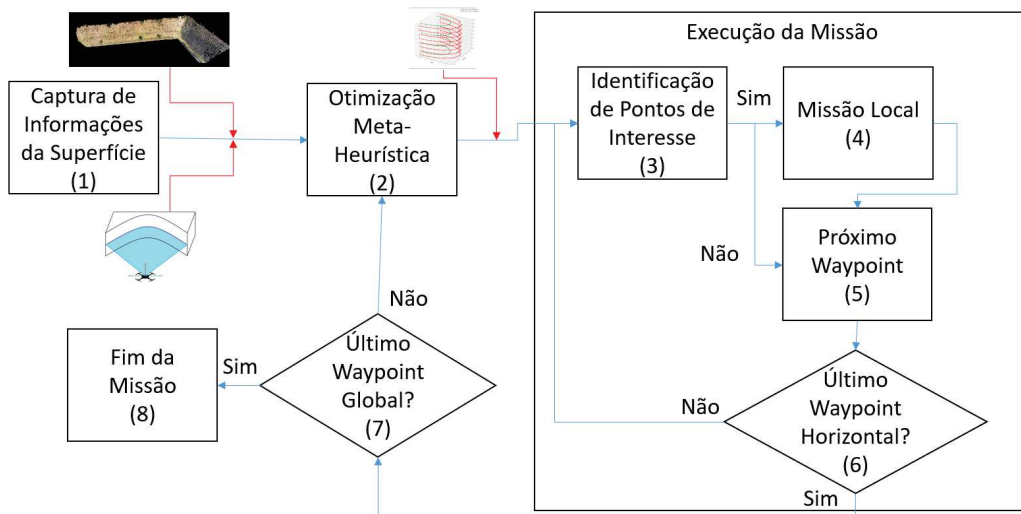
Ao chegar no *waypoint*, será utilizado o algoritmo de identificação de objetivos dinâmicos (Figura 6 - Etapa 3). A identificação tem como propósito localizar pontos de interesse e realizar missões locais para aumentar a qualidade das imagens obtidas. Se o ponto for identificado, será necessário executar uma submissão de

aproximação (Figura 6 - Etapa 4). Após essas verificações, e até mesmo a missão local de aproximação, o VANT parte para o próximo *waypoint* (Figura 6 - Etapa 5), reiniciando o processo de verificação.

Ao chegar no *waypoint*, é verificado se este é o ponto final do translado horizontal (Figura 6 - Etapa 6). Caso sim, verifica-se se é o último ponto da missão (Figura 6 - Etapa 7), e o VANT retorna à posição de decolagem, finalizando o algoritmo (Figura 6 - Etapa 8). Se não for o *waypoint* final global, o algoritmo calcula um novo CPP com o tempo disponível e a área que necessita ser vistoriada (Figura 6 - Etapa 2).

As Seções 3.1 (Etapa 1), 3.2 (Etapa 2), 3.3 (Etapa 6) e 3.4 (Etapa 7) apresentam detalhes sobre a captura de informações do ambiente, o planejamento do caminho, identificação de objetivos dinâmicos e as missões de aproximação, respectivamente.

Figura 6 – Metodologia do Algoritmo de Planejamento de Caminho por Cobertura.



Fonte: Elaborado pelo autor.

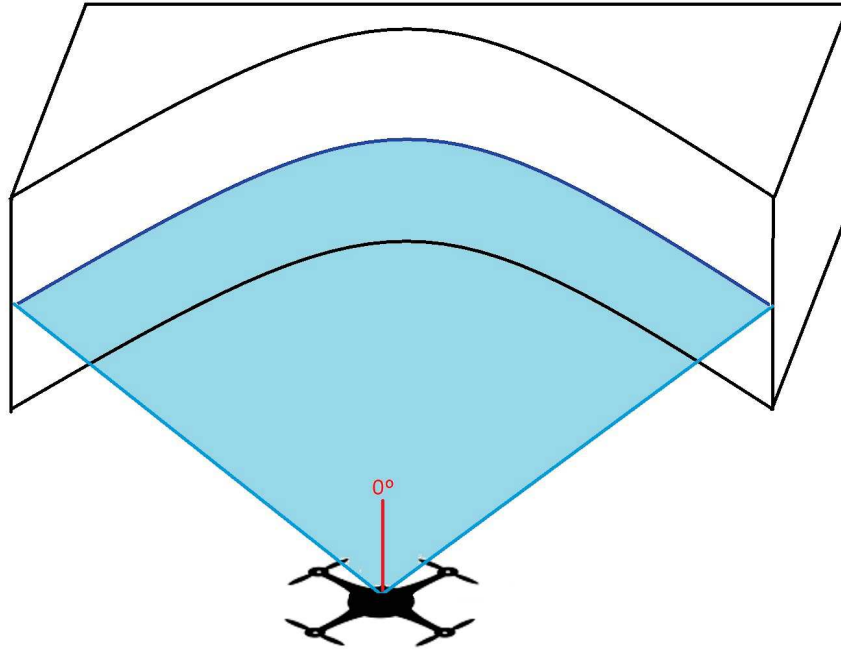
### 3.1 Captura de Informações do Ambiente

A metodologia proposta tem como objetivo realizar o CPP sem que tenha informações previamente fornecidas da superfície de forma *offline*, partindo de uma

proposta de abordagem *online* baseada em sensor. Desta forma, é necessário que o VANT consiga receber informações do ambiente em que ele está incluído, assim como a realização da identificação do ambiente que deseja realizar a cobertura.

A captura do formato da superfície em voos tridimensionais ocorreu através da leitura de um *LaserScan*. A escolha pelo *LaserScan* se deu por ser uma tecnologia de medição com alta precisão que permite executar levantamentos tridimensionais/bidimensionais. Além disso, o laser é um sensor ativo que tem bons resultados para ambientes iluminados ou sem a presença de luz. Inicialmente ocorre a decolagem do VANT (*Takeoff* em Inglês) para uma altura de 1 metro, permitindo segurança na captura para que a mesma não capture informações do solo. Após a decolagem, inicia a aquisição de todos os sensores do laser do VANT, como ilustrado na Figura 7. O *LaserScan* captura as distâncias,  $d_{\text{laser}}$  e o respectivo ângulo de leitura,  $\theta_{\text{laser}}$ . O intervalo de aquisição para o talude na captura do laser variou entre 2 e 20 metros. A distância do VANT em relação à superfície na hora da leitura é de acordo com a área a ser vistoriada e a capacidade do equipamento em realizar as medições. Para superfícies maiores, é necessário um sensor com capacidade superior e uma distância maior a superfície para que a mesma seja identificada em sua totalidade. O tempo que o VANT deve ficar estacionado varia em relação a cada equipamento de leitura. No caso, foi escolhido o tempo de 3 leituras completas armazenando todos os dados.

Figura 7 – Funcionamento do *LaserScan* e orientação do VANT.



Fonte: Elaborado pelo autor.

A partir da posição original do VANT,  $Pos_{VANT-x}$ ,  $Pos_{VANT-y}$ ,  $z_{VANT-z}$  e  $\theta_{VANT-Yaw}$ , respectivamente, e a partir da a posição nos eixos  $x$ ,  $y$  e  $z$  e ângulo  $Yaw$  do VANT, e das leituras do *LaserScan*, é possível calcular a posição desses pontos pelas Equações 3.1 e 3.2.

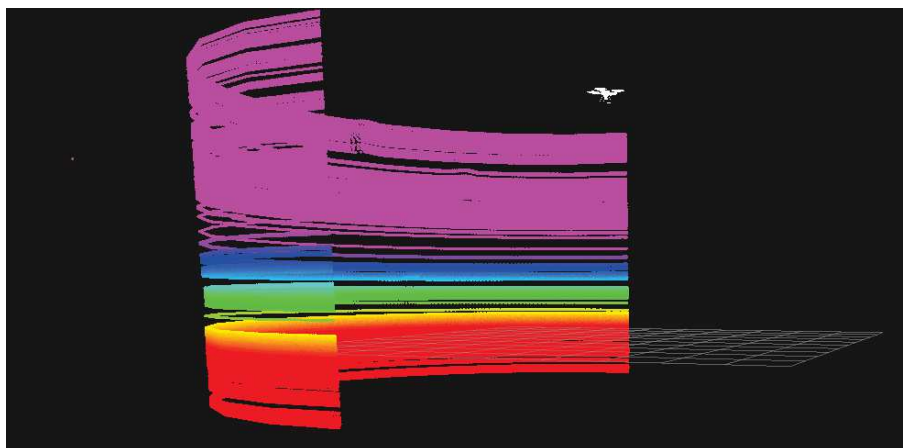
$$X_{Scan} = Pos_{VANT-x} + d_{laser} \cdot \cos(\theta_{laser} + \theta_{VANT-Yaw}) \quad (3.1)$$

$$Y_{Scan} = Pos_{VANT-y} + d_{laser} \cdot \sen(\theta_{laser} + \theta_{VANT-Yaw}) \quad (3.2)$$

Dessa forma, o vetor  $X_{Scan}$  e  $Y_{Scan}$  caracterizam a posição dos pontos nos eixos  $x$  e  $y$  que a superfície se encontra. Deve ser observado que a altura é a mesma que a do VANT durante a aquisição. A partir dessas capturas ocorre o incremento da altitude até que os sensores não captem mais leituras do laser. Isso ocorre devido ao fato de que sem as leituras é identificado que a superfície não

se encontra à frente do equipamento, e portanto, rastreado em sua totalidade. A Figura 8 mostra o resultado deste procedimento em um simulador Gazebo-ROS [Koenig e Hsu 2013]. Percebe-se que todo o talude foi vistoriado, sendo possível identificar além do formato da estrutura quanto a altura da mesma,  $h_{\max}$ .

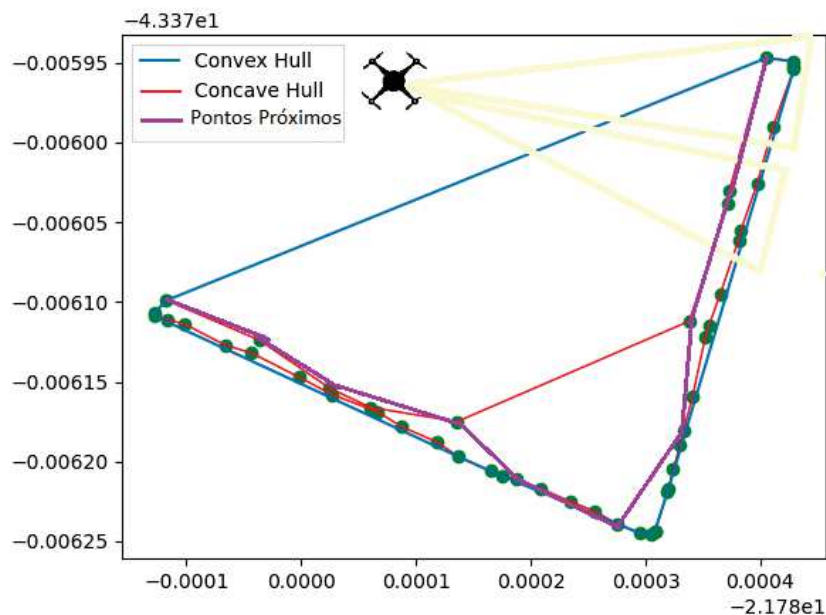
Figura 8 – Resultado Final do Procedimento de Captura de informações do Ambiente em Simulador Gazebo-ROS.



Fonte: Elaborado pelo autor.

Existe a necessidade desses pontos serem filtrados, removendo *outliers* e diminuindo os pontos que serão entregues aos algoritmos de otimização. Alguns algoritmos para esta tarefa são o *Convex Hull* e o *Concave Hull*. O casco convexo de objetos geométricos é o menor conjunto convexo que contém os objetos. Existem algoritmos para pontos na literatura em dois, três e até mesmo espaços euclidianos de dimensão superior [Nguyen et al. 2019]. O casco côncavo é um algoritmo que encontra um objeto côncavo que envolve todos os pontos, usando métodos como “*Alpha Shapes*” ou algoritmo K-vizinho mais próximo [Kalinina et al. 2018]. A Figura 9 mostra os dois algoritmos em pontos de uma reconstrução, observando apenas uma altura de toda a reconstrução. Os pontos são verdes, com a linha azul formada pelo casco convexo e a linha vermelha do casco côncavo. Embora o casco côncavo tenha melhores resultados na representação da superfície, o tempo computacional do algoritmo é maior que o do casco convexo. Outra possibilidade, a partir da posição do VANT no momento de captura, é a escolha dos pontos mais

Figura 9 – Comparação entre algoritmo Concave hull, Convex Hull e pontos próximos.

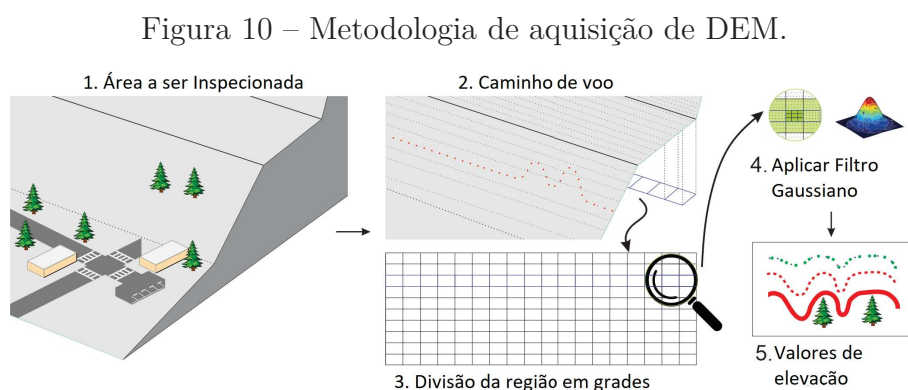


Fonte: Elaborado pelo autor.

próximos em ângulos pré-definidos como ilustrado em amarelo, criando o caminho em roxo. A distância utilizada é a euclidiana. Em relação à segurança da missão, o melhor método é o *Concave Hull*, pois apresentou uma superfície que tem todos os pontos fora da superfície, porém o tempo computacional para a aquisição dessa região foram todos maiores que 15 minutos, impossibilitando a aplicação para uma proposta online já que o número de faixas de leitura são sempre maiores que cinco. Já utilizando o *Convex Hull* é necessário remover a conexão entre o primeiro e último pontos para que a superfície não seja identificada somente como uma reta e regiões atrás da superfície, o que pode ocasionar colisões por informações erradas. Dessa forma foi utilizado o método de pontos próximos, com ângulos de  $5^\circ$  a  $10^\circ$ , variando o número de pontos que definem a superfície 36 a 18 em cada altura.

Outra forma de analisar é a partir de mapas digitais de elevação (*digital elevation models* ou DEM) [Zhang e Montgomery 1994] quando o objetivo é a cobertura em voos acima da região. O processo de geração de DEM pretende ser

rápido, produzindo uma boa representação de superfície e sendo usado em casos online. A Figura 10 mostra uma representação do processo aplicado. Observe na Figura 10 (1.) uma região exemplo para ser inspecionada. A região é simplificada como mostrada na Figura 10 (2.) onde as superfícies mais complexas são descartadas e somente o solo e grandes objetos são destacados. Uma vez que o processo de reconstrução geralmente executa algumas etapas de filtragem e o *octomap* produz dados de entrada razoavelmente limpos no processo online. A partir deste ponto, um agrupamento de grade é executado conforme representado na Figura 10 (3.). Em cada divisão, um filtro de kernel gaussiano [Martino e Read 2021] é aplicado nos valores de altura média do espaço, como exemplificado na Figura 10 (4.). Todos os pontos de informação de altura são combinados em uma única informação de altura. Isso resulta em uma matriz de altura de saída que representa a área de superfície suavizada, conforme representado na Figura 10 (5.). Inicialmente pode-se não ter informação nenhuma do ambiente e o mapa pode ser criado a partir de leituras de sensores como laser e câmeras estéreo enquanto executa o voo.



Fonte: Elaborado pelo autor.

### 3.2 Planejamento do Caminhos

A partir dos pontos recolhidos pelo procedimento de captura de informações, detalhado na Seção 3.1, é necessário criar um caminho para que o VANT realize a missão de capturar imagens da superfície objetivo que possam realizar reconstruções 3D com qualidade.

Como descrito na Seção 2.1, o planejamento de caminho é bastante complexo computacionalmente. Como o planejamento é realizado *online*, o tempo de processamento deve ser o menor possível para evitar desperdício de baterias do VANT esperando resultado para começar sua missão de varredura.

Descritos na Seção 2.2, algoritmos meta-heurísticos permitem o controle do tempo de execução através da escolha do número de gerações e a população de indivíduos. Isto possibilita respostas em tempo controlável a problemas complexos, ainda que não possuam garantia de otimalidade global. Isto é, a resposta encontrada pelos algoritmos meta-heurísticos não tem garantia de ser o ótimo global do problema de otimização. Apesar disso, o resultado obtido pode ajudar a encontrar soluções para que o VANT realize sua missão, possibilitando captura de imagens para reconstruções 3D com boas características de resolução. O algoritmo proposto tem identificação dinâmica de objetivos sobre a superfície, variando o voo de acordo com que este vai sendo realizado, sendo assim considerado um CPP *online*.

Um aspecto essencial do algoritmo CPP é a sobreposição entre imagens. Esse espaço coincidente é um requisito, uma vez que um ponto deve aparecer em pelo menos três imagens para que sua posição seja calculada adequadamente. Existem dois tipos de sobreposição neste caso: (i) horizontais e (ii) verticais. O movimento horizontal é definido através dos pontos escolhidos pela etapa de identificação. A distância mínima entre dois pontos é definida como ( $D_{\min}$ ). O tamanho de cada imagem depende da distância do VANT à superfície ( $Dist_H$ ) e do ângulo de abertura da câmera ( $\theta_{hor}$ ), conforme mostrado na Equação 3.3, em que a distância horizontal à superfície é definida por ( $Dist_{hor}$ ).

$$\tan\left(\frac{\theta_{hor}}{2}\right) = \frac{Dist_{hor}}{2Dist_H} \quad (3.3)$$

A taxa de cobertura horizontal ( $Cobertura_{hor}$ ) é fornecida pela razão entre  $Dist_{hor}$  e  $D_{\min}$ , conforme apresentado na Equação 3.4.

$$Cobertura_{hor} = 1 - \frac{D_{\min}}{Dist_{hor}} \quad (3.4)$$

A dimensão vertical ( $Dist_{vert}$ ) de cada imagem também depende da distância do VANT à superfície ( $Dist_H$ ) e do ângulo de abertura da câmera ( $\theta_{vert}$ ). A equação



3.5 mostra essa dependência.

$$\tan\left(\frac{\theta_{\text{vert}}}{2}\right) = \frac{Dist_{\text{vert}}}{2Dist_{\text{H}}} \quad (3.5)$$

A Equação 3.6 mostra que o deslocamento vertical ( $Offset_{\text{vert}}$ ) depende da quantidade de *waypoints* na missão vertical ( $Num_{\text{way}}$ ), altura mínima ( $h_{\text{min}}$ ) e altura máxima ( $h_{\text{max}}$ ).

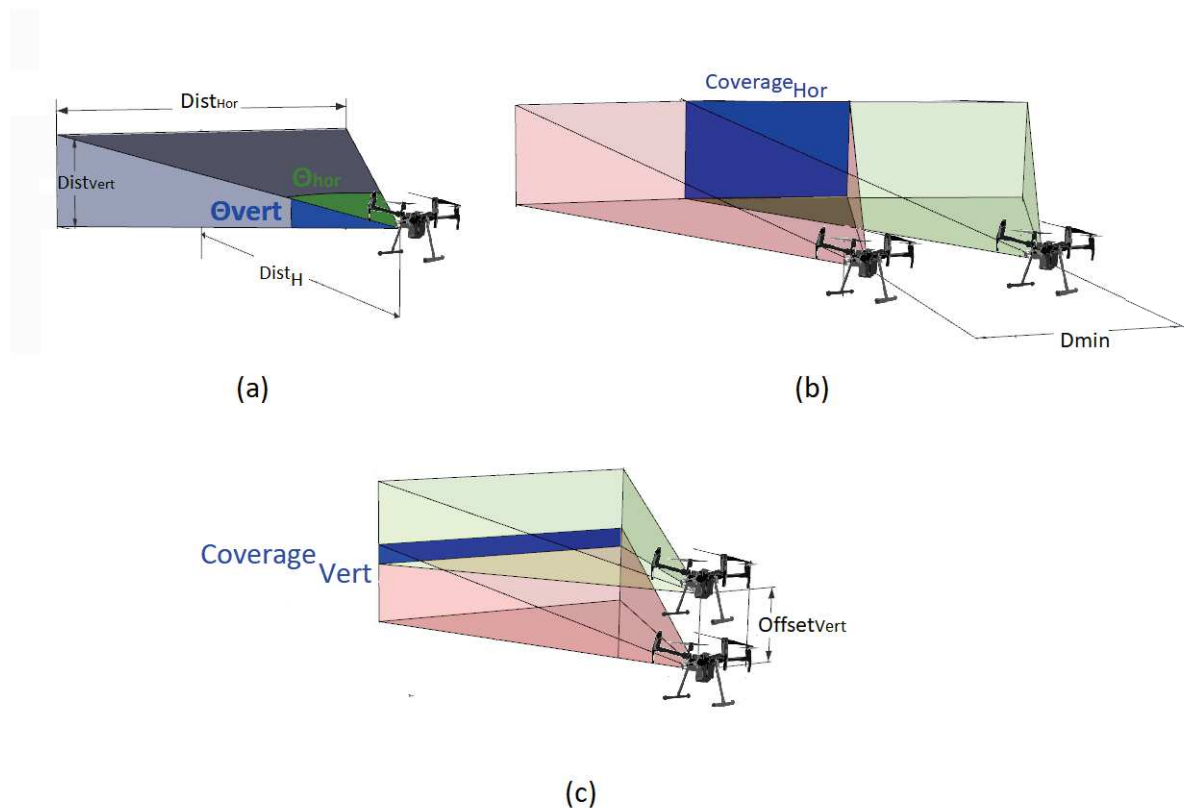
$$Offset_{\text{vert}} = \frac{|h_{\text{max}} - h_{\text{min}}|}{(Num_{\text{way}} + 1)} \quad (3.6)$$

A taxa de cobertura vertical ( $Cobertura_{\text{vert}}$ ) é fornecida pela razão entre  $Dist_{\text{ver}}$  e  $Offset_{\text{vert}}$  como na Equação 3.7.

$$Cobertura_{\text{vert}} = 1 - \frac{Offset_{\text{vert}}}{Dist_{\text{hor}}} \quad (3.7)$$

A Figura 11 ilustra o esquema do problema. A mesma demonstra todos os valores usados nas equações. A Figura 11 (a) mostra as distâncias verticais ( $Dist_{\text{vert}}$ ) e horizontais ( $Dist_{\text{hor}}$ ), além dos respectivos ângulos de visão dos equipamentos ( $\theta_{\text{vert}}$  e  $\theta_{\text{hor}}$ ) junto com a distância da superfície ( $Dist_{\text{H}}$ ). Figuras 11 (b) e 11 (c) demonstram as coberturas horizontal e vertical, onde cada VANT representa o equipamento em diferentes momentos respectivos.

Figura 11 – Esquemático do Problema.



Fonte: Elaborado pelo autor.

Considerando  $D_{\min}$ , o objetivo da cobertura tenderá a diminuir esse valor para os *waypoints* de fotografia, ou seja, para os *waypoints* onde o VANT irá parar e tirar foto, com área de sobreposição maior. No entanto, essa diminuição gera muitos pontos de fotografia, e em cada uma, existe a desaceleração do VANT para captura da imagem, aumentando o tempo geral da missão. Assim, indo contra o objetivo de minimização do tempo. Observando  $Dist_H$ , o objetivo de cobertura tende a aumentar a distância para que a taxa de cobertura aumente, diminuindo o tempo da missão. Porém, a qualidade dos pontos capturados diminui, criando nuvens mais esparsas. Dessa forma, foi escolhido um multiplicador  $G_{\text{dist}} \in [0, 1]$  que escala o *fitness* proporcionalmente à distância do talude, sendo 1 para 1 metro e 0 para 20 metros, permitindo que missões mais próximas sejam escolhidas. No caso

de ( $Num_{way}$ ), a meta de cobertura aumentará o número de *waypoints* verticais, aumentando então o tempo total de voo do VANT, já que o número total de *waypoints* é o produto dos pontos horizontais com os dos verticais, além que cada *waypoint* vertical gerará um translado horizontal, aumentando o caminho do VANT. Esses três valores serão otimizados para o planejamento do caminho de cobertura do VANT.

Para ambas as meta-heurísticas, foi necessário definir estruturas que representam o problema, sendo chamados de indivíduos para o algoritmo genético, e morcegos para o algoritmo de morcegos. A Equação 3.8 mostra o vetor de três posições que são a estrutura do problema.

$$[D_{\min}, Dist_H, Num_{way}] \quad (3.8)$$

onde  $D_{\min} \in \mathbb{R}^+$ ,  $Dist_H \in \mathbb{R}^+$  e  $Num_{way} \in \mathbb{N}$ .

A partir dessas três variáveis e dos pontos recolhidos pela etapa de identificação, é possível criar missões com características de forma semelhantes à superfície. Inicialmente, é necessário achar os pontos com distância  $Dist_H$  dos pontos recolhidos pela etapa de reconhecimento. Para isso, foram calculadas retas paralelas aos pontos com distância  $Dist_H$ . Considere como exemplo os dois primeiros pontos como  $lat_1, lon_1, lat_2, lon_2$  para suas respectivas latitudes e longitudes.

Dada a equação da reta, como a Equação 3.9, o ângulo da reta (a) pode ser calculado pela Equação 3.10 e o coeficiente linear (b) pela Equação 3.11. Para evitar erros computacionais referentes a pontos na mesma latitude e longitude, o que ocasionária um erro matemático de divisão por zero, um valor baixo é adicionado ao denominador.

$$Lon = a \cdot lat = b \quad (3.9)$$

$$a = \frac{lon_1 - lon_2}{lat_1 - lat_2} \quad (3.10)$$

$$b = lon_1 - a_1 \quad (3.11)$$

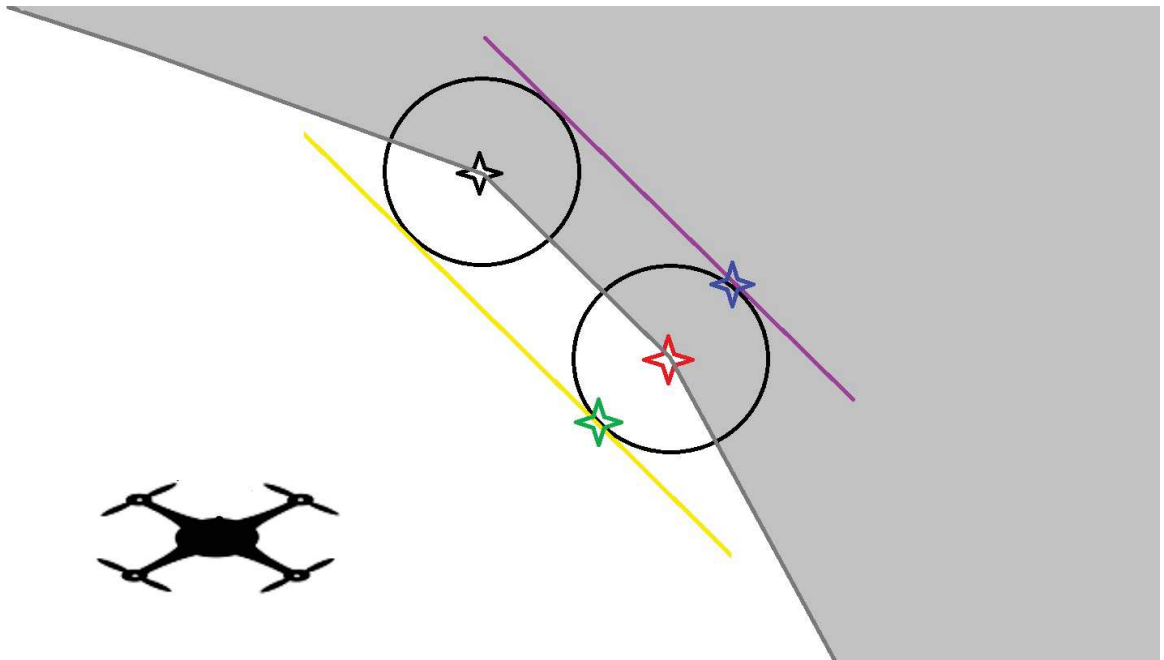
Retas paralelas possuem o mesmo coeficiente angular, sendo necessário calcular o coeficiente linear para encontrá-las. No caso, existem duas retas paralelas que satisfazem o problema. Nesse caso, o problema de otimização da Equação 3.12 significa achar um valor de  $x$  que representa os dois fatores lineares das retas, aqui chamados de  $b_1$  e  $b_2$ .

$$\underset{x}{\text{minimizar}} \quad Dist_H \pm \frac{x - b}{\sqrt{(a^2 + 1)}} \quad (3.12)$$

A partir dos valores encontrados na Equação 3.12, é necessário encontrar pontos com distância  $Dist_H$  do ponto  $[lat, lon]$  que estejam sobre essas retas. O problema de otimização da Equação 3.13, para o coeficiente  $b_1$ , realiza esse cálculo. O mesmo deve ser feito para o coeficiente  $b_2$ . O ponto com distância euclidiana mais próxima do VANT será o escolhido, pois este terá sua posição fora da superfície. A Figura 12 mostra o esquemático dessas equações. Percebe-se que as duas posições originais em preto e vermelho geram retas paralelas, em amarelo e roxo. Existem dois pontos, azul e verde, que satisfazem as condições de estarem a uma distância  $Dist_H$  do original. Porém, somente o verde está fora da superfície, indicado pelo cinza claro, sendo este com distância euclidiana menor ao VANT que o ponto azul.  $x$  e  $y$  são as posições de latitude e longitude, respectivamente, com distância  $Dist_H$  aos pontos  $[lat, lon]$ .

$$\begin{aligned} \underset{x,y}{\text{minimizar}} \quad & |Dist_H - \sqrt{(x - lat)^2 + (y - lon)^2}| \\ \text{sujeito a:} \quad & y - a \cdot x - b_1 = 0 \\ & -180 \leq x, y \leq 180 \end{aligned} \quad (3.13)$$

Figura 12 – Esquemático do Cálculo de Pontos Paralelos com Distância  $Dist_H$ .



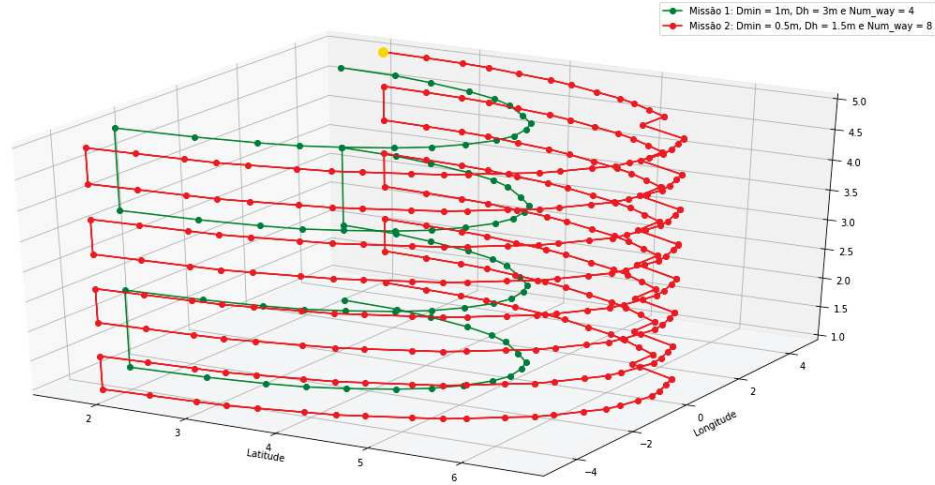
Fonte: Elaborado pelo autor.

A partir dessa estratégia, é possível gerar cenários distintos para cada valor da estrutura, como visto na Figura 13. A primeira missão, em verde, é com  $D_{\min} = 1$  m,  $Dist_H = 3$  m e  $Num_{way} = 4$ . Enquanto a segunda, em vermelho, com valores de  $D_{\min} = 0.5$  m,  $Dist_H = 1.5$  m e  $Num_{way} = 8$ . Em ambos os casos, as missões possuem formatos próximos à superfície.

Dado o primeiro ângulo de leitura do laser, pode-se considerar que o mesmo está perpendicular com a superfície. Dessa forma, uma maneira de calcular os ângulos de Yaw que o VANT deve executar para que o mesmo mantenha o movimento perpendicular pode ser calculado pela Equação 3.14.

$$\theta_{YAW-i} = \theta_{YAWi-1} + atan2(lat2 - lat1, lon2 - lon1) \quad (3.14)$$

Figura 13 – Exemplos de Missões desenvolvidas para duas respostas da Meta-heurística.



Fonte: Elaborado pelo autor.

A partir das missões criadas usando as estruturas e com a velocidade média do VANT,  $vel_{VANT}$ , é possível calcular o tempo estimado que a missão irá ser executada,  $T_{missão}$ , demonstrada na Equação 3.15.  $T_{shot}$  é o tempo que o VANT fica parado para captura das imagens. Com as taxas de coberturas estimadas pelas Equações 3.4 e 3.7 é possível calcular os *fitness* de cada estrutura.

$$T_{Missão} = \frac{Distância_{hor} \cdot Num_{way} + |h_{max} - h_{min}|}{vel_{UAV}} + T_{shot} N_{Waypoints} \quad (3.15)$$

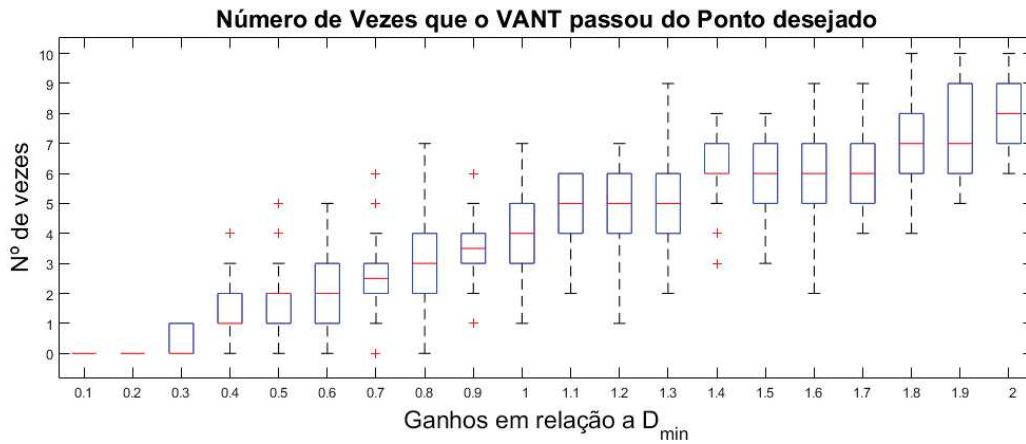
A velocidade média do VANT,  $vel_{VANT}$ , pode ser uma entrada fixa para o processo de otimização, onde o mesmo executa o processo de aceleração, permanece nessa velocidade e no final desacelera. Porém, em ambientes maiores a captura de imagens acontece em posições mais espaçadas do que em ambientes menores sendo que manter a velocidade entre os pontos pode ocasionar desperdício de energia do VANT quando a mesma é baixa ou gastos excessivos na aceleração e desaceleração quando é alta. Em casos onde a velocidade é muito alta pode acontecer até mesmo sobreposição entre os caminhos executados pelo VANT, pois o mesmo terá que

executar retornos, pois ultrapassou a posição do *waypoint* especificado por não conseguir desacelerar. Este efeito contraria os critérios de CPP. Dessa forma existe uma relação entre  $vel_{VANT}$  e  $D_{min}$  como mostrado na Equação 3.16 para permitir que em missões mais espaçadas a velocidade entre os pontos seja maior. Existe também um limitante superior de 6 m/s para evitar acidentes em que o controlador não consiga reassumir o controle do VANT antes de uma colisão.

$$vel_{VANT} = 0,275 \cdot D_{min} \quad (3.16)$$

O valor constante da Equação 3.16 foi obtido através de resultados de simulação apresentados na Figura 14. Para o teste, realizado no simulador Gazebo-ROS, foram utilizadas distâncias de 0.5 m até 3 m, variando de 0.5 m em 0.5 m, para  $D_{min}$  e ganhos variando de 0.1 até 2 para a constante. Para cada valor de ganho e de  $D_{min}$  foi criada uma missão em linha reta com 10 pontos e foi executada uma aceleração e um desaceleração para que a imagem seja capturada nos pontos desejados. Caso o VANT não consiga desacelerar, passe do ponto e tenha que retornar, será considerado um ponto que o VANT passou do ponto desejado. Percebe-se que o VANT passou do ponto nenhuma vez com ganhos abaixo de 0,3, enquanto no valor de 0,3 tivemos algumas missões que tiveram 1 ponto que descumpriu os critérios definidos por Cao et. all [Cao e Huang 1988]. Para ganhos maiores tivemos um aumento do número de vezes que a missão teve que retornar ao *waypoint*, sendo que com ganho máximo temos quase todos os pontos com esse problema, apesar de aumentar a velocidade média e diminuir o tempo máximo de missão.

Figura 14 – Teste para análise do Ganho de velocidade máxima.



A função de *fitness* de cada estrutura é dada pela soma de três funções, conforme definido nas Equações 3.17 e 3.18 e mostrada na Equação 3.19. Quanto menor o tempo de missão, maior será a qualidade de cada resposta. Note também que quanto maior a taxa de cobertura, vertical ou horizontal, o mesmo efeito também será visto. Para equilibrar os objetivos também temos os ganhos de tempo e de fotometria, que podem tender a missão para cada objetivo proposto. Para os testes ambos os ganhos são unitários.

$$Fitness_{Tempo} = \frac{-10T_{Missão}}{Tempo_{máximo}} + 10 \quad (3.17)$$

$$Fitness_{Cobertura} = \frac{5Cobertura_{em\%}}{100} \quad (3.18)$$

$$Fitness_{Total} = G_{Tempo} \cdot Fitness_{Tempo} + G_{Foto} \cdot (Fitness_{Cobertura - Horizontal} + Fitness_{Cobertura - Horizontal}) \quad (3.19)$$

Essas funções têm como objetivo representar requisitos do problema de otimização, que são diminuir a duração da missão e aumentar a cobertura vertical ou horizontal. Como são alvos conflitantes, já que diminuir o tempo de missão



provoca o espaçamento dos pontos, foi necessário identificar qual objetivo é mais vital para a segurança e qualidade. Dessa forma, o *fitness* de tempo tem como máximo igual a 10 para a duração da missão para que seja considerado principal, e assim, evitar acidentes de queda do VANT. Os valores de cobertura horizontal e vertical foram escalados para 5 para que os seus efeitos somados pudessem ter o mesmo impacto do tempo. Porém, valores de tempo maiores que o valor máximo são eliminados do processo de evolução.

Outro problema é que ambos os objetivos tendem a afastar o VANT da superfície, gerando missões mais curtas e com maior interseção. Porém, esse afastamento afeta a acurácia da reconstrução 3D, onde 40 metros corresponde a acurácia de 6 a 10 cm [Dai e Feng 2014]. Dessa forma, é necessária a adição de um ganho no *fitness* para a correção desse parâmetro como visto nas Equações 3.20 e 3.21. Os valores  $D_{\max} - D_{\min}$  são as distâncias máximas e mínimas, respectivamente, em relação à superfície. Note que pela Equação 3.20 o ganho será unitário quando  $Dist_H = D_{\min}$  e zero quando  $Dist_H = D_{\max}$ . Para os resultados têm  $D_{\min} = 1$  m para evitar batidas com a superfície e  $D_{\max} = 40$  m para evitar perda da qualidade da reconstrução 3D.

$$G_{\text{dist}} = \frac{-Dist_H}{D_{\max} - D_{\min}} + \frac{D_{\min}}{D_{\max} - D_{\min}} + 1 \quad (3.20)$$

$$Fitness_{\text{final}} = G_{\text{dist}} * Fitness_{\text{total}} \quad (3.21)$$

Outra adição à missão é o ângulo em relação à superfície. Para aumentar a qualidade da reconstrução 3D, o ângulo entre a normal da superfície e a câmera deve ser mais próximo de  $0^\circ$ , chamado de ângulo de incidência [Dai e Feng 2014]. Para o cálculo desse ângulo utiliza-se o ângulo perpendicular à reta formada pelos pontos paralelos da Figura 12. Outras características que influenciam na qualidade da reconstrução, tais como distância focal e resolução da câmera são inerentes ao equipamento e não à missão, não sendo consideradas para a otimização.

Para realizar a otimização meta-heurística foi utilizado o pacote MEALPY [Thieu 2020] com as configurações padrões descritas pelo pacote. Algumas modificações foram necessárias para a utilização, já que o mesmo utiliza minimização

da função de *fitness*, não restringindo a valores positivos. No caso foi necessário modificar a função de *fitness* para a Equação 3.22. Para todas as meta-heurísticas testadas, foi utilizada a mesma função objetivo.

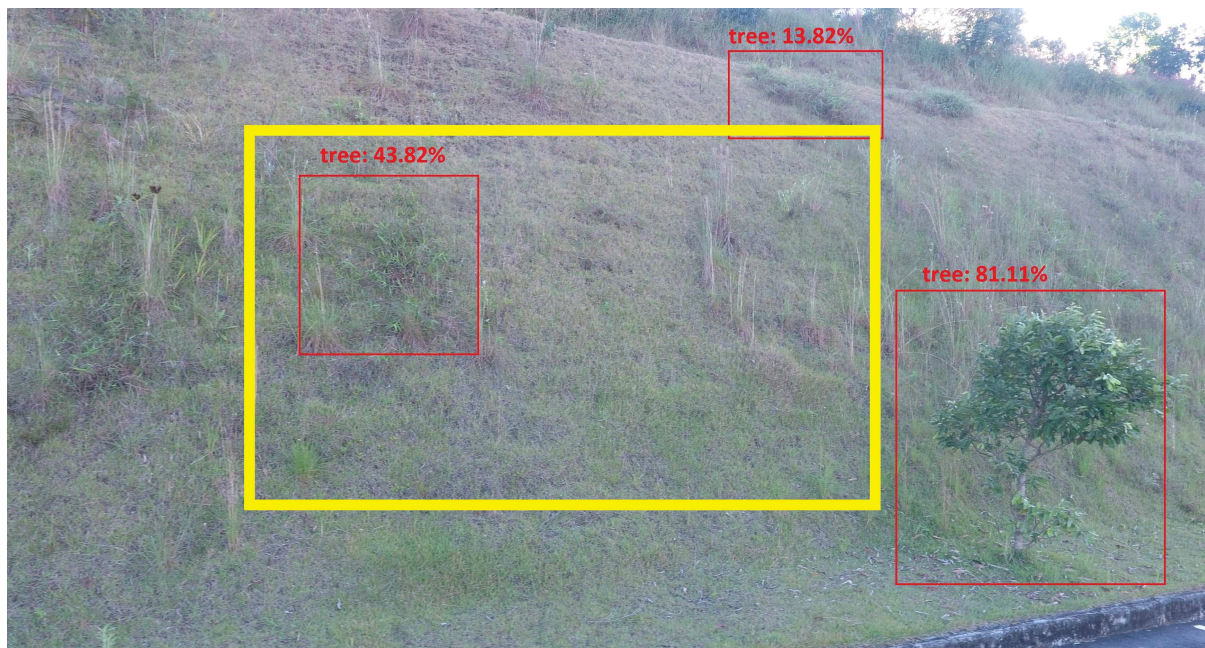
$$Fitness_{Mealpy} = \frac{1}{Fitness_{Final}} \quad (3.22)$$

### 3.3 Identificação de Objetivos Dinâmicos

Após os *waypoints* serem entregues pela etapa de planejamento do caminho o VANT, inicializa-se então a missão. Ao chegar em cada ponto, o VANT captura uma foto da câmera frontal para a reconstrução 3D. Realiza-se a análise dos pontos de interesse nesta imagem.

Um mundo virtual com a presença de pontos de cores distintos do fundo foi utilizado um filtro de cor do OpenCV [Zhang et al. 2020]. Em ambientes reais, considera-se um detector de objetos como o ImageAI [Moses e Olafenwa 2018]. A Figura 15 mostra a identificação usando o algoritmo ImageAI [Moses e Olafenwa 2018], treinado com um conjunto de imagens para identificar vegetação. No caso, somente os pontos identificados dentro do quadrado amarelo são referencias para as missões de aproximação. Essa escolha busca evitar que pontos na borda gerem missões que não acompanharão esse objeto.

Figura 15 – Identificação de Pontos de Interesse em mundo real.

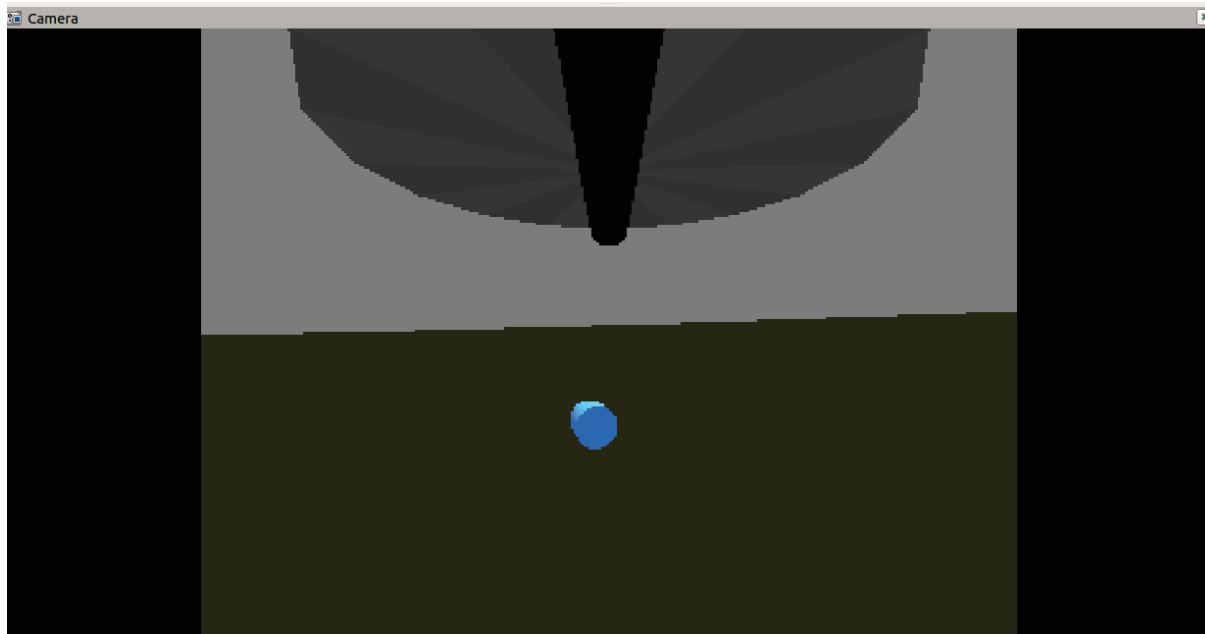


Fonte: Elaborado pelo autor.

O algoritmo realiza a identificação para averiguar se naquela imagem existe um ponto de interesse, como mostrado na Figura 16. Caso tenha, o VANT executa uma pequena submissão de aproximação no local. Caso não, tem-se a continuação da missão para o próximo *waypoint*. Na reconstrução 3D, vários desafios surgem pela análise da deformação baseada em nuvem de pontos. Apesar do crescente número de métodos para detectar e quantificar deslocamento e deformação utilizando nuvens de pontos, muitos desafios não foram resolvidos [Holst e Kuhlmann 2016]. Além disso, muitas aplicações se beneficiam dos dados morfológicos da superfície. Por exemplo, erosão, sedimentação e outros fatores ambientais podem remodelar estruturas e criar riscos para as pessoas e para o meio ambiente [Zhou e Gong 2015]. O monitoramento da alteração estrutural é uma das principais preocupações em relação a estruturas. Dessa forma, a boa identificação dos locais de interesse podem servir como referências para o casamento de reconstruções e o aumento da qualidade das mesmas. Caso seja realizada a aproximação em toda a superfície, a missão não teria tempo de execução muito maior que o tempo máximo disponível,

impossibilitando a cobertura de toda região.

Figura 16 – Identificação de Pontos de Interesse em Simulador.

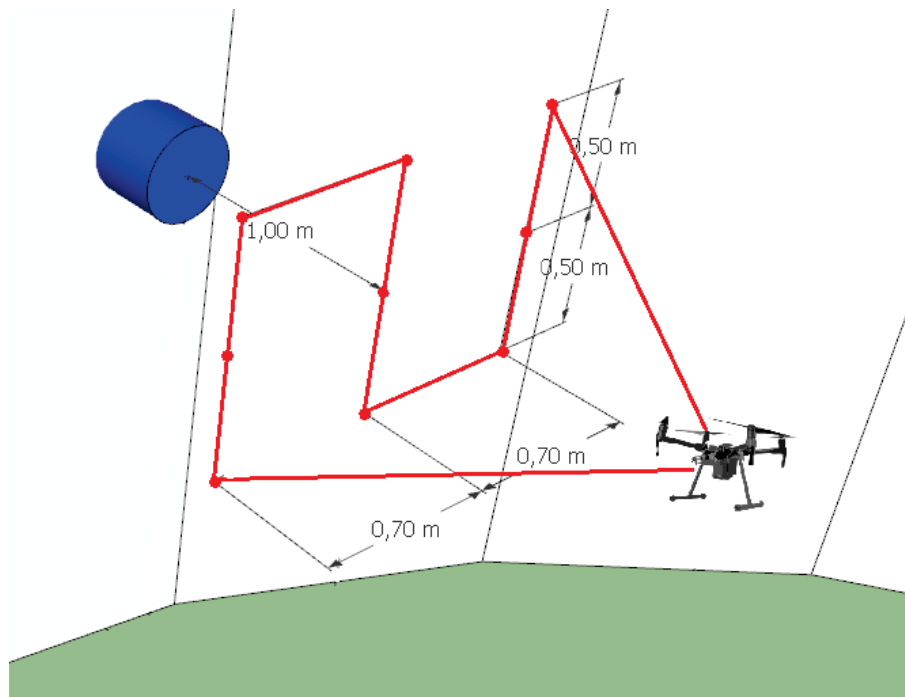


Fonte: Elaborado pelo autor.

### 3.4 Missões de Aproximação e Readequação da Missão Geral

Após a identificação do ponto de interesse, existe uma mudança em relação ao movimento do VANT. A missão original é pausada para a realização de uma submissão vertical local, mudando do horizontal original. A partir da identificação é feita uma aproximação, como visto na Figura 17, onde são criados 9 *waypoints* com distância de 1 metro da superfície. Os pontos centrais têm a mesma altura do VANT, enquanto os outros 2 conjuntos estão em altitude superior e inferior ao VANT de 0.5 m. No caso de regiões com inclinação vertical, a missão seguirá o comportamento da superfície, mantendo todos os pontos a uma distância de 1 m.

Figura 17 – Criação de Pontos com Aproximação de 1m.



Fonte: Elaborado pelo autor.

Esta pequena missão local aumenta o registro de pontos para a reconstrução 3D, aumentando a qualidade em torno do interesse. Porém, a missão original não considera essa mudança, alterando o tempo disponível para a realização da mesma. Desta forma, é necessário realizar uma nova otimização dos pontos, agora com o novo tempo disponível e com a área ainda não fotografada.

A cada fim de translado horizontal, é realizada uma nova iteração do CPP para a aquisição de *waypoints* com as novas entradas de tempo e área. Logo, além das missões de aproximação, essa otimização também considera outras perdas de performance durante a missão, pois trabalha com o tempo disponível para o VANT.

### 3.5 Conclusões Parciais

O método proposto tem como objetivo criar um CPP visando critérios fotométricos e de segurança para missões de vistorias em superfícies. Vale destacar que o processo é otimizado por meta-heurísticas, levando em consideração os dois

objetivos. A metodologia controla o VANT desde sua decolagem, passando pela captura de informações do ambiente e da própria missão, até o retorno ao ponto de partida. São capturadas imagens para que sejam transformadas em reconstruções 3D. No próximo capítulo, temos os resultados da implementação desse algoritmo em dois ambientes, além do funcionamento do mesmo quando as informações da superfície são entregues para a nuvem de pontos.

As meta-heurísticas otimizam os três parâmetros que definem a missão proposta, criando missões adequadas para a realização do voo de cobertura. Apesar de ser um processo que tem um custo computacional proveniente da própria meta-heurística, o processo não é realizado em todo o voo, e sim, no início e em cada final de translado, possibilitando o encaixe dos algoritmos à proposta de um CPP *online*. Além disso, a execução da otimização no final de cada parte da missão permite que o algoritmo se adapte aos possíveis atrasos gerados tanto por questões não esperadas quanto pelas missões locais de aproximação, ajustando o tempo restante à região ainda não vistoriada.

## 4 Resultados e Discussões

O Capítulo apresenta os resultados de simulação e em ambiente real para a metodologia proposta. A Seção 4.1 apresenta os resultados realizados no simulador Gazebo-ROS explicados na Seção 4.1.1, além dos métodos de comparação. A Seção 4.1.2 apresenta a comparação entre as meta-heurísticas da Seção 2.2 e sua aplicação no problema proposto com o objetivo de identificar o tamanho da população ideal e qual método será escolhido para os próximos estudos. A Seção 4.1.3 mostra a aplicação das missões locais e seu impacto no planejamento global, além do replanejamento. A Seção 4.1.4 compara as mudanças permitidas na função de aptidão: mudança do tempo máximo da missão e da relação entre os objetivos de tempo e fotometria.

A Seção 4.2 apresenta os resultados em ambiente real para o algoritmo, destacando-se os impactos das variáveis de decisão em reconstruções tridimensionais. Já a Seção 4.3 mostra a comparação da metodologia proposta com outros algoritmos da literatura em ambiente real.

### 4.1 Testes em simulação

#### 4.1.1 Informações sobre simuladores, VANT virtual e metodologia de comparação

Para a realização dos testes da metodologia, implementou-se um talude fictício para testes no simulador Gazebo [Koenig e Hsu 2013] com um VANT virtual. A simulação em trabalhos na área de robótica é uma ferramenta essencial para criar um ambiente seguro onde os comportamentos podem ser estudados sem que tenham riscos de acidentes e danos em equipamentos. Um simulador bem projetado torna possível testar rapidamente algoritmos, projetar robôs, executar testes de regressão, e treinar o sistema de inteligência artificial usando cenários realistas. O Gazebo oferece a capacidade de simular com precisão e eficiência populações de robôs em ambientes internos e externos complexos, com funcionalidades de cinemática e dinâmica robustos, gráficos de alta qualidade e interfaces gráficas convenientes.

O modelo de VANT escolhido foi o “Hector\_Quadrotor” [Meyer et al. 2012]. A Figura 18 apresenta o modelo no Gazebo do Quadrotor. O VANT utiliza modelos

Figura 18 – Modelo utilizado no Gazebo.



Fonte: Retirado de [Meyer et al. 2012].

dinâmicos e cinemáticos para simular forças que atuam sobre os equipamentos, tais como torque de cada um dos seus quatro motores, centro de massa, entre outros. Esses modelos permitem estimar com certa precisão o comportamento do VANT como atitude, posição e velocidade. O modelo feito por [Meyer et al. 2012] apresenta sensores implementados como *plugins* do Gazebo independentes e podem ser conectados ao VANT, simulando os sensores disponíveis no mercado. Todos os sensores compartilham um modelo de erro comum de Gauss Markov de primeira ordem [Meyer et al. 2012]. O Hector\_Quadrotor possui os seguintes sensores:

- *Inertial Measurement Unit* (IMU);
- Sensor Barométrico;
- Sensor Ultrassônico de Altura para Decolagem e Pouso;
- Sensor de campo magnético;
- *Global Positioning System* (GPS);
- A Hokuyo UTM-30LX laser scanner.



Além dos sensores, o Hector\_Quadrotor possui total compatibilidade com as funções em ROS. O *Robot Operating System* [Quigley et al.] (ROS ou sistema operacional de robôs) é uma coleção de *frameworks* de *softwares* para desenvolvimento de robôs. O ROS fornece serviços padrões de sistema operacional, tais como abstração de *hardware*, controle de dispositivos de baixo nível, a implementação de funcionalidades comuns, passagem de mensagens entre processos e gerenciamento de pacotes. É importante ressaltar que o Hector\_Quadrotor já vem implementado com funcionalidades, que, junto aos modelos dinâmicos do sistema, aproximam o simulador a VANT's reais, sendo o motivo de escolha do mesmo.

- Algoritmos de Decolagem;
- Algoritmos de Pouso;
- Algoritmos de ida ao Ponto  $[x, y, z]$  com Orientação [roll, pitch, yaw];
- Leitura de *LaserScan*;
- Entre outros;

A comparação dos resultados utilizou o método de análise da variância (*Analysis of variance - ANOVA*) [Kim 2017]. ANOVA é uma técnica estatística para comparações entre médias de populações, permitindo identificar se existe uma diferença significativa entre as médias usando o teste  $F$  para verificar a igualdade da média dos grupos. O teste ANOVA  $F$  também é chamado de teste abrangente, pois testa a hipótese nula não específica, ou seja, se todas as médias do grupo são iguais. A saída é uma tabela ANOVA como exemplificado na Tabela 3.

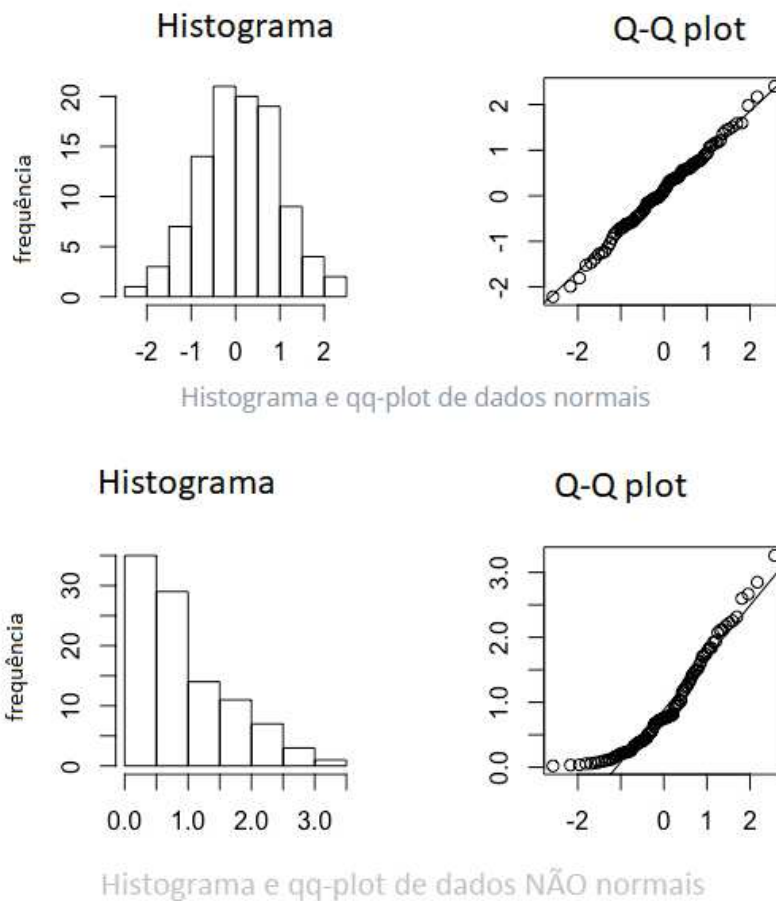
Assim como o teste t, a ANOVA possui algumas restrições para ser usada [Franco 2016]:

- Os dados devem ser normais ou próximos da normalidade.
- As variâncias de cada amostra devem ser parecidas. Caso não seja, é possível padronizar dividindo pela variância amostral de forma a obter variância igual a um em todos os grupos.

- As amostras devem ser independentes.

A ferramenta mais simples e a mais recomendada para verificar a distribuição dos seus dados com o histograma ou o *quantile plot* (qq-plot) que irá dispor em um gráfico uma comparação dois a dois dos quantis teóricos de uma Normal e os quantis de seus dados. Se os pontos se concentrarem em torno de uma reta, então temos indícios de que a distribuição é Normal. A Figura 19 representa um histograma e um qq-plot para uma distribuição normal (superior) e uma distribuição exponencial (inferior).

Figura 19 – Exemplo de distribuições normal e não normal.



Fonte: Retirado de [Franco 2016]

Tabela 3 – Exemplo de tabela ANOVA.

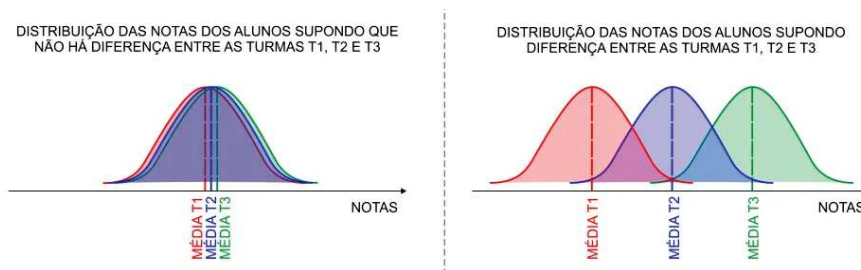
Tabela ANOVA					
	Df	Sum Square SS	Mean Square MS	Valor $F$	Pr(>F)
Grupo	2	56,806	28,403	5,25	0,010
Resíduos	33	178,783	5,418		

Para entender a tabela ANOVA considere o seguinte exemplo: suponha que um curso preparatório possui um corpo docente com três professores para a mesma disciplina, cada um responsável para uma turma. Para verificar se o desempenho dos alunos é relacionada ao trabalho desenvolvido por seus professores calcularam as médias das notas de cada turma, sendo essas: 760; 810 e 800 para as turmas A, B e C, respectivamente. Olhando somente para essa informação, o professor que teve melhor desempenho foi o da turma B, mas essa informação é suficiente para afirmar que o desempenho dos alunos de cada turma é diferente? E se um dos professores tiver em sua turma um aluno que não se preparou e errou quase todas as questões, afetando a média do grupo? Para verificar se o desempenho variou com o professor é necessário utilizar um teste estatístico além das médias [Oliveira 2021].

Um dos objetivos da aplicação da ANOVA é verificar se há diferença entre distribuição de uma medida entre grupos. Em nosso exemplo, podemos definir as hipóteses do teste como: Não existe diferença entre o desempenho das notas dos alunos de cada professor e há pelo menos um professor com alunos com desempenho diferente, sendo essas as hipóteses  $H_0$  e  $H_1$ . Caso os alunos apresentem a mesma variabilidade e a mesma média, suas distribuições tendem a se sobrepor, enquanto se as médias são diferentes as distribuições se diferenciam, como mostrado na Figura 20, onde a distribuição da esquerda estão sobrepostas, enquanto a da direita se diferenciam. As informações geradas na análise de variância estão resumidas na Tabela 3.

A primeira linha da tabela corresponde à análise da variância dos grupos, enquanto que a segunda linha representa a análise de variância dos resíduos. Grupos é o número de professores, ou seja, os agrupamentos de dados. Já os resíduos são relacionados aos alunos dentro de cada grupo. A variância é uma medida de

Figura 20 – Exemplo de distribuições entre as turmas.

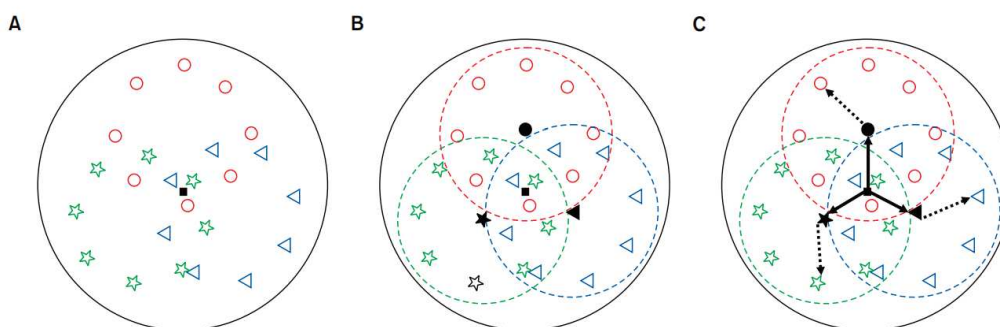


Fonte: Retirado de [Oliveira 2021]

dispersão que mostra o quão distante cada valor desse conjunto está do valor central. Se menor é a variância, mais próximos os valores estão da média; mas quanto maior ela é, mais os valores estão distantes. Na primeira coluna temos o grau de liberdade (DF). DF é a relacionado com base no tamanho dos grupos a serem analisados e no número total de amostras, sendo caracterizado pela dimensão da amostra menos o número de parâmetros estatísticos a serem avaliados na população. É um estimador do número de categorias independentes num teste estatístico. A próxima coluna é a soma dos quadrados. A soma de quadrados dos grupos mede a variação entre os professores de cada turma e a soma dos quadrados dos resíduos mede a variação dentro de cada turma. Na próxima coluna tem-se o quadrado médio ajustado (*Mean Sq*). É a soma dos desvios quadrados divididos pelo DF. Quanto mais distante a média do grupo está, maior esse número se torna. Em seguida, temos o valor da estatística F utilizada para realizar o teste de hipótese (Fator F) e por último o valor-p ( $\Pr(>F)$ ) [Oliveira 2021].

O Fator F, definido pela Equação 4.1 e representado na Figura 21, mostra a relação entre as médias de cada grupo e a média geral. Por exemplo considera a Figura 21 (A) que mostra um conjunto de dados. Nesse conjunto de dados estão presente três grupos: círculos vermelhos, estrelas verdes e triângulos azuis. Os dados possuem três diferentes médias de grupos - Figura 21 (B), destacadas pelas figuras pretas preenchidas que são distintas. Para avaliar a eficiência da divisão do conjunto de dados em três grupos é comparada duas métricas: as distâncias das médias do grupo à média geral e as distâncias das médias do grupo a cada um dos dados. A distância entre as médias dos grupos e a média geral (setas sólidas)

Figura 21 – Exemplo de variância intergrupo e intragrupo.



Fonte: Retirado de [Kim 2017].

representa a variância intergrupo e a distância entre as médias dos grupos e os dados de cada grupo (setas pontilhadas) representam as variâncias intragrupo - Figura 21 (C) [Kim 2017].

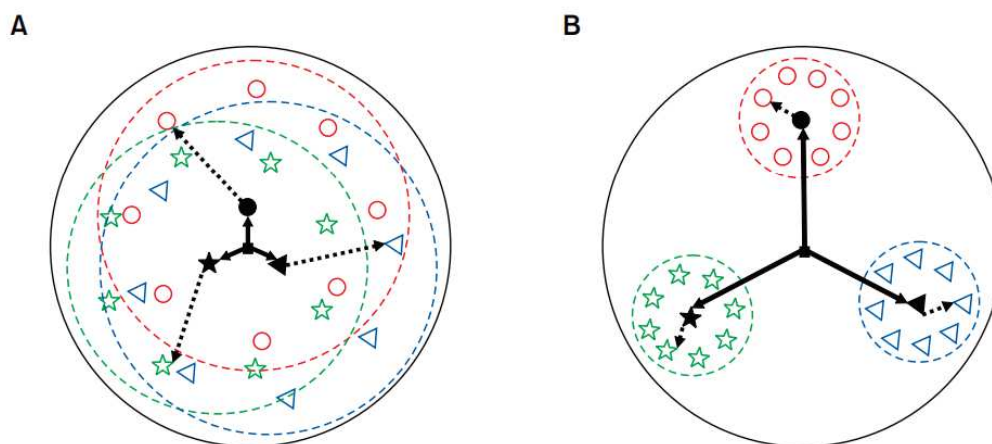
$$F = \frac{\text{Variância Intergrupo}}{\text{Variância Intragrupo}} \quad (4.1)$$

A Figura 22 mostra dois exemplos para fator  $F$  pequeno (A) e grande (B). Quanto maior é o Valor  $F$ , mais distintos são os grupos. Percebe-se que no exemplo A a divisão dos dados em três grupos distintos não parece válida, mesmo que as médias sejam distintas. Já no exemplo B têm-se as divisões em grupos mais destacadas.

O valor- $p$  indica a probabilidade de se observar uma diferença tão grande ou maior do que a que foi observada sob a hipótese. Um nível de significância de 0,05 no valor  $p$ , ou seja, se o valor de  $p$  for menor que 0,05, então existe pelo menos uma diferença significativa entre dois grupos. Um valor- $p$  baixo não quer dizer que a média é distinta em todos os grupos, mas sim que existe pelo menos uma média diferente das demais. Tomando como base a Tabela 3, pode-se concluir que existe pelo menos um professor com alunos com desempenho significativamente diferente ao avaliar o valor- $p = 0,010$  (menor que o nível de significância estabelecido de 0,05). O valor de significância de 5% foi baseado em trabalhos utilizando ANOVA na área de robótica, tais como: [Chien e Wang 2010] e [Mohsin et al. 2020].

Por isso a importância de se observar os gráficos para verificar qual é

Figura 22 – Exemplo de grupos iguais e grupos distintos.



Fonte: Retirado de [Kim 2017].

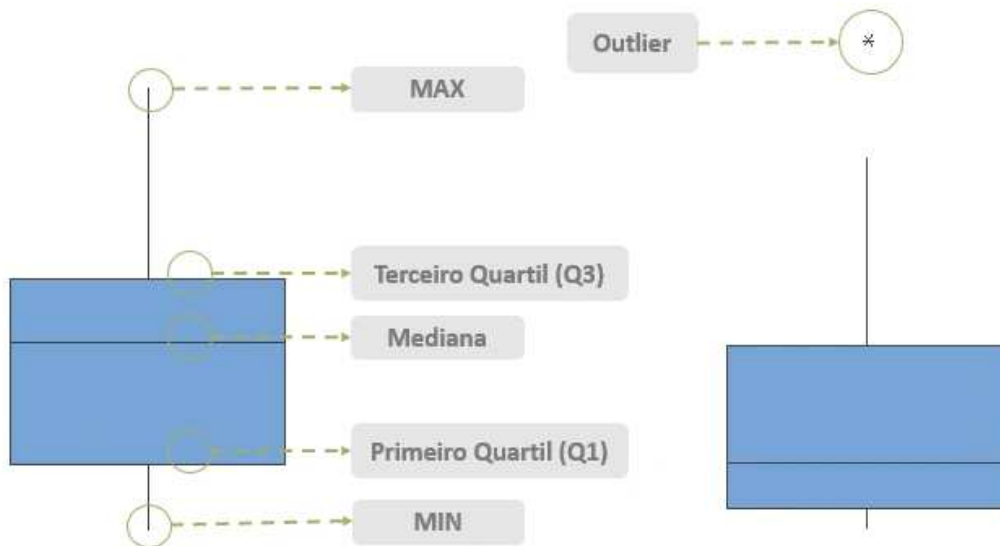
o conjunto diferente. O gráfico escolhido para mostrar os dados foi o boxplot [Benjamini 1988]. Dessa forma para o critério de identificar nos resultados se eles são distintos e podem ser analisados de forma separada utiliza-se a significância do valor-p, onde se o Valor  $F$  for menor que 0,05 têm-se conjuntos distintos que podem ser comparados. Se o Valor for maior que 0,05 então os conjuntos não podem ser divididos e serão analisados como um conjunto só.

No boxplot a parte central do gráfico contém os valores que estão entre o primeiro quartil (Q1) e o terceiro quartil (Q3). As hastes inferiores e superiores se estendem, respectivamente, do primeiro quartil até o menor valor (MIN), limite inferior, e do terceiro quartil até o maior valor (MAX). Do valor mínimo até o início da caixa, estão representados 25% dos dados, já dentro da caixa estão representados mais 50% dos dados, e por fim, a haste superior representa os outros 25% restantes. Um exemplo de boxplot pode ser visto na Figura 23.

Quando um boxplot é mais achatado, indica uma baixa variabilidade e desvio padrão. Já em casos onde a diferença entre Q1 e Q3 é maior, a variabilidade é muito grande, trazendo certa imprevisibilidade, pois ora se tem valores baixos, ora se tem valores maiores. Dessa forma, além da análise do valor médio, demonstrado pela mediana, é necessário verificar a diferença entre os quartis para um sistema mais previsível. No boxplot, quando se tem um *outlier* facilmente conseguimos visualizá-

los através do asterisco apresentado no boxplot. Esses dados são observações fora do comum, anomalias ou valores atípicos.

Figura 23 – Exemplo de Boxplot.



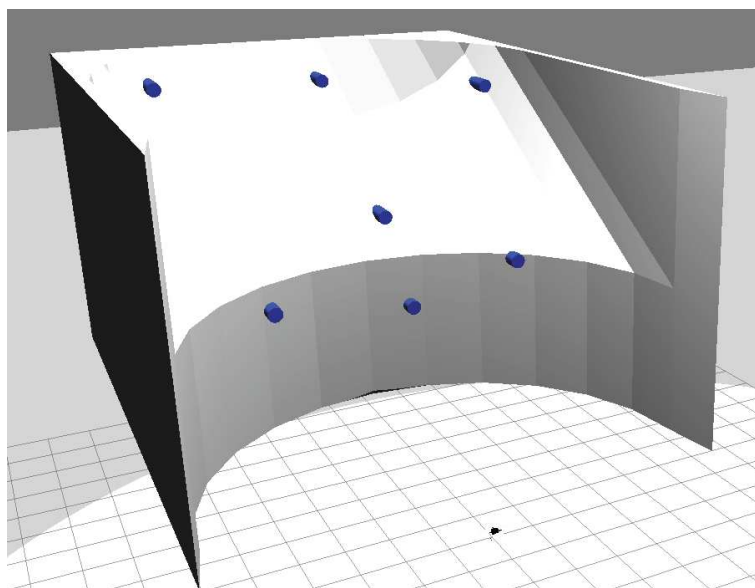
Fonte: Retirado de [Rodrigues 2021].

#### 4.1.2 Comparação entre meta-heurísticas

A Seção 2.2 mostra o conceito das meta-heurísticas escolhidas. Nesta Seção teremos a aplicação de cada meta-heurística ao problema com o objetivo de escolher o melhor método, além da melhor quantidade de indivíduos. O mundo virtual para comparação, visualizado na Figura 24, é um talude convexo com inclinação na metade superior. Este mundo representa uma aproximação de um talude do vertedouro da Usina Hidrelétrica Barra Grande - BAESA em Pinhal da Serra, Rio Grande do Sul, Brasil. A estrutura possui em sua base um talude sem inclinação de 4 metros de altura que tem formato de um semicírculo de 6 metros de raio. Entre a altura de 4 e 9 metros tem-se uma variação na inclinação do Talude, passando de 6 metros de profundidade até 14 metros. A Figura 25 apresenta uma foto do local inspirador para este mundo. Além disso, tem-se a aplicação de canos de 50 cm de diâmetro na cor azul como pontos de interesse. As informações do ambiente são

entregues a meta-heurística através de sensores *Laser* no VANT após o filtro de pontos próximos, como mostrado na Figura 26. Para este talude o tempo máximo de execução foi de 10 minutos. Não foi executada a identificação dinâmica de pontos para esse resultado.

Figura 24 – Talude inclinado para teste de meta-heurística.



Fonte: Elaborado pelo Autor.

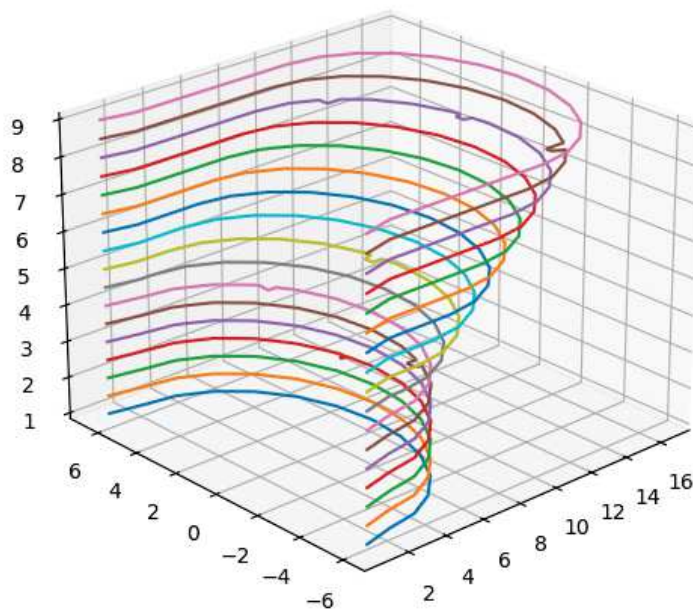


Figura 25 – Foto retirada de um talude do vertedouro da Usina Hidrelétrica Barra Grande em Pinhal da Serra, Rio Grande do Sul, Brasil .



Fonte: Elaborado pelo Autor.

Figura 26 – Leituras após filtro do talude inclinado para teste de meta-heurística.



Fonte: Elaborado pelo Autor.

As metodologias escolhidas para comparação são: Algoritmo genético (GA), *Flower Pollination Algorithm* (FPA), *Bat Algorithm* (BA), *Particle swarm optimization* (PSO), *Ant Colony Optimization* (ACO), *Bird Swarm Algorithm* (BSA), *Simulated Annealing* (SA) e *Culture Algorithm* (CA) por apresentar resposta com tempo de execução menor que o tempo máximo da simulação. A Tabela 4 mostra os hiper-parâmetros utilizados nas meta-heurísticas. As outras metodologias citadas no documento e na biblioteca MEALPY foram testadas, porém, seus resultados tiveram tempo de aquisição maior que os 10 minutos disponíveis para a missão ou valores que geraram interseções abaixo de 20% tanto no eixo vertical quanto horizontal de cobertura.

Para cada tamanho de população foram criadas 50 populações iniciais e entregues igualmente para cada algoritmo. Para a criação dessas populações foram usados valores aleatórios nos seguintes intervalos:  $D_{\min} = [1; 10] \in \mathbb{R}$ ,  $Dist_H = [1; 100] \in \mathbb{R}$  e  $Num_{\text{way}} = [1; 20] \in \mathbb{N}$ .

Como tratamento de soluções ineficazes, como valores negativos para as distâncias e para o número de traslados, foi considerado um fitness igual a uma constante grande, já que o algoritmo utilizado trabalha com minimização.  $Dist_H < 1$  também teve o mesmo tratamento por questões de segurança, já que o VANT ficaria muito próximo da superfície. Missões com tempo maiores que o tempo máximo permitido tiveram seus fitness multiplicados por 10000, não sendo diretamente retirados do processo de otimização, mas dando uma penalidade grande no processo de busca. Porém, na execução essas missões não são realizadas, fazendo o VANT retornar uma mensagem de erro ao operador e retornar ao ponto de partida para evitar danos ao equipamento.

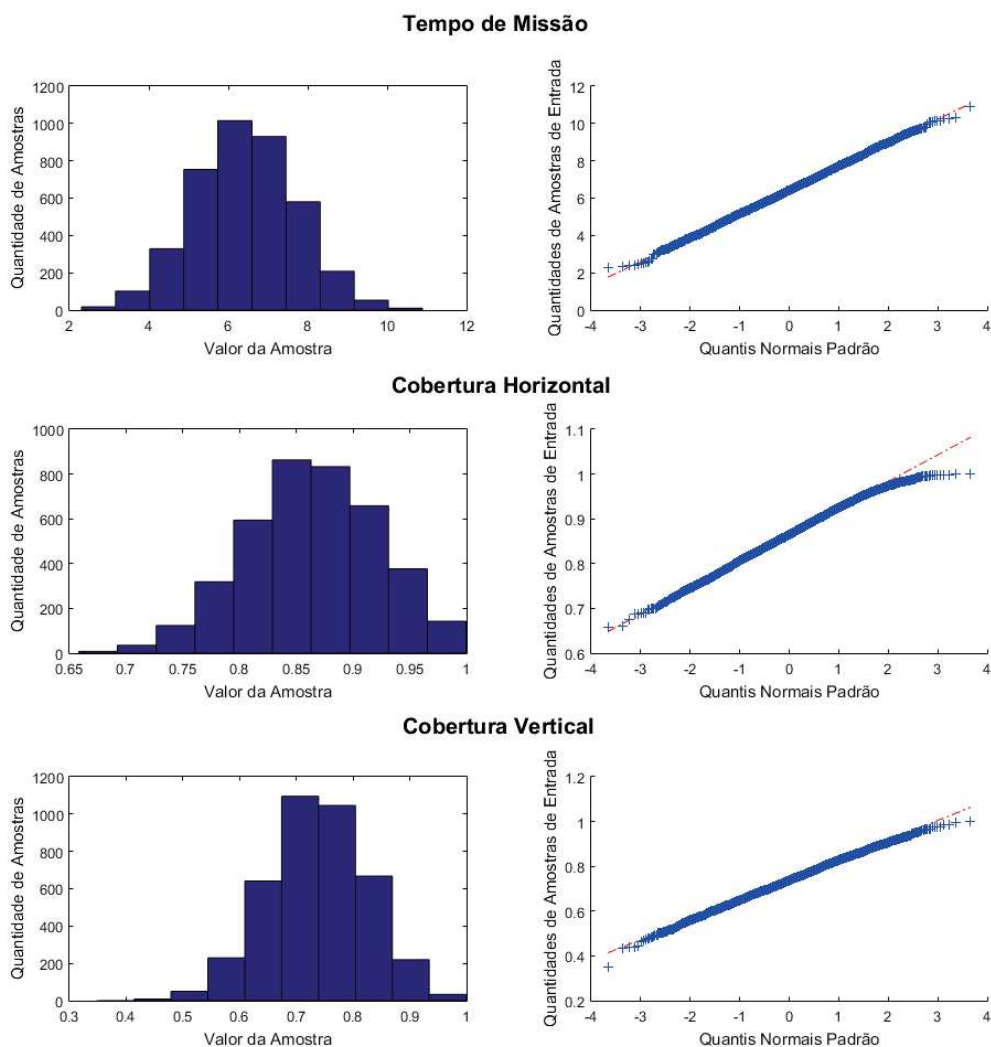
Para a aplicação da Tabela ANOVA foi realizado o teste de normalidade das amostras, como mostrado na Figura 27. Percebe-se pelos histogramas e pelo qq-plot que os dados têm uma distribuição próxima da normal, sendo que os dados de Cobertura horizontal tem-se uma maior variação a reta perto das amostras de entrada próximos a 1, que é o limite superior da interseção entre as imagens (100%). No processo de otimização, quando o deslocamento horizontal é maior que a FoV horizontal, esse valor é tratado manualmente, gerando missões próximas a 95% de cobertura. Os valores de médias para o tempo de missão, cobertura horizontal e

Tabela 4 – Hiper-parâmetros das metaheurísticas utilizadas.

<b>Algoritmo Genético (GA)</b>				
Probab. de Crossover	Probab. de Mutação	Método de Seleção	Porcentagem da População para cada torneio	Tipo de Crossover
0,95	0,025	Torneio	0,2	Uniforme
<b><i>Flower Pollination Algorithm (FPA)</i></b>				
Probabilidade de Troca				
0,8				
<b><i>Bat-inspired Algorithm (BA)</i></b>				
Amplitude (Amax;Amin)			Taxa de Pulso (rmin;rmax)	
(1,0;2,0)			(0,15; 0,85)	
<b><i>Particle Swarm Optimization (PSO)</i></b>				
Coef. Local	Coef. Global	Peso Mínimo das Aves	Peso Máximo das Aves	
2,05	2,05	0,4	0,9	
<b><i>Ant Colony Optimization (ACO)</i></b>				
Número de amostras recém-geradas		Fator de seleção		Relação Desvio-Distância
50		0,5		1
<b><i>Bird Swarm Algorithm (BSA)</i></b>				
Frequência de Voo	Probabilidade de procurar alimentos	Coeficiente cognitivo e Coeficiente social (c1;c2)	Efeito indireto e direto nos comportamento de vigilância das aves (a1;a2)	O coeficiente para seguir a liderança
10	0,8	(2;2)	(1,5;1,5)	0,5
<b><i>Simulated Annealing (SA)</i></b>				
Número Máximo de Subiterações	Temp. inicial	Temp. Final	Taxa de Mutação	Tamanho do Passo de Mutação
5	1000	1	0,1	0,1
<b><i>Culture Algorithm (CA)</i></b>				
Probabilidade de Taxa de Aceitação				
0,15				

vertical são, respectivamente, de 6,41; 0,86 e 0,73, enquanto o desvio padrão é dado por: 1,24; 0,06 e 0,09. Sobre a variância, como são distintas para cada população, todas foram normalizadas pela variância amostral de forma a obter variância igual a um em todos os grupos [Franco 2016].

Figura 27 – Histograma e qq-plot dos dados de Tempo de Missão, Cobertura Horizontal e Cobertura Vertical.



Fonte: Elaborado pelo Autor.

A primeira etapa de avaliação tem como objetivo a escolha da população

de indivíduos que se adéquem aos objetivos propostos e ao tempo de aquisição de resultados. Essa população tem que criar missões com coberturas e tempo de missão adequadas, porém o processamento não pode superar o período do traslado horizontal da missão, já que processo de otimização ocorre durante essa etapa. Para isso as populações variam de 100 até 6000 indivíduos. Valores superiores a 6000 foram descartados, pois, o tempo de processamento era maior que os 10 minutos disponíveis para a missão sem adição de qualidade nos resultados.

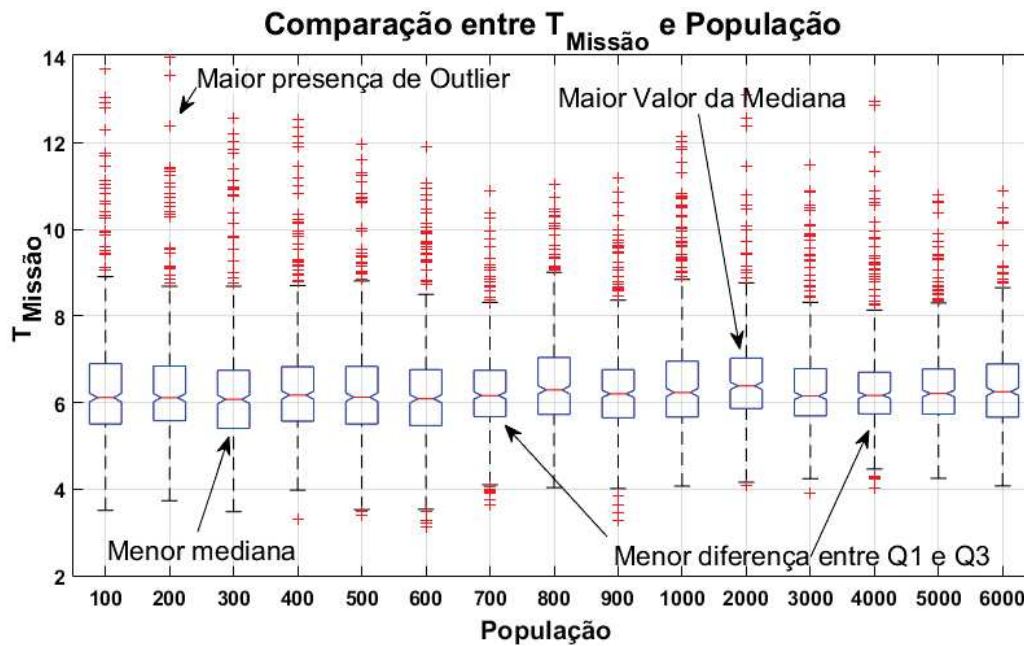
A primeira análise foi em relação aos objetivos da meta-heurística: tempo da missão, cobertura horizontal e vertical. Os grupos foram tamanhos de população variando entre 100 até 6000 indivíduos com o objetivo de analisar se os dados são distintos e dessa forma podem ser comparados.

Os resultados em relação ao tempo de missão são mostrados na Tabela ANOVA 5 e no boxplot da Figura 28. O fator-p foi inferior aos 5%, indicando que existe pelo menos uma diferença significativa entre os grupos. Como os dados são distintos, as próximas análises foram através do boxplot da Figura 28. As populações que tem a maior previsibilidade, ou seja, a menor diferença entre Q1 e Q3 são os grupos de 700 e 4000 indivíduos, sendo candidatos a escolha junto com população de 300 que tem a menor mediana, já que o objetivo é de minimização. Os grupos de 200 e 2000 tiveram o pior resultado, com a maior presença de *outliers* e o maior valor da mediana, respectivamente. Apesar dessas constatações, percebe-se que em grande maioria todos os resultados ficaram entre 5 e 6 minutos e mesmo que os conjuntos sejam distintos, não estão com resultados muito distantes.

Tabela 5 – Tabela ANOVA de  $T_{Missão}$  com a variação da população

Tabela ANOVA					
	Df	Sum Square SS	Mean Square MS	Valor $F$	Pr(>F)
População	14	44,2239	3,1588	1,9773	0,015825
Resíduos	6735	10759,2781	1,5975		

Figura 28 – Boxplot entre tempo de missão e a população.



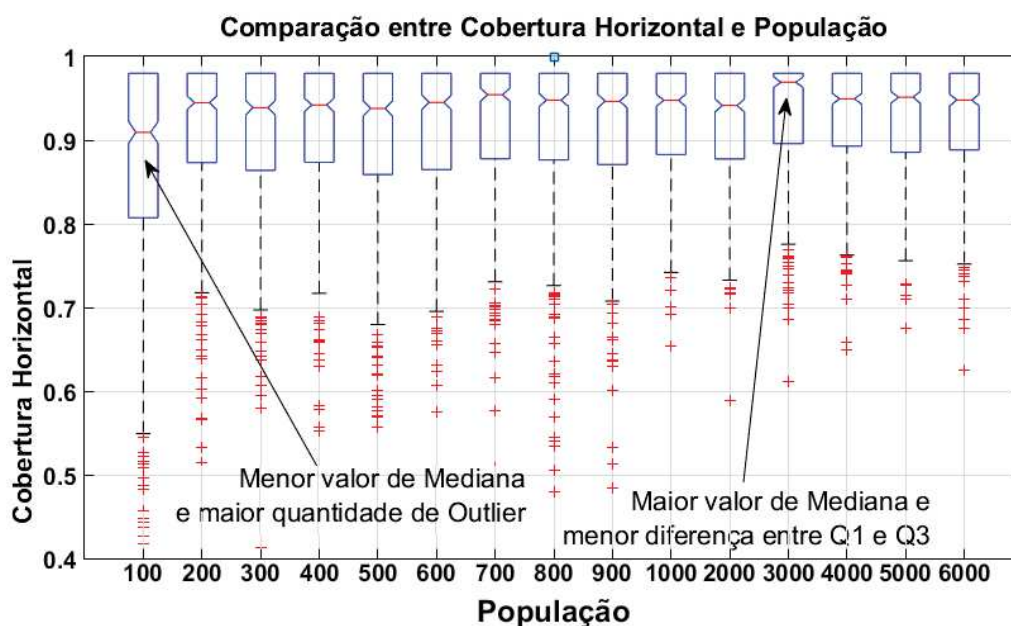
Fonte: Elaborado pelo Autor.

O segundo objetivo a ser avaliado é a cobertura horizontal mostrado na Tabela ANOVA 6 e na Figura 29. Novamente tem-se o valor do fator-p muito abaixo dos 5%, podendo ser considerado zero pela sua grandeza. Como o objetivo é maximizar a cobertura temos o pior candidato o grupo de 100 indivíduos, já que tem a menor mediana e a maior quantidade de *outliers*. Considerando a previsibilidade e o valor da mediana, o melhor resultado foi o de 3000 indivíduos, já que possui menor diferença entre os quartis e maior valor da mediana. Porém, se removermos a população de 100 indivíduos, percebe-se que todas as populações tiveram seus quartis acima de 95% de cobertura, o que indica alta cobertura horizontal.

Tabela 6 – Tabela ANOVA de  $Cobertura_{\text{Horizontal}}$  com a variação da população

Tabela ANOVA					
	Df	Sum Square SS	Mean Square MS	Valor $F$	Pr(>F)
População	14	2,168	0,15486	19,0794	8,62e-48
Resíduos	6735	54,664	0,0081164		

Figura 29 – Boxplot entre cobertura horizontal e a população.



Fonte: Elaborado pelo Autor.

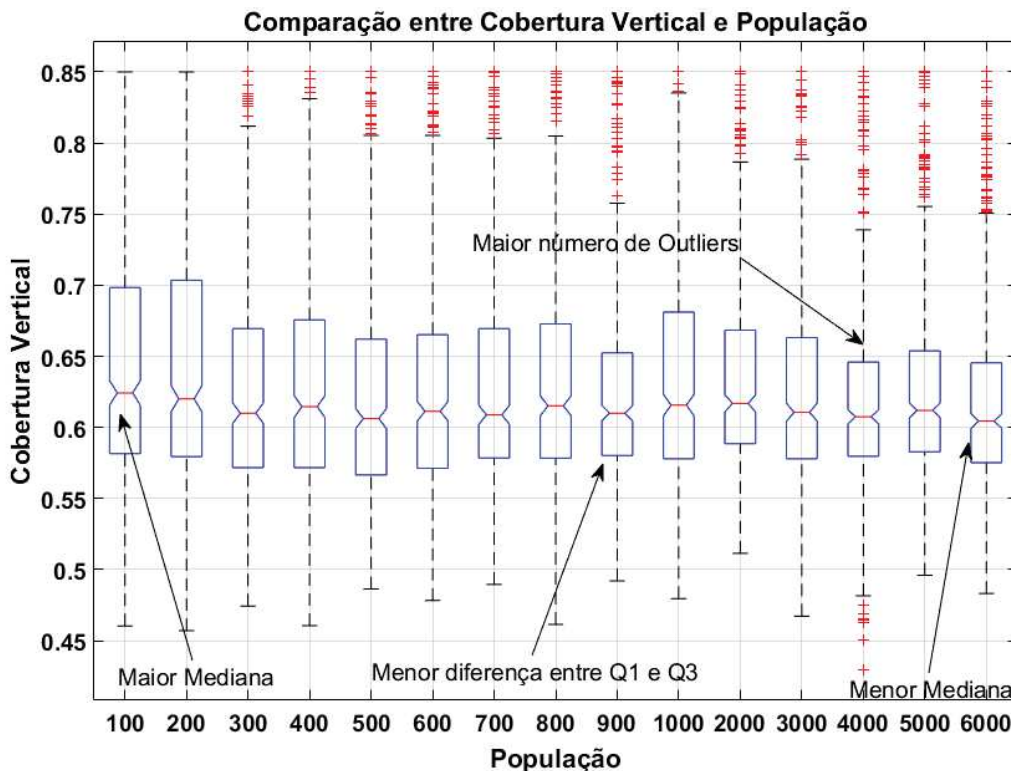
O terceiro objetivo a ser investigado é a cobertura vertical mostrada na Tabela ANOVA 7 e na Figura 30. Novamente, temos o fator-p abaixo dos 5%, sendo grupos distintos. Neste exemplo os melhores candidatos são a população de 100 indivíduos, que teve a maior mediana e a população de 900 que teve a menor diferença entre os quartis. A população de 4000 teve o maior número de *outliers*, enquanto a população de 6000 teve a menor mediana. Também é considerado que todos tiveram coberturas entre 55% e 75%.

Tabela 7 – Tabela ANOVA de *Cobertura<sub>Vertical</sub>* com a variação da população

Tabela ANOVA					
	Df	Sum Square SS	Mean Square MS	Valor <i>F</i>	Pr(>F)
População	14	0,3691	0,026364	3,4633	1,14e-05
Resíduos	6735	51,2691	0,0076123		

O próximo objetivo é o tempo de processamento, que pode ser dividido na avaliação de ciclos de CPU na Tabela ANOVA 8 e nas Figuras 31 e 32. Um

Figura 30 – Boxplot entre cobertura horizontal e a população.



Fonte: Elaborado pelo Autor.

ciclo é tecnicamente um pulso sincronizado por um oscilador interno, mas para nossos propósitos, eles são uma unidade básica que ajuda a entender a velocidade de uma CPU (*Central Processing Unit* - Unidade de Central de Processamento). Durante cada ciclo, bilhões de transistores dentro do processador entram em corte e saturação. Uma CPU com velocidade de *clock* de 3,2 GHz executa 3,2 bilhões de ciclos por segundo. Uma maneira mais precisa de medir o desempenho de um programa é calculando os ciclos de CPU utilizados para a execução do mesmo, já diferentes tipos de instruções podem levar mais ou menos de um ciclo para serem concluídos. Além disso, essa contagem retira os impactos da qualidade do *hardware* e dos processos que o computador está executando em segunda mão. Apesar disso, também foi avaliada a questão de processamento em segundos, mostrado na Tabela ANOVA 9 e nas Figuras 33 e 34 para melhor visualizar os resultados em comparação com o tempo disponível de missão. Em ambas as tabelas ANOVA,

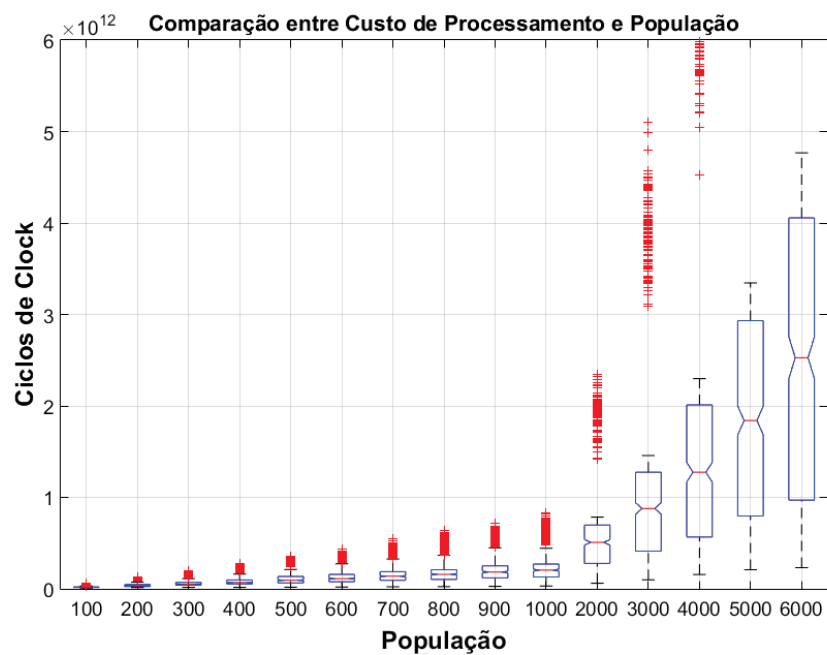


o fator-p é igual a zero, o que indica nenhuma relação entre as medianas. Isto é esperado já que os algoritmos meta-heurísticos tem seu tempo de processamento ligado diretamente ao tamanho de sua população. Outro fator a adicionar é que as populações acima de 2000 indivíduos tiveram custo de processamento maior que o tempo máximo da missão de 10 minutos. Devido a isso, devem ser eliminadas para o CPP *online*, mesmo que os outros objetivos forem melhores. Dessa forma, foram adicionadas as Figuras 32 e 33 para visualizar o tempo de processamento entre as populações menores, permitindo destacar esses dados. Percebe-se que o processo de crescimento é uma exponencial, devendo considerar essas informações para a próxima análise.

Tabela 8 – Tabela ANOVA de tempo de processamento em ciclos de clock com a variação da população.

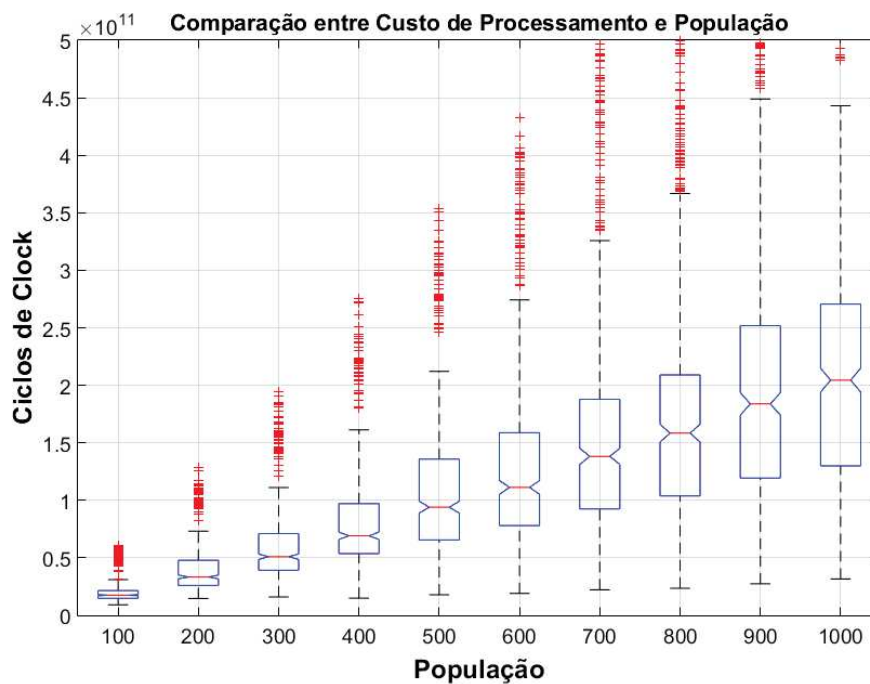
Tabela ANOVA					
	Df	Sum Square SS	Mean Square MS	Valor $F$	Pr(>F)
População	14	1,16e+28	8,34e+26	270,8091	0
Resíduos	6735	2,07e+28	3,08e+24		

Figura 31 – Boxplot entre Tempo de Processamento em ciclos de CPU e todas as populações.



Fonte: Elaborado pelo Autor.

Figura 32 – Boxplot entre Tempo de Processamento em ciclos de CPU e as populações menores.

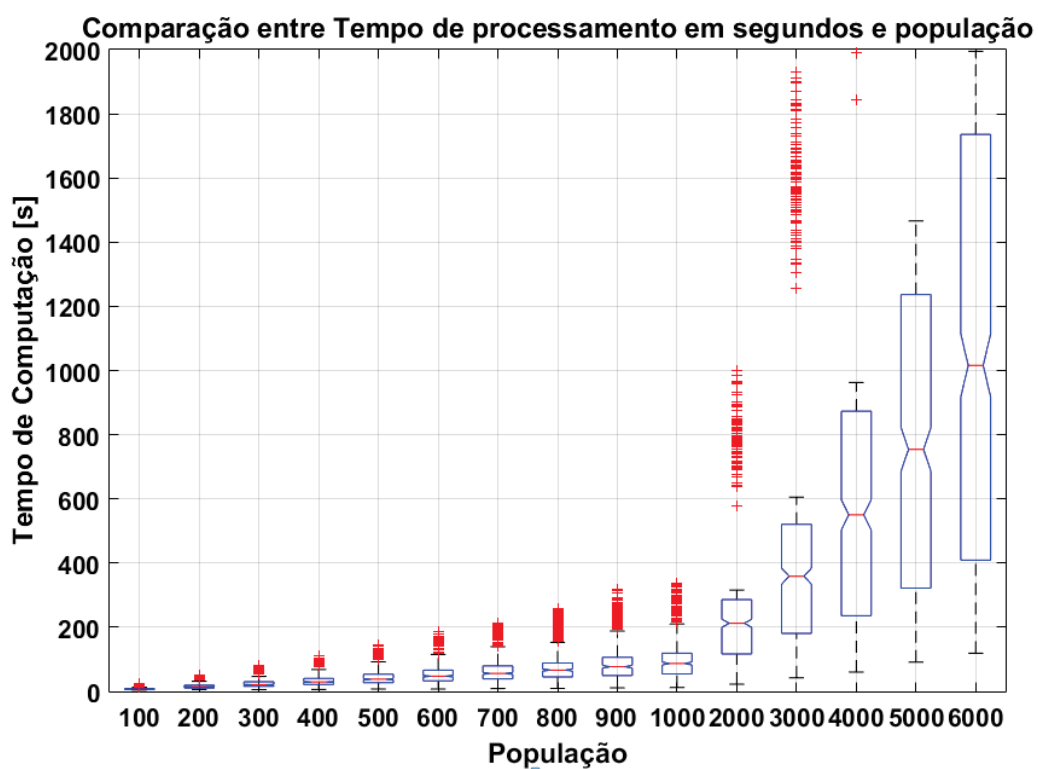


Fonte: Elaborado pelo Autor.

Tabela 9 – Tabela ANOVA de Tempo de Processamento em segundos com a variação da população.

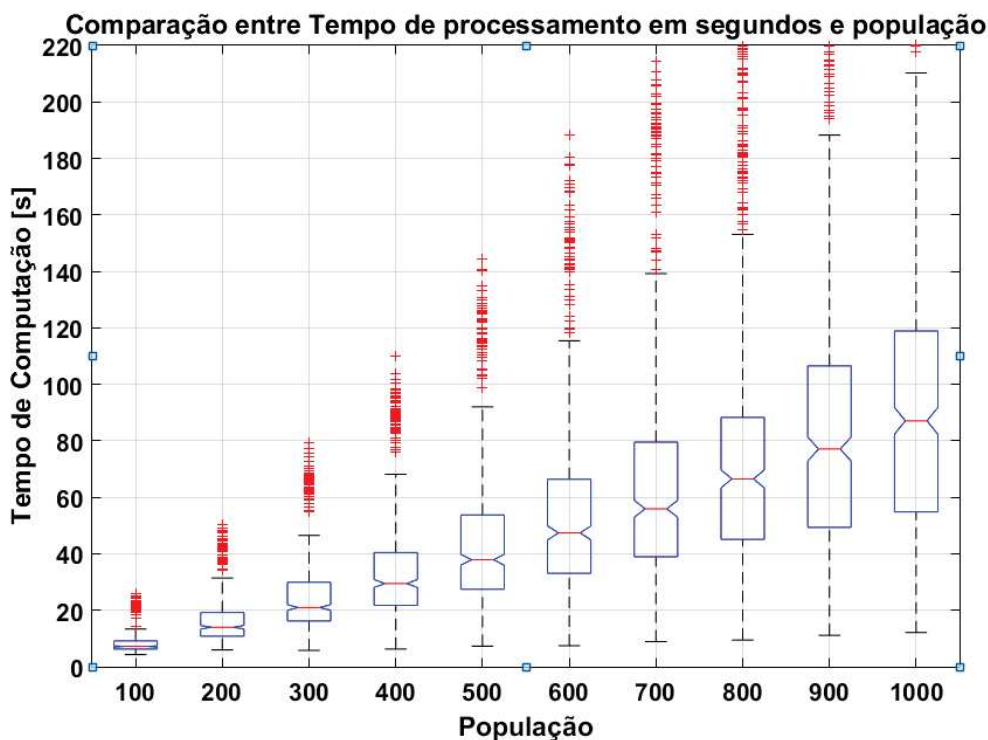
Tabela ANOVA					
	Df	Sum Square SS	Mean Square MS	Valor $F$	Pr(>F)
População	14	2,07e+9	1,48e+8	269,3885	0
Resíduos	6735	3,71e+9	5,51e+5		

Figura 33 – Boxplot entre Tempo de Processamento em segundos e todas as populações.



Fonte: Elaborado pelo Autor.

Figura 34 – Boxplot entre Tempo de Processamento em segundos e as populações menores.



Fonte: Elaborado pelo Autor.

Para a escolha da população utilizada nos próximos testes foi realizada a remoção dos piores resultados presentes nos objetivos de tempo, cobertura horizontal e vertical, além do custo computacional. Considerando o tempo de processamento foram removidas as populações acima de 2000 indivíduos. O grupo de 100 indivíduos e o de 200 foram retirados pelo pior resultado na cobertura horizontal e no tempo de missão. A cobertura vertical removeria os grupos de 4000 e 6000, já removidos pelo tempo de processamento. Desta forma a análise final tem os grupos com as populações entre 300 e 2000 indivíduos. Para esses grupos, foram criados novamente Tabelas ANOVA como mostrado nas Tabelas 10, 11 e 12, respectivamente para tempo de missão, cobertura horizontal e cobertura vertical, os três objetivos propostos. Destaca-se que todos os fatores p foram acima de 5% (12%, 7% e 14%, respectivamente), indicando que as médias desses valores são muito próximas, não sendo distintas. Dessa forma, foi escolhida a população

de 300 indivíduos para as próximas etapas, pois tem o menor custo computacional entre as populações restantes.

Tabela 10 – Tabela ANOVA de tempo de missão com a variação da população reduzida.

Tabela Anova					
	Df	Sum Square SS	Mean Square MS	Valor $F$	Pr(>F)
População	6	45,1346	7,5224	1,6612	0,12644
Resíduos	3143	14232,3077	4,5283		

Tabela 11 – Tabela Anova de cobertura horizontal com a variação da população reduzida.

Tabela Anova					
	Df	Sum Square SS	Mean Square MS	Valor $F$	Pr(>F)
População	6	18,5696	3,0949	1,9321	0,07205
Resíduos	3143	5034,4986	1,6018		

Tabela 12 – Tabela Anova de cobertura vertical com a variação da população reduzida.

Tabela Anova					
	Df	Sum Square SS	Mean Square MS	Valor $F$	Pr(>F)
População	6	0,078518	0,013086	1,5894	0,14599
Resíduos	3143	25,878	0,0082335		

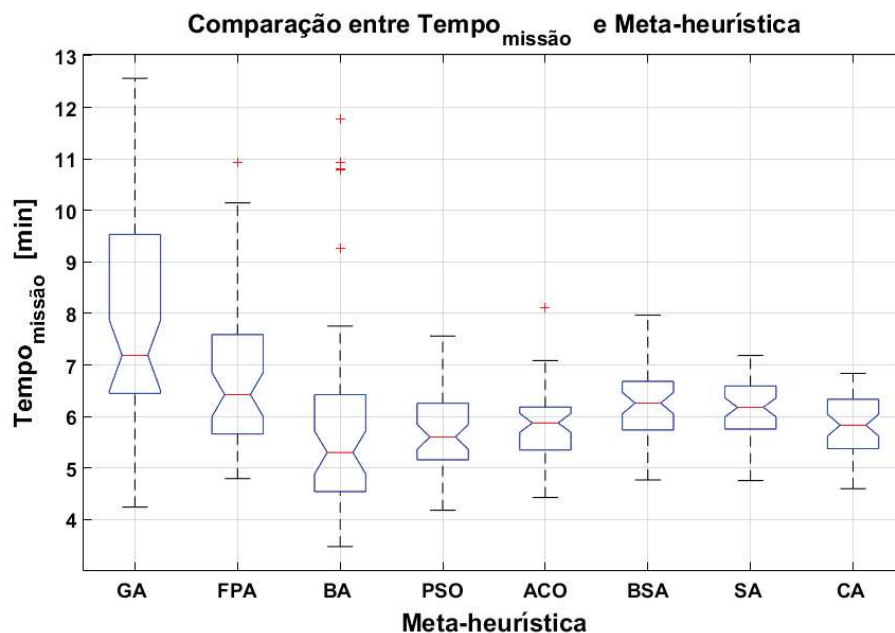
A partir da população de 300 indivíduos, a próxima etapa de avaliação é qual meta-heurística tem o melhor comportamento em relação ao problema proposto. As meta-heurísticas propostas foram: Algoritmo genético (GA), *Flower Pollination Algorithm* (FPA), *Bat Algorithm* (BA), *Particle swarm optimization* (PSO), *Ant Colony Optimization* (ACO), *Bird Swarm Algorithm* (BSA), *Simulated Annealing* (SA) e *Culture Algorithm* (CA). O critério de avaliação são os mesmos que os propostos para as populações com o adicional do *fitness* geral.

O primeiro objetivo é o tempo de missão como mostrado na Tabela ANOVA 13 e a Figura 35. Percebe-se que o fator-p é muito abaixo dos limites estabelecidos, o que indica que as médias são distintas. No caso se tem o pior tempo de missão entregue pelo GA, com maior máximo, maior mediana e maior Q3. O que teve menor mediana foi o BA, enquanto o que teve menor diferença entre os quartis foi o SA. Neste caso, como o objetivo é reduzir o tempo de missão o melhor algoritmo foi o BA, apesar da presença de *outliers*.

Tabela 13 – Tabela Anova de tempo de missão com a variação da meta-heurística.

Tabela Anova					
	Df	Sum Square SS	Mean Square MS	Valor $F$	Pr(>F)
Meta-heurística	7	197,55	28,22	17,29	5,71e-20
Resíduos	392	639,61	1,63		

Figura 35 – Boxplot entre tempo de missão e as meta-heurísticas com 300 de população.



Fonte: Elaborado pelo Autor.

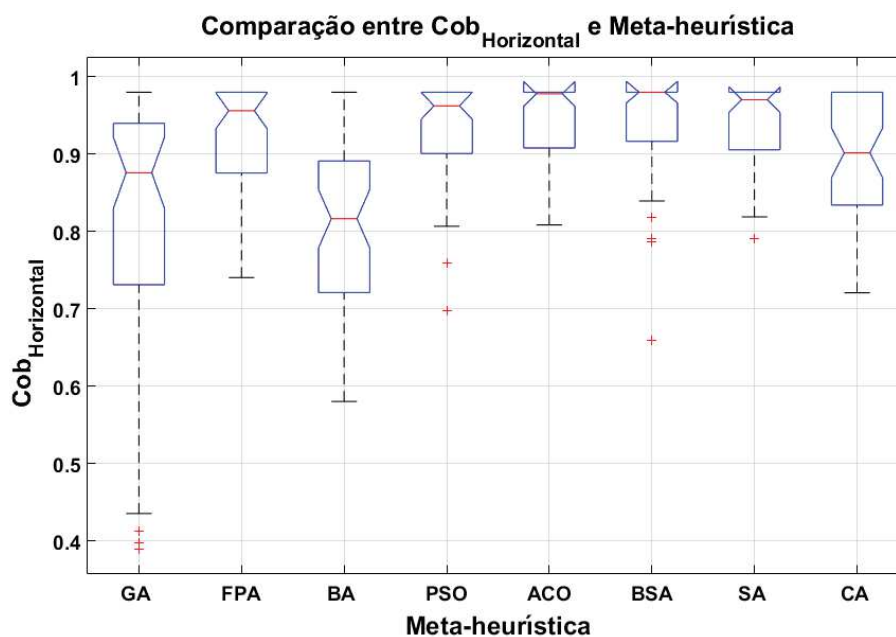
O segundo objetivo é a cobertura horizontal como mostrado na Tabela

ANOVA 14 e a Figura 36. O fator-p foi abaixo do limiar novamente, porém mais próximo que no tempo de missão. Neste caso temos a melhor performance do BSA, com maior mediana e menor distância entre os quartis. O que teve pior desempenho foi o BA tendo a menor interseção entre as imagens horizontalmente. Todas as medianas estão acima de 80%, mostrando eficácia quanto a esse objetivo.

Tabela 14 – Tabela Anova da cobertura horizontal com a variação da meta-heurística.

Tabela Anova					
	Df	Sum Square SS	Mean Square MS	Valor $F$	Pr(>F)
Meta-heurística	7	1,18	0,16	20,25	3,29e-23
Resíduos	392	3,26	8,3349e-3		

Figura 36 – Boxplot entre cobertura horizontal e as meta-heurísticas com 300 de população.



Fonte: Elaborado pelo Autor.

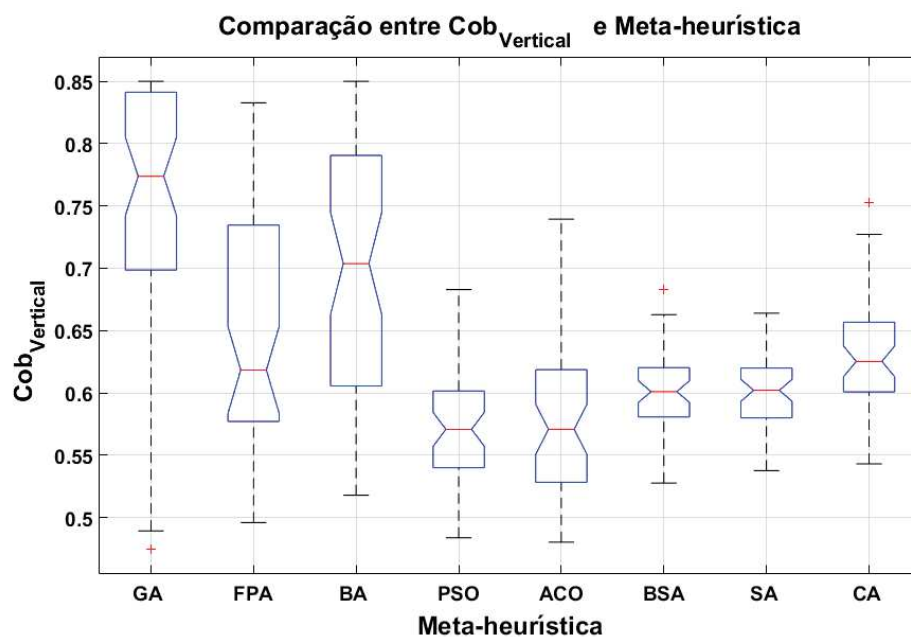
O terceiro objetivo é a cobertura vertical, objetivo que teve maior discrepância entre os algoritmos mostrados na Tabela ANOVA 15 e na Figura 37. Como

é ligado diretamente a uma variável inteira de  $Num_{way}$ , o resultado teve valores que variaram de 50% no ACO até 85% no GA. O fator-p pode ser considerado zero pela sua ordem de grandeza, mostrando que os grupos tiveram resultados bem distintos.

Tabela 15 – Tabela Anova da cobertura vertical com a variação da meta-heurística.

Tabela Anova					
	Df	Sum Square SS	Mean Square MS	Valor $F$	Pr(>F)
Meta-heurística	7	1,4157	2,0e-1	41,5	1,25e-43
Resíduos	392	1,9103	4,87e-3		

Figura 37 – Boxplot entre cobertura vertical e as meta-heurísticas com 300 de população.



Fonte: Elaborado pelo Autor.

Como é uma otimização multi-objetivo onde tiveram resultados diferentes entre coberturas e tempo foi necessário analisar o *fitness* completo dado pela Equação 3.21. Dessa forma, é possível analisar a performance olhando a maximização das coberturas e a minimização do tempo de missão de forma conjunta. A Tabela

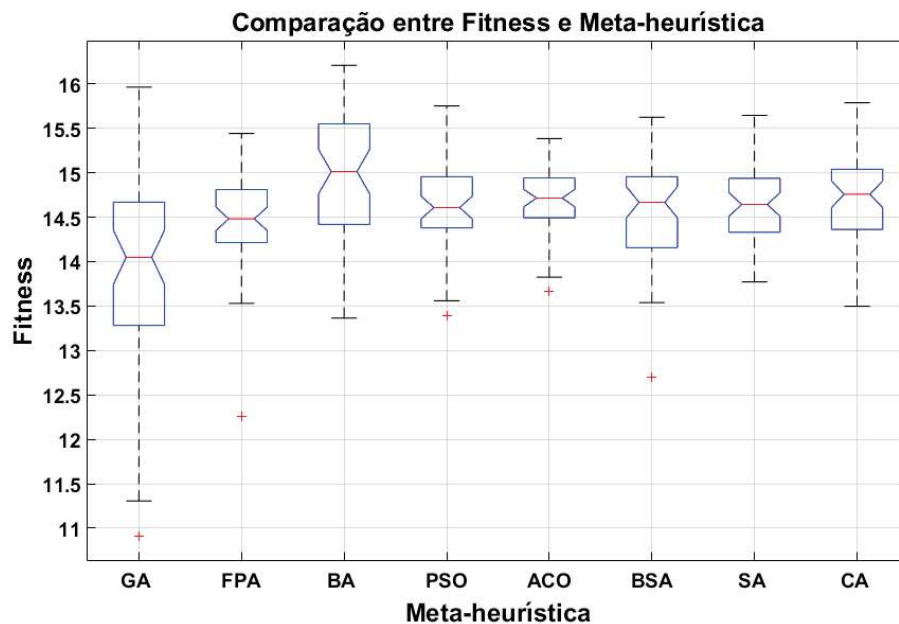


ANOVA 16 e a Figura 38 mostram esses resultados. O fator-p teve valor abaixo do limiar de 5%, mostrando que as médias são diferentes. O BA teve o maior MAX com valores de 16 de *fitness*, sendo que o máximo é 20. O BA teve MIN mais destacado, próximo a 11,5. Considerando a distância entre os quartis o que teve melhor resultado foi o ACO, com valores entre 14,5 e 15. O que tiveram os maiores medianas foram o BA. Assim, foi escolhido o BA como algoritmo, já que mesmo mais imprevisível que o ACO, os valores dentro dos quartis ainda são maiores que os do ACO.

Tabela 16 – Tabela Anova de *fitness* com a variação da meta-heurística

Tabela Anova					
	Df	Sum Square SS	Mean Square MS	Valor $F$	Pr(>F)
Meta-heurística	7	197,5548	28,2221	17,2965	5,7191e-20
Resíduos	392	639,6128	1,6317		

Figura 38 – Boxplot entre Fitness e as meta-heurísticas com 300 de população.



Portanto, com relação ao tempo de processamento, o tamanho da população

foi de 300 indivíduos e em relação ao *fitness*, a escolha foi do BA. Apesar da melhor performance nos resultados encontrados, existe a necessidade da otimização de todos os hiper-parâmetros, além que afirmar que uma meta-heurística é melhor que outra não é correto pelo teorema do *no free lunch* (NFL) [Wolpert e Macready 1997]. A segunda superfície investigada é um talude inclinado em rampa, com diferentes níveis planos, tal como mostrado na Figura 39. Esse talude teve sua leitura dos *laser* mostrada na Figura 40 e a missão criada na Figura 41. O terceiro talude é um talude com sua região inversa ao primeiro talude como apresentado na Figura 42. Figura 43 mostra a leitura dos sensores enquanto a Figura 44 mostra a missão para esta superfície. O quarto talude é a adição de um obstáculo no meio do primeiro talude, como visto na Figura 45, sendo o sensoriamento dado pela Figura 46 e a missão pela Figura 47. Percebe-se que as missões do algoritmo seguem os mais diversos formatos de superfície, mantendo a mesma distância e a mesma interseção entre todos os pontos.

Figura 39 – Imagem do Talude 2.

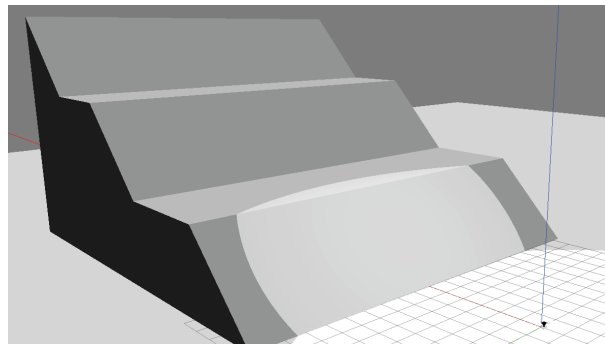


Figura 40 – Leituras do Laser para o Talude 2.

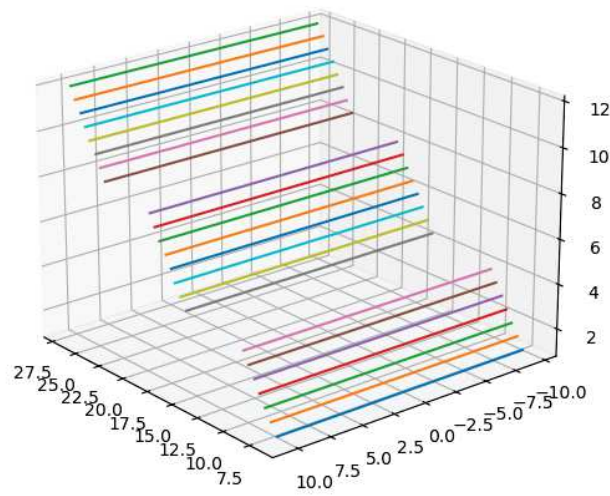


Figura 41 – Missão para o Talude 2.

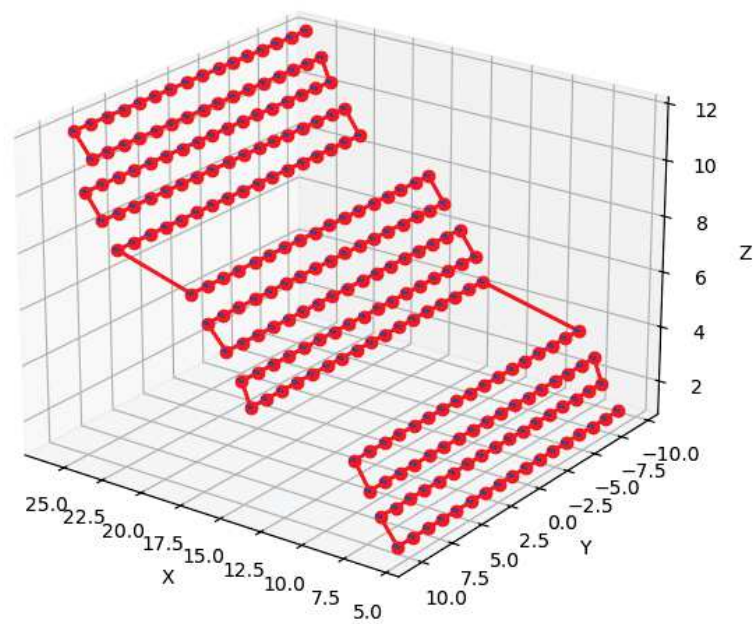


Figura 42 – Imagem do Talude 3.

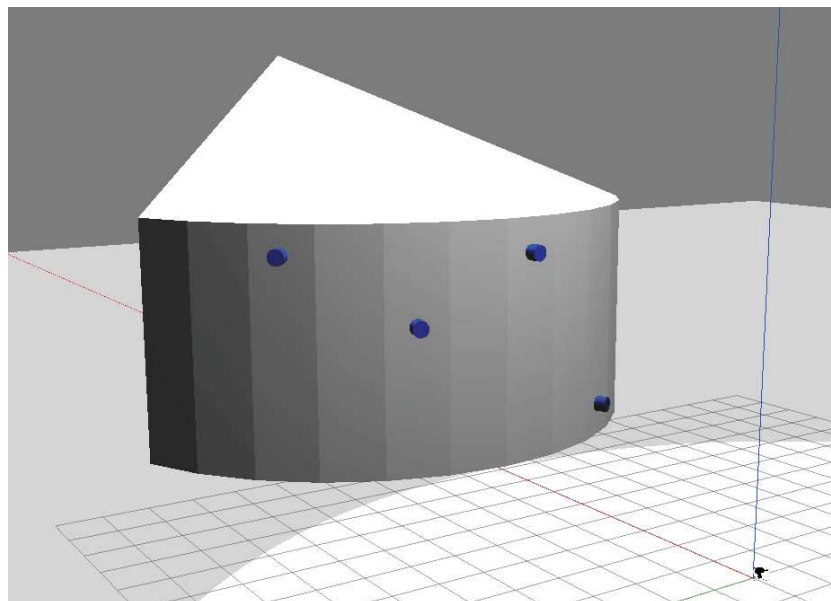


Figura 43 – Leituras do Laser para o Talude 3.

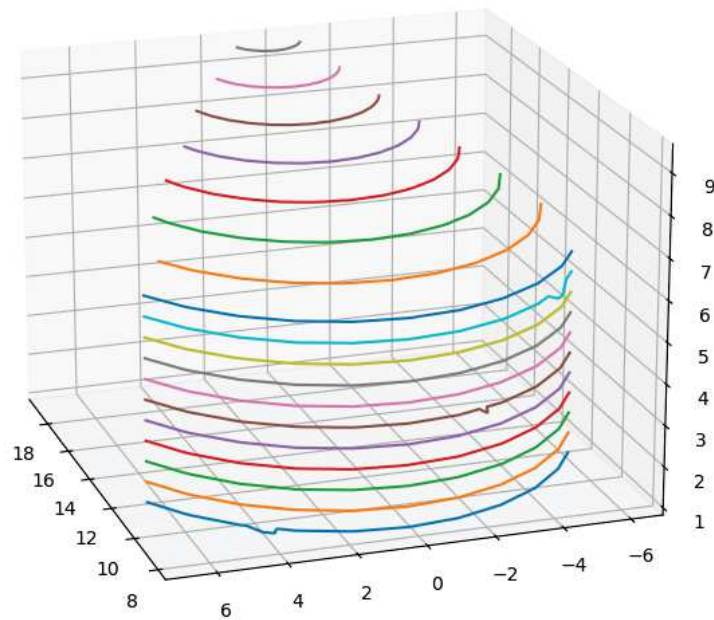


Figura 44 – Missão para o Talude 3.

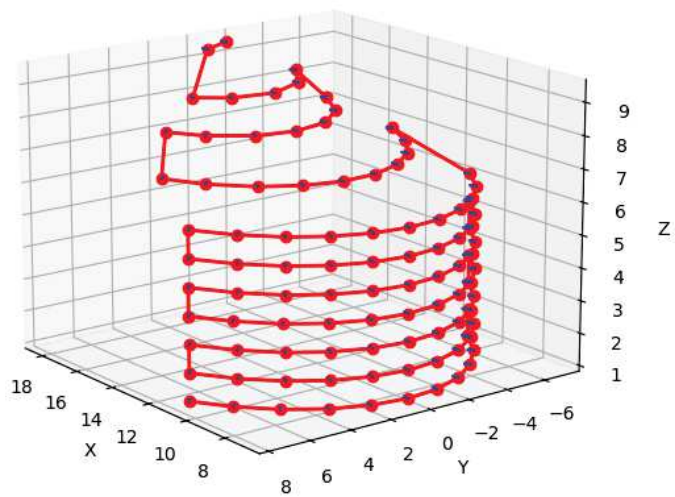


Figura 45 – Imagem do Talude 4.

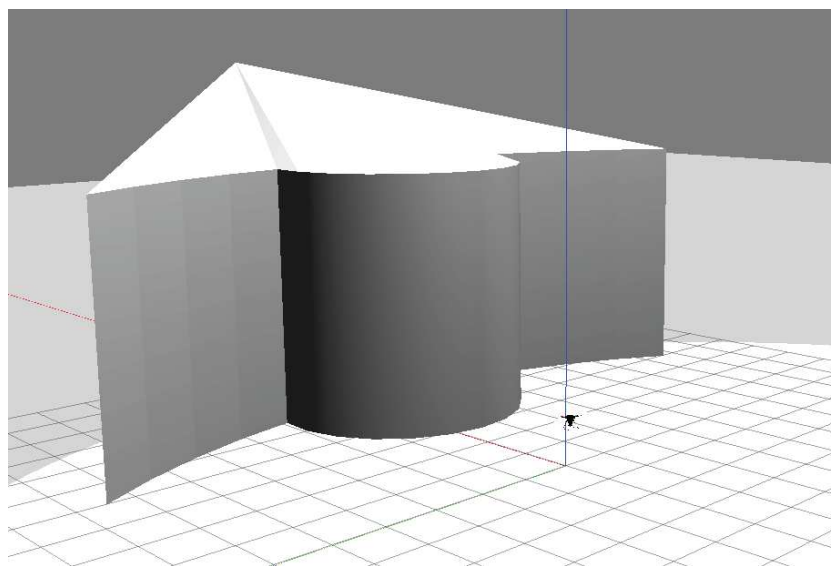


Figura 46 – Leituras do Laser para o Talude 4.

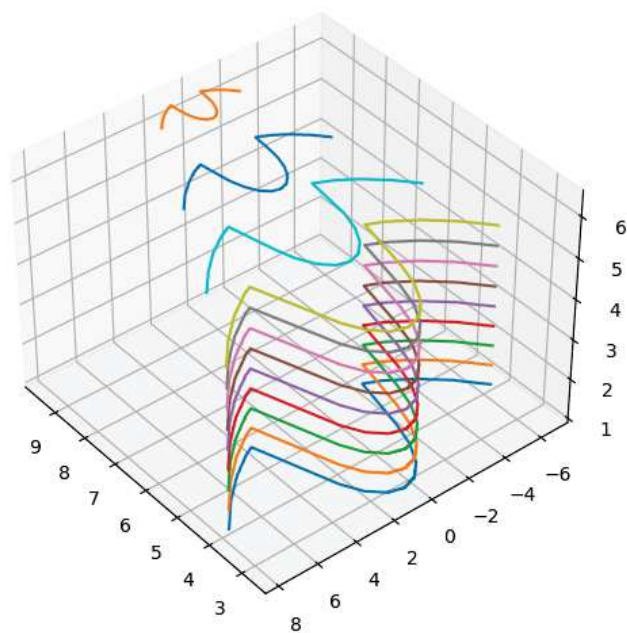
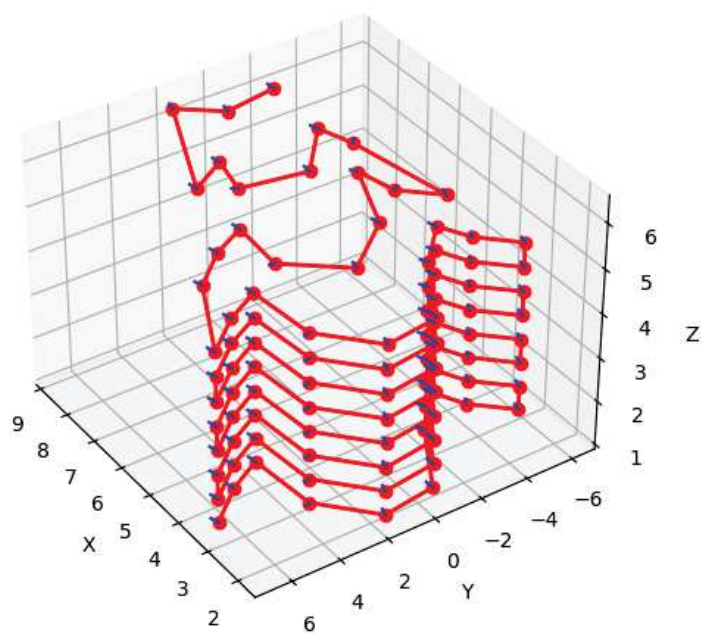


Figura 47 – Missão para o Talude 4.



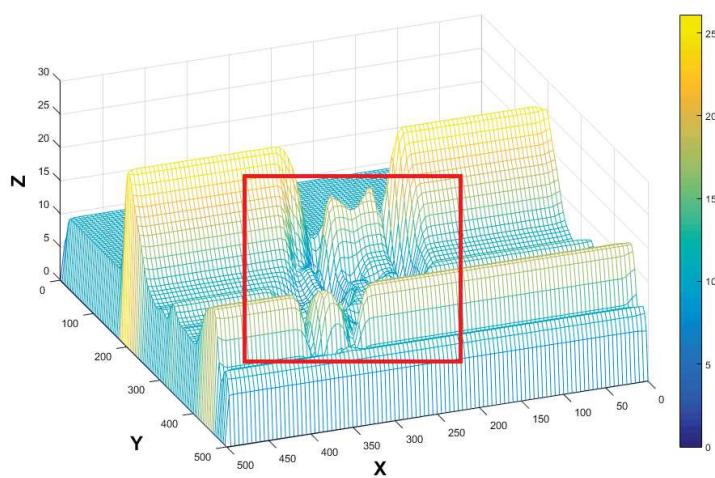
Outra fonte de entrada das informações do ambiente é através de DEM. Essas missões são em áreas maiores, onde a cobertura do VANT ocorre em voos superiores à região. Na Figura 48, é apresentada a visualização de uma região criada em simulador para avaliação do algoritmo nestas regiões. Na Figura 49, é mostrado o DEM da região. Note que o quadrado vermelho em destaque é equivalente à Figura 48. Percebe-se que é uma planície no Y variando de 0 metros até próximo a 200 metros, depois a presença de uma montanha, um vale onde tem as residências em destaque e depois mais uma pequena montanha. É uma região de 250.000 m<sup>2</sup>, sendo considerado uma região grande para inspeção. A Figura 50 mostra a missão gerada para esse mapa com tempo de navegação de 20 minutos. A distância aos objetos, sendo neste caso o solo ou elementos sobre eles, foi de 42 metros e as coberturas horizontais e verticais foram de 78% e 52%, respectivamente. A missão teve duração de 18 minutos. Neste caso, a maior parte do tempo a câmera esteve voltada para baixo, variando a mesma nas montanhas e na região residencial.

Figura 48 – Região no Simulador Gazebo-ROS para missões com DEM.



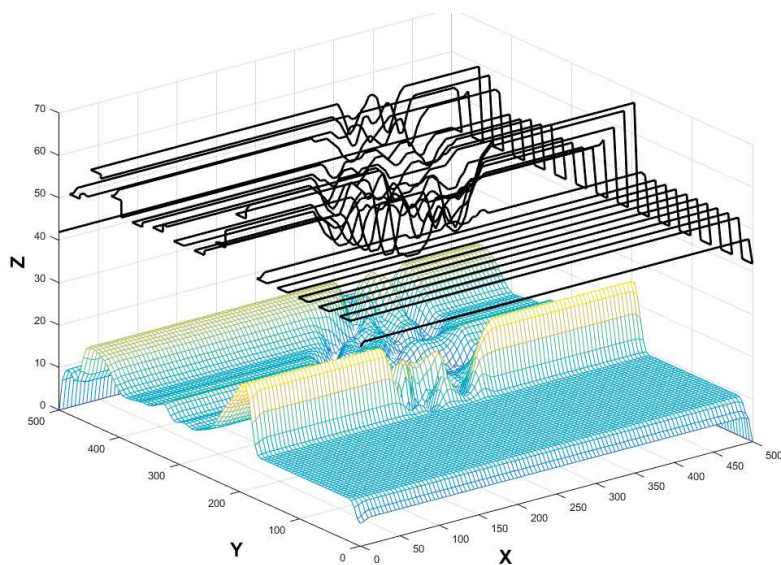
Fonte: Elaborado pelo Autor.

Figura 49 – DEM da região simulada. Em destaque a região das casas.



Fonte: Elaborado pelo Autor.

Figura 50 – Missão criada para um mapa de DEM.



Fonte: Elaborado pelo Autor.



#### 4.1.3 Identificação de pontos de interesse e missões dinâmicas

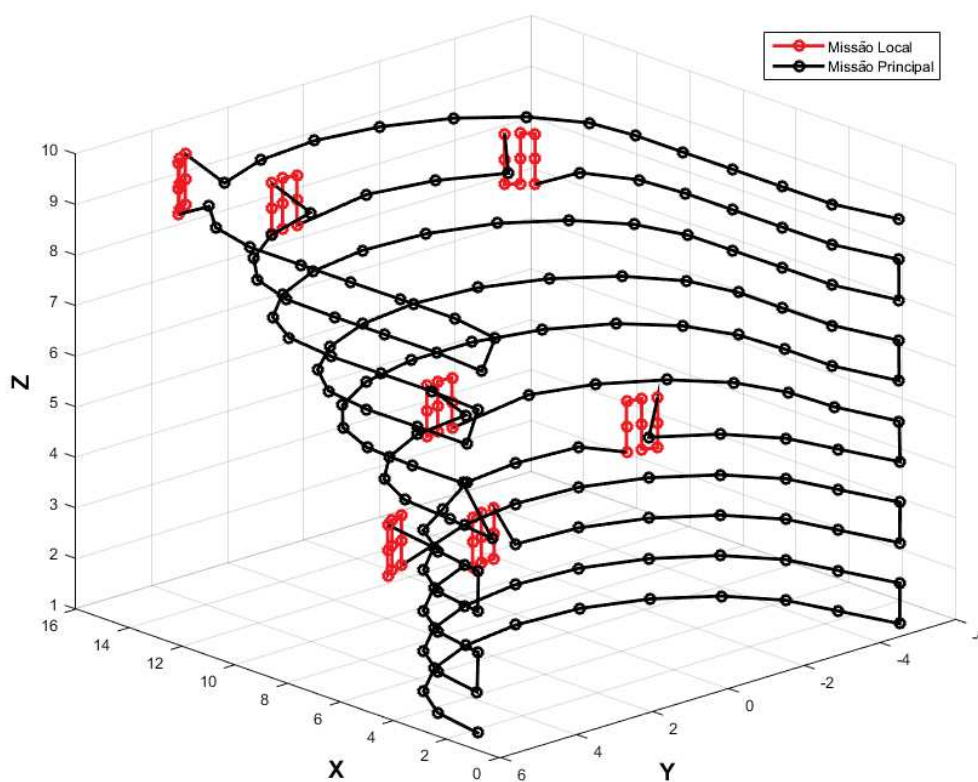
Uma proposta que foi destacada na Seção 3.4 é a identificação dinâmica de objetivos. Para demonstrar a aplicação dessa parte da metodologia escolheu-se o mesmo talude 1 da Figura 24 com os pontos em azul como pontos de interesse. Figura 51 mostra a missão realizada no simulador e a Tabela 17 com o resumo dos resultados. Na primeira linha da Tabela 17 temos a primeira otimização, onde o tempo disponível era de 20 minutos. Com esse tempo, foi realizado o planejamento de uma missão com distância de 1,68 metros entre os pontos, distância ao talude de 2,06 metros e 11 níveis de missão. O tempo planejado para toda a missão era de 11 minutos e 26 segundos. Ao executar o primeiro translado, com tempo planejado de 1 minuto e 40 segundos, ocorreu a identificação do objetivo dinâmico no 13º ponto. Nesse momento o drone aproxima de 1 metro de distância ao talude e executa a missão em vermelho, como mostrado na Figura 51. Após o último ponto da missão local, o mesmo retorna ao próximo ponto da missão original, continuando sua execução. Pela presença dessa missão local e de outras questões, como diferença de aceleração e freada do VANT simulado, a missão que deveria durar 1 minuto e 40 segundos durou 2 minutos e 10 segundos, diminuindo o tempo disponível para a próxima etapa de otimização.

A segunda etapa de otimização recebe agora um tempo máximo de 17min50seg, sendo o que sobrou do tempo total disponível menos o tempo gasto do primeiro translado. Neste translado a missão planejada foi de 1min30seg, já que no talude acontece a redução da distância percorrida pela inclinação do mesmo. A cobertura horizontal teve um decréscimo pelo aumento de  $D_{\min}$  sem alteração proporcional em  $Dist_H$ . A missão reduziu em 1 o número de translados, uma vez que a região a ser investigada também diminuiu com a execução da primeira missão. Neste translado ocorreu duas missões dinâmicas, nos pontos 30 e 33 da missão global completa.

Nas próximas etapas de otimização, ocorreram sempre o mesmo processo: a redução do tempo disponível com o tempo gasto nas missões anteriores e a readequação da missão com a área a ser investigada. Percebe-se que apesar da variação natural pela otimização meta-heurística, os valores de cobertura horizontal sempre ficaram acima de 70%, enquanto a vertical acima de 69%. Neste caso

o número de translados diminuiu em uma unidade em cada execução da meta-heurística com a diminuição da área vistoriada, o que manteve os valores das coberturas verticais próximos. O tempo final disponível ainda para a execução da missão foi de 4 minutos e 9 segundos, sendo uma folga para eventuais problemas. Com uma média de 25 a 30 segundos por missão local, é sempre verificado ao identificar o ponto se o tempo de sobra é superior a 1 minuto. Assim, evitando possíveis quedas ao executar a missão local e não ter bateria suficiente para o resto da missão.

Figura 51 – Missão com identificação dinâmica executada no simulador Gazebo-ROS.



Fonte: Elaborado pelo Autor.

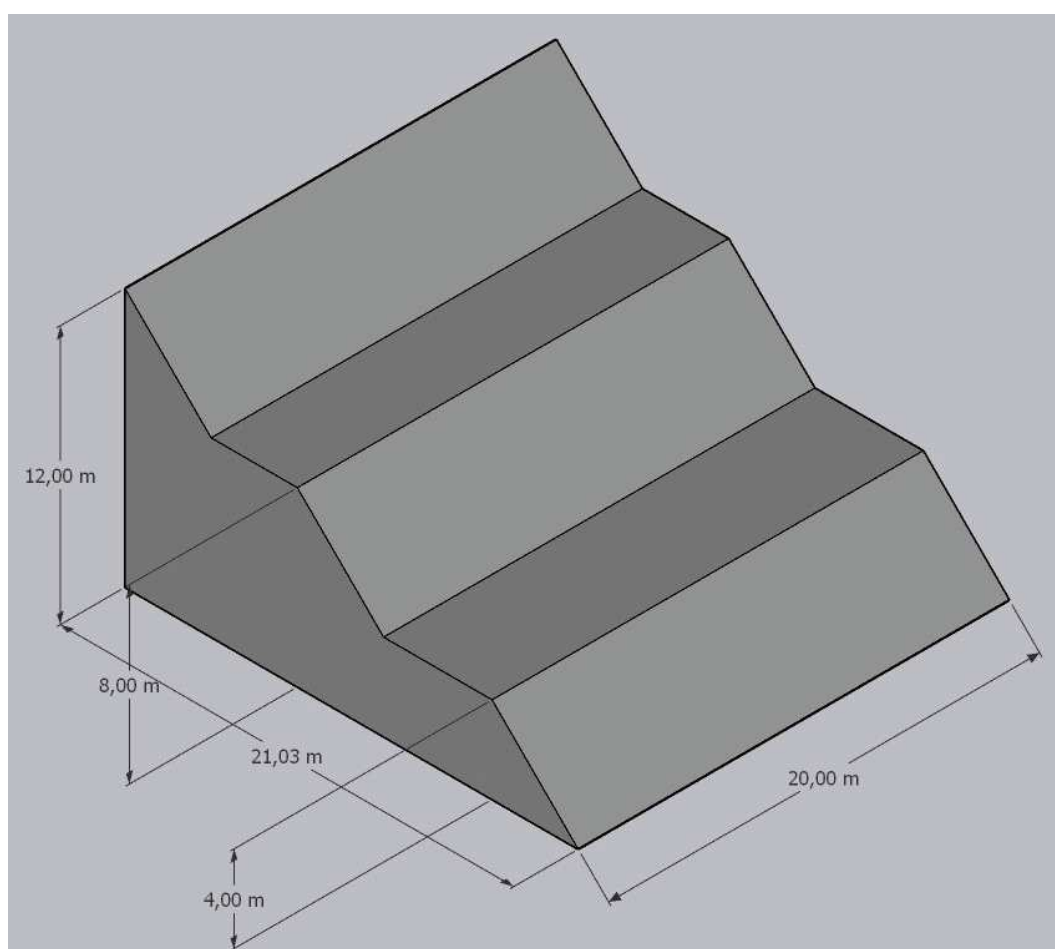
Tabela 17 – Resultados da missão dinâmica com identificação de pontos de interesse.

Otim.	Tempo Disponível [min:seg]	Tempo Planejado Total [min:seg]	Tempo Planejado Translado [min:seg]	Cob. Hor. [%]	Cob. Vert. [%]
1	20:00	12:13	01:40	76,5	69,4
2	17:50	10:33	01:30	71,8	70,0
3	15:31	09:03	01:21	74,3	69,9
4	14:01	07:42	01:12	74,4	70,1
5	12:46	06:29	01:05	75,9	71,1
6	11:36	05:24	00:59	71,4	69,4
7	09:59	04:25	00:53	73,5	69,6
8	08:31	03:32	00:54	74,3	70,1
9	07:09	02:38	00:52	75,1	71,8
10	05:54	01:46	00:52	75,0	71,1
11	05:11	00:54	00:54	74,4	70,6
Otim.	$D_{\min}$ [m]	$Dist_H$ [m]	NumWay	Tempo Executado Translado [min:seg]	MiniMissão
1	1,68	2,06	11	02:10	1
2	2,05	2,1	10	02:19	2
3	1,86	2,09	9	01:30	0
4	1,87	2,11	8	01:14	0
5	1,82	2,18	7	01:10	0
6	2,04	2,06	6	01:37	1
7	1,9	2,07	5	01:28	1
8	1,88	2,11	4	01:22	1
9	1,92	2,23	3	01:15	1
10	1,89	2,18	2	00:56	0
11	1,9	2,14	1	00:58	0

#### 4.1.4 Comparação entre tempos máximos permitidos e ganhos de *fitness*

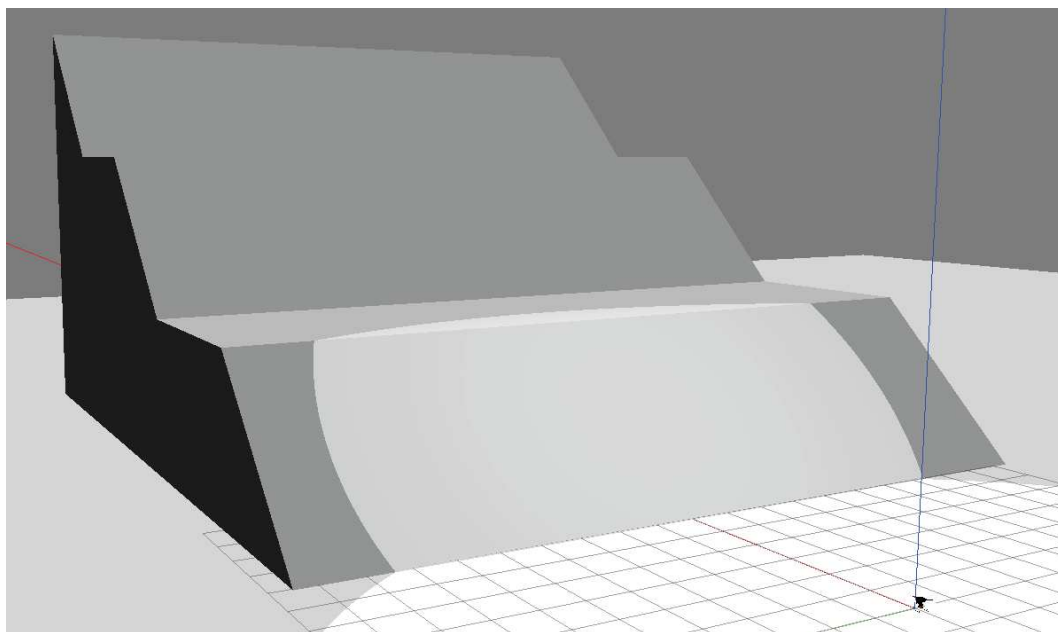
O segundo teste de simulação teve como objetivo comparar as questões de tempo máximo entregue para a missão. Para este teste a escolha foi de um mundo virtual com um talude com rampas em diferentes níveis, como visto na Figura 52. Esse talude de 520 m<sup>2</sup> foi escolhido para avaliar o impacto de variar os tempos disponíveis da missão principal, se baseando em taludes construídos com o mesmo formato. Na Figura 53 mostra a inserção deste talude num mundo virtual no simulador Gazebo-ROS.

Figura 52 – Talude para teste variação do tempo máximo de missão.



Fonte: Elaborado pelo Autor.

Figura 53 – Talude para teste variação do tempo máximo de missão no simulador Gazebo-ROS.

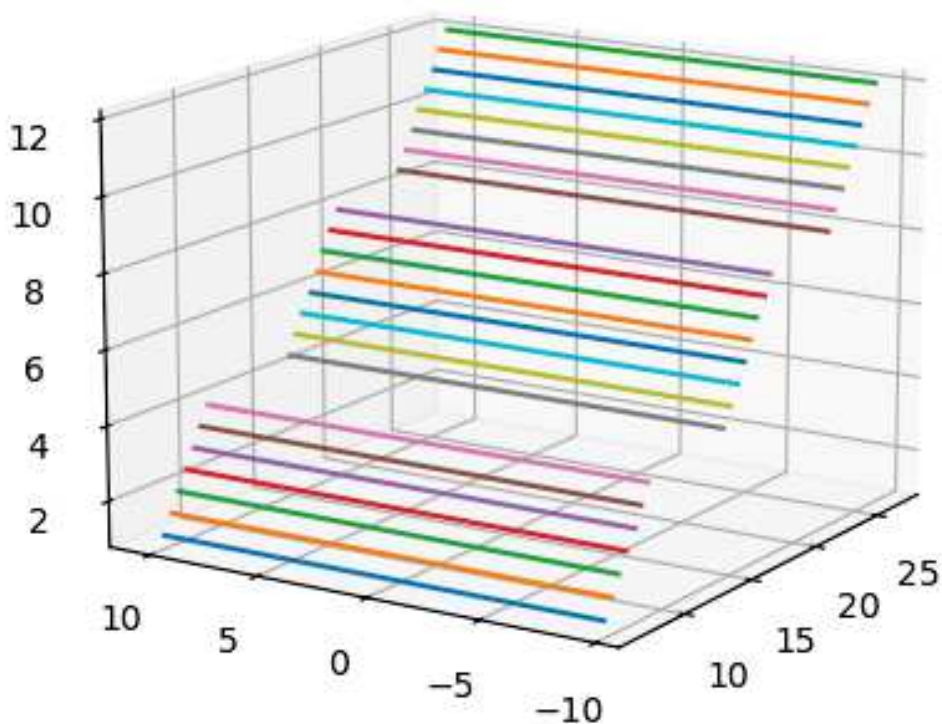


Fonte: Elaborado pelo Autor.

Iniciando a simulação, tem-se a leitura dos sensores laser demonstradas na Figura 54 após a aplicação do filtro de posições próximas. Este processo de captação das leituras e da execução do filtro demorou 48 segundos passando estes dados para a meta-heurística. O VANT simulado consegue identificar a altura da superfície e seu formato sem nenhuma informação previamente disponibilizada. Esse processo também pode ser entregue ao VANT em solo a partir de uma nuvem de pontos georreferenciadas. Neste teste não foram introduzidos pontos de detecção dinâmica para avaliar o impacto do tempo máximo no modo de otimizar uma missão.

Para verificar o impacto de diversos tempos máximos nas estruturas das missões foram escolhidos três tempos de missão: 1 minuto, 5 minutos e 10 minutos. As missões geradas foram representadas nas Figuras 55, 56 e 57 com os tempos máximos respectivos e na Tabela 18 observam-se os resultados de forma resumida. O primeiro ponto a salientar é que a otimização meta-heurística conseguiu atender todas as missões com tempo previstos menores que os máximos exigidos. A missão

Figura 54 – Leituras do sensor laser após a aplicação do filtro de pontos próximos.



Fonte: Elaborado pelo Autor.

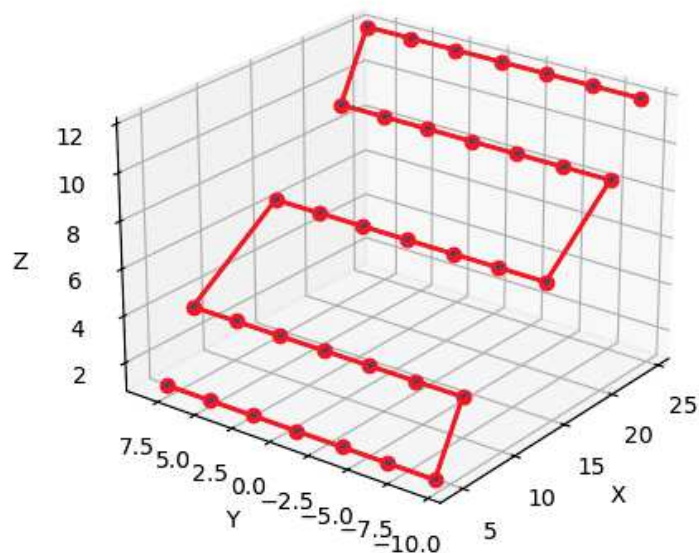
de 1 minuto se tornou a mais distante do talude e com menos translados horizontais, mas mesmo assim, ainda permitiu coberturas horizontais e verticais acima de 50%. Com o aumento do tempo disponível, a restrição temporal começou a ser relaxada permitindo a escolha de pontos mais próximos e com maior número de translados, aumentando a qualidade de uma futura reconstrução 3D. A missão de 10 minutos possui a menor distância ao talude permitida, já que por segurança, as missões devem ser sempre com 1 metro de distância da superfície para evitar danos ao equipamento com uma colisão. Quando a missão é muito próxima da superfície, a alteração entre os translados pode ocasionar colisões entre as superfícies. Dessa forma, como mostrado para a mesma missão com tempo máximo de 10 minutos na Figura 58, é necessário adicionar um ponto entre os translados de forma que o VANT deslocará na mesma altura do anterior, e depois, descer verticalmente. Essa opção sempre é testada no final de cada translado para verificar se a distância

Tabela 18 – Comparação da variação do tempo máximo em relação as missões.

Tempo Máximo [min]	$D_{\min}$ [m]	$Dist_H$ [m]	$Num_{way}$	Tempo da Missão [min:seg]	Cob. Horizontal [%]	Cob. Vertical [%]
1	2,89	3,81	5	0:46	66	52
5	1,91	1,8	10	2:17	69	64
10	2,05	1	15	5:12	52	60

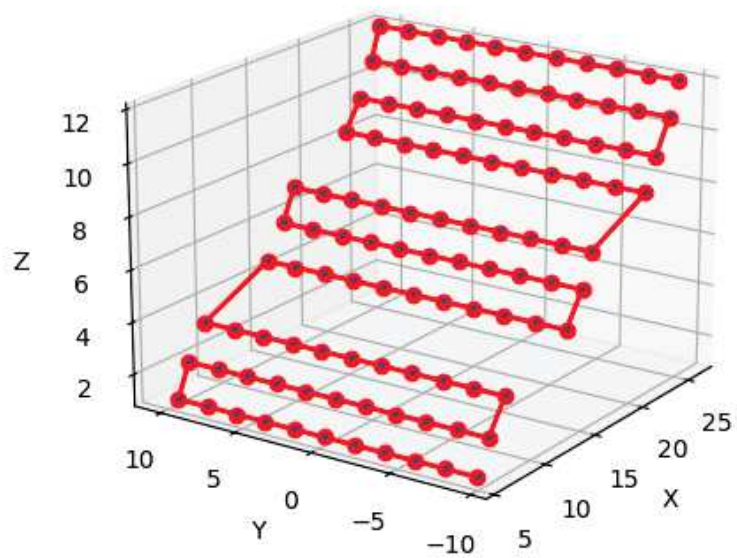
média entre os translados fica a menos de 1 metro da superfície, criando a missão em verde e removendo o caminho em vermelho.

Figura 55 – Missão de teste de tempo máximo igual a 1 minutos.



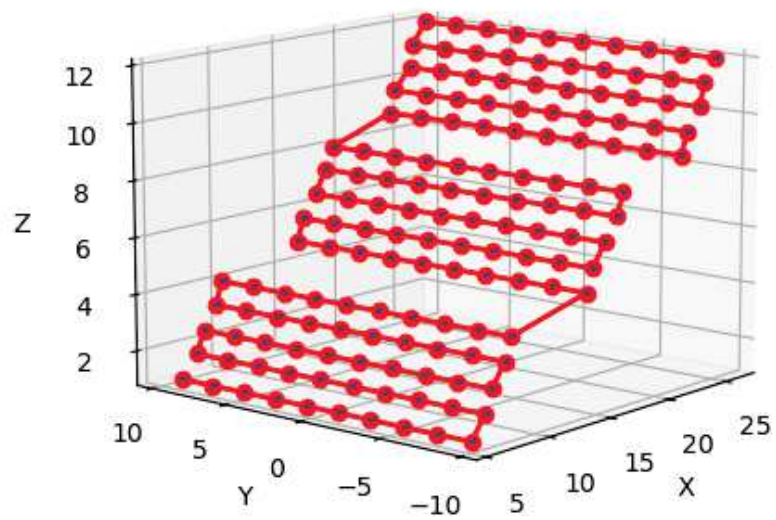
Fonte: Elaborado pelo Autor.

Figura 56 – Missão de teste de tempo máximo igual a 5 minutos.



Fonte: Elaborado pelo Autor.

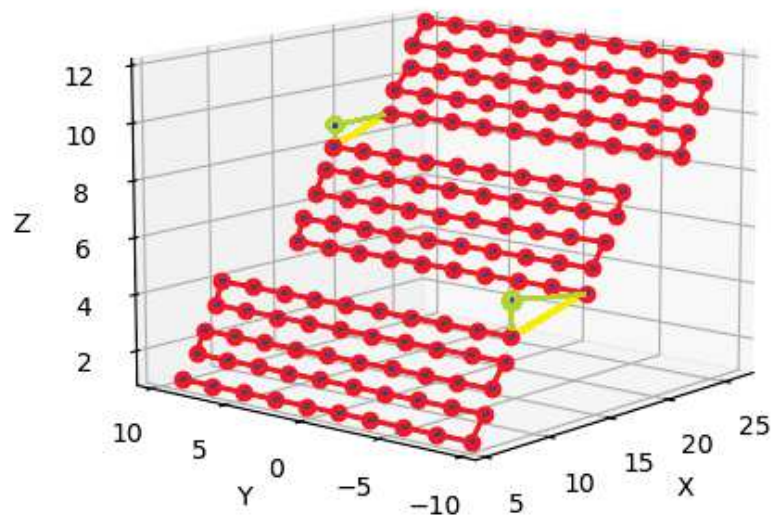
Figura 57 – Missão de teste de tempo máximo igual a 10 minutos.



Fonte: Elaborado pelo Autor.



Figura 58 – Missão de teste de tempo máximo igual a 10 minutos com correções de segurança.



Fonte: Elaborado pelo Autor.

O próximo teste é a variação dos ganhos da relação entre os objetivos: (i) tempo e (ii) fotometria. Observando os três objetivos, temos as Tabelas ANOVA 19, 20 e 21 e os boxplot nas Figuras 59, 60 e 61 respectivamente para o tempo de missão, cobertura horizontal e vertical. Todos os fatores  $p$  são bem menores que o limiar de 0,05, mostrando que em relação aos ganhos os dados são distintos. Na Figura 59, percebe-se que quando a relação  $\frac{G_{Tempo}}{G_{Fotometria}}$  tende para o objetivo de tempo, como nos valores 10 e 100, o tempo máximo de missão foi reduzido partindo de valores acima do tempo limite de 20 minutos quando o objetivo fotométrico está em destaque e indo para valores abaixo de 2,5 minutos quando o objetivo de tempo é extremo. Nas Figuras 60 e 61, é possível observar o efeito oposto. A cobertura horizontal inicialmente está acima de 90% na maioria dos casos, enquanto reduz para valores próximos de 40% quando o objetivo de tempo é destacado. Para a cobertura vertical, é verificada uma variação de 80% para a maioria dos valores até valores próximos de 30%. Sendo isso, a variação dos ganhos pode ser utilizada para regiões menores para aumentar a cobertura, já que o tempo das missões é menor ou aumentar a restrição de tempo para regiões grandes onde o risco do limite de

tempo é um ponto crucial.

Tabela 19 – Tabela Anova de Tempo de missão com a variação da relação entre os ganhos.

Tabela Anova					
	Df	Sum Square SS	Mean Square MS	Valor $F$	Pr(>F)
Ganhos	4	13762,6557	3440,6639	1048,8983	8,483e-153
Resíduos	245	803,6648	3,2803		

Figura 59 – Boxplot do tempo de missão com a variação da relação entre os ganhos.

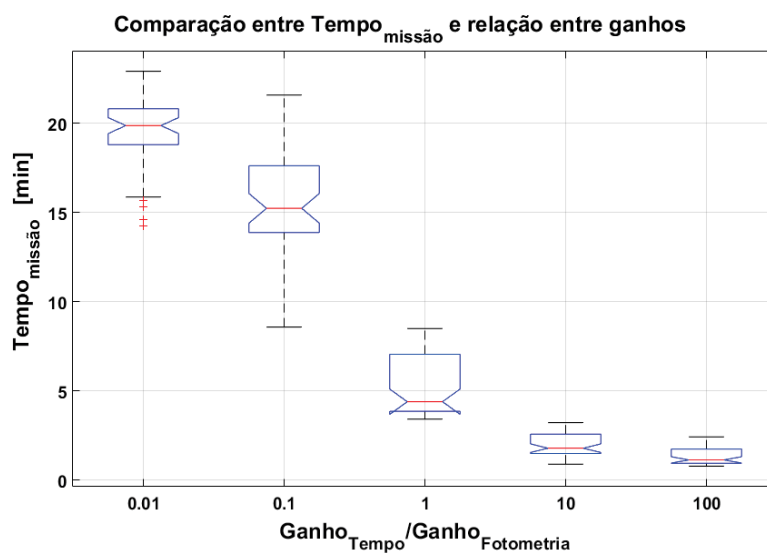


Tabela 20 – Tabela Anova de cobertura horizontal com a variação da relação entre os ganhos.

Tabela Anova					
	Df	Sum Square SS	Mean Square MS	Valor $F$	Pr(>F)
Ganhos	4	5,5866	1,3966	106,4227	2,0898e-52
Resíduos	245	3,2153	0,013124		

Figura 60 – Boxplot da cobertura horizontal com a variação da relação entre os ganhos.

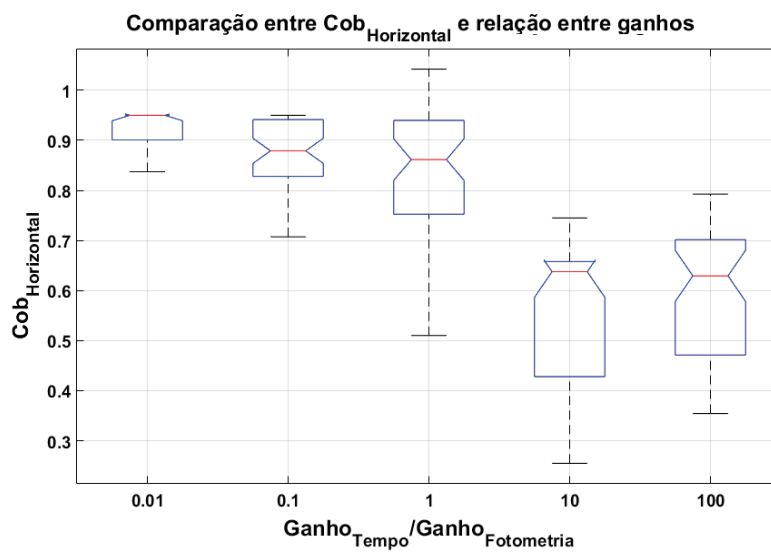
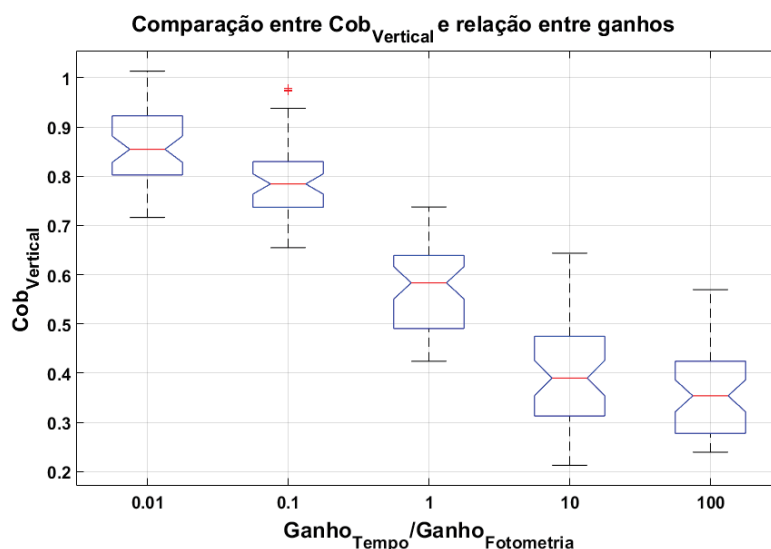


Tabela 21 – Tabela Anova de cobertura vertical com a variação da relação entre os ganhos.

Tabela Anova					
	Df	Sum Square SS	Mean Square MS	Valor $F$	Pr(>F)
Ganhos	4	10,1406	2,5352	309,7301	1,5454e-94
Resíduos	245	2,0053	0,0081851		

Figura 61 – Boxplot da cobertura vertical com a variação da relação entre os ganhos.



Fonte: Elaborado pelo Autor.

## 4.2 Testes em ambiente real

A metodologia proposta apresenta a aquisição de dados através de sensores ativos, como o *LaserScan*. Para estruturas com dimensões ainda maiores, é difícil que sensores capturem toda a informação do ambiente, sendo necessário que essa informação chegue ao *software* de pós-processamento de outra forma. Uma solução é utilizar uma reconstrução 3D previamente realizada. As reconstruções 3D previamente realizadas contêm as informações das dimensões e formatos da estrutura, podendo substituir a utilização de sensores durante a execução do voo. Dessa forma o planejamento inicial pode ser realizado em solo, de maneira *offline*, iniciando a missão com as informações da superfície já filtradas e a primeira missão já planejada. O algoritmo vai realizar os replanejamentos e as missões locais de forma *online*, ajustando aos dados disponibilizados em solo. Esse processo permite vistoria em áreas maiores, onde há necessidade de sensores mais potentes e que gastam mais bateria do VANT, além da diminuição da carga presente no drone.

As reconstruções 3D podem ser divididas em três categorias: *(i)* representações baseadas em voxel; *(ii)* representações baseadas em pontos; *(iii)* representações

Figura 62 – Estrutura Real utilizada para testes.



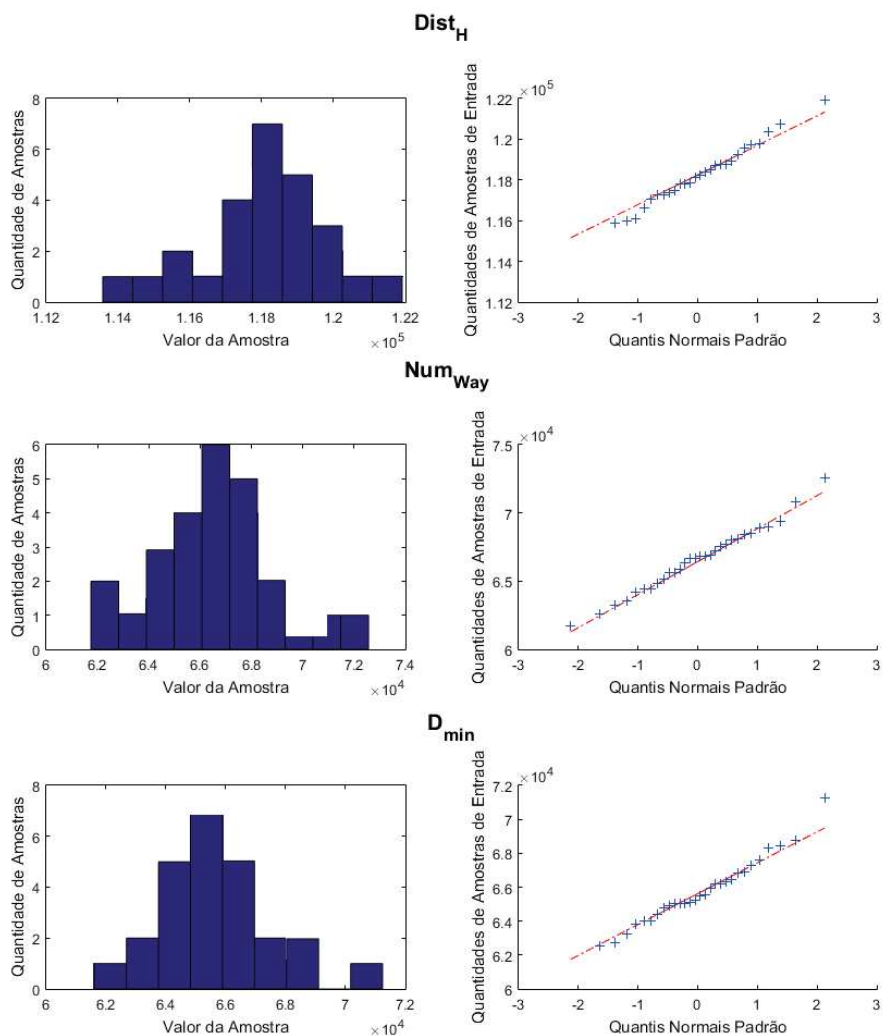
Fonte: Elaborado pelo Autor.

de malha. As representações Voxel são uma generalização direta de pixels para o caso 3D [Mescheder et al. 2019]. As reconstruções devem obedecer ao sistema de coordenadas do VANT, com destaque para ECEF e GPS. ECEF, que é um acrônimo para *earth-centered, earth-fixed*, é um sistema de coordenadas geográficas e cartesianas. Representa posições como coordenadas  $X, Y, Z$ . O ponto  $(0, 0, 0)$  é definido como o centro de massa da Terra [Rahman e Farrell 2019]. O GPS é um sistema de navegação por rádio baseado em satélite que representa a posição do globo terrestre por latitude, longitude e altura em relação a um modelo elipsoidal da Terra [Jr et al. 2019]. Dessa forma, as informações da superfície mudam do sensor ativo para a estrutura reconstruída georreferenciada.

Para comparar a metodologia apresentada, foram realizados voos otimizados usando um Phantom Pro 4 da empresa DJI [Peppas et al. 2019] em uma superfície presente no CRITT - UFJF. A Figura 62 apresenta a região objetivo do voo. Foram realizadas comparações em relação à distância ao talude em questão, distância entre os pontos, e número de missões horizontais.

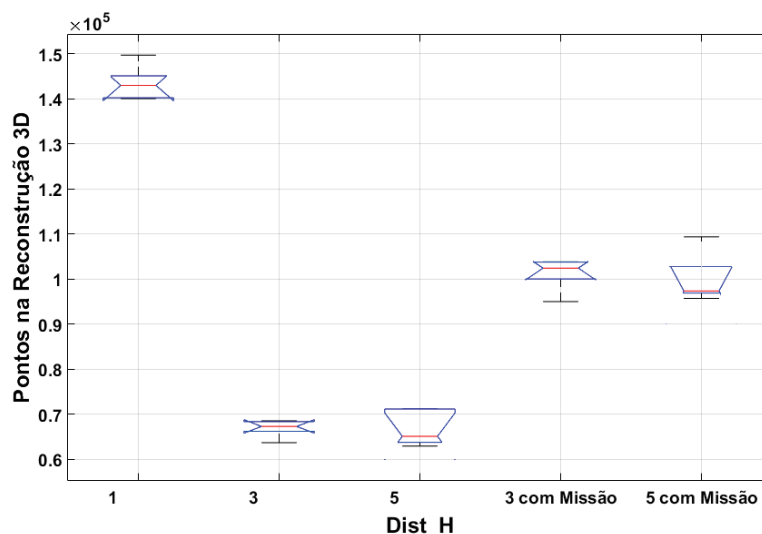
Como critério para utilizar as Tabelas ANOVA foi realizado o teste de normalidade dos resultados, mostrados na Figura 63. Todos os três dados tem composição próxima à normal, destacada tanto pelo histograma quanto pelo qq-plot. No caso poucos testes foram realizados, totalizando uma total de 25 voos para cada teste. Dessa forma uma maior amostragem aproximaria ainda mais os gráficos para normalidade. A média foram  $1.18 \cdot 10^5$ ,  $6.6 \cdot 10^4$  e  $6.6 \cdot 10^4$  *pixels* para os testes, respectivamente para  $Dist_H$ ,  $Num_{way}$  e  $D_{min}$ .

Figura 63 – Análise de normalidade das variáveis de decisão em ambiente real.



Fonte: Elaborado pelo Autor.

Os voos foram realizados com distâncias diferentes para entender o impacto da distância na reconstrução. Como exemplo, foram cinco voos para cada distância. A Tabela ANOVA 22 e a Figura 64 mostra os resultados para missões com e sem a presença de missões locais. O fator-p é menor que 0,05, mostrando que os dados são distintos para a variação da distância a superfície. Percebe-se que voos sem missões locais apresentam mais pontos na reconstrução 3D quanto mais próximos da superfície. O problema dessas missões com distâncias menores é que podem

Figura 64 – Boxplot da variação de  $Dist_H$ .

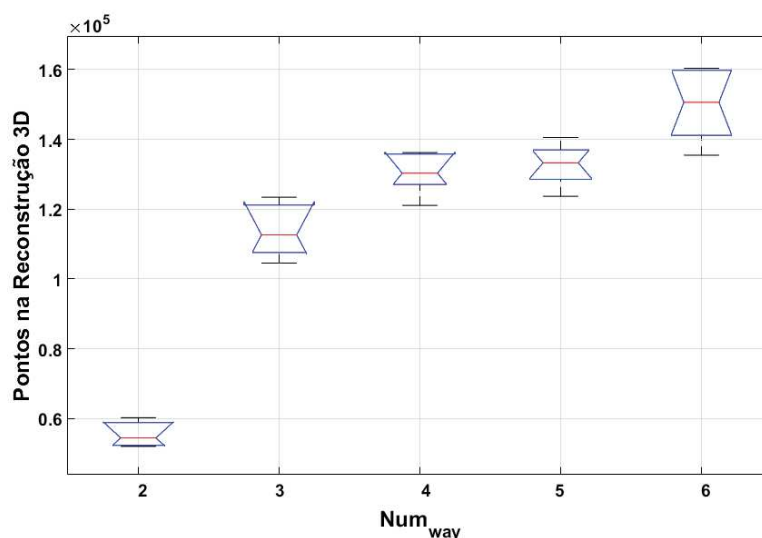
Fonte: Elaborado pelo Autor.

causar problemas de segurança com erros de GPS, além de aumentar o tempo da missão. Desta forma, foi proposto o uso de missões locais para aumentar a densidade em alguns pontos específicos. A vegetação na superfície foi escolhida como ponto de interesse, como mostrado na Figura 15. Na reconstrução com 3 metros, o aumento da densidade foi em torno de 51 %, e na de 5 metros foi de 54%. O aumento no tempo para realizar as missões foi de 10% a 20% do tempo gasto.

Tabela 22 – Tabela Anova de pontos na reconstrução 3D com a variação da distância ao talude.

Tabela Anova					
	Df	Sum Square SS	Mean Square MS	Valor $F$	Pr(>F)
$Dist_H$	4	1,99e10	4,98e9	286,0486	2,42e-17
Resíduos	20	3,48e8	1,74e7		

O mesmo estudo foi realizado para o número de *waypoints* verticais para missões com distância de 2 metros da superfície e de 5 metros de altura, com abertura de câmera de 60°. Tabela ANOVA 23 e Figura 65 mostram os resultados obtidos. O fator-p é menor que 0,05. Observe que o maior número de *waypoints*

Figura 65 – Boxplot da variação de  $Num_{way}$ .

Fonte: Elaborado pelo Autor.

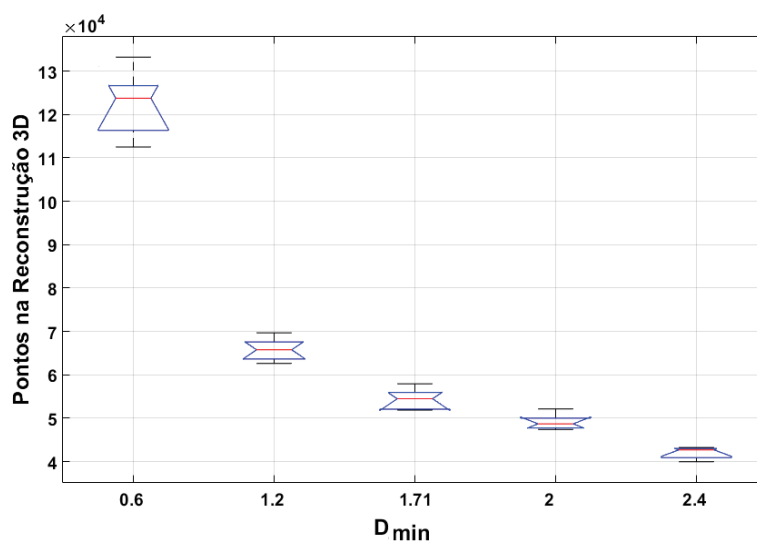
aumenta a densidade de pontos na reconstrução 3D, justificando o aumento na cobertura vertical sendo um fator da função de aptidão. No entanto, o aumento no número de  $Num_{way}$  aumenta drasticamente o tempo da missão.

Tabela 23 – Tabela Anova de pontos na reconstrução 3D com a variação do número de translados horizontais.

Tabela Anova					
	Df	Sum Square SS	Mean Square MS	Valor $F$	Pr(>F)
$Num_{way}$	4	2,64e10	6,61e9	123,4974	8,44e-14
Resíduos	20	1,07e9	5,36e7		

Os resultados em relação a  $D_{min}$  são apresentados na Tabela ANOVA 24 e na Figura 66. Os resultados foram feitos com  $Dist_H = 2$  m,  $Num\_Way = 6$  e abertura horizontal da câmera de  $100^\circ$ . O valor está fortemente relacionado ao número de pontos na reconstrução 3D. Observe que quanto maior a cobertura horizontal, maior é a densidade de pontos. Suponha que a câmera do VANT possa capturar a imagem sem parar. Dessa forma, diminuir  $D_{min}$  é uma alternativa para aumentar a captura de pontos, impactando de menor forma o tempo de missão.



Figura 66 – Boxplot da variação de  $D_{\min}$ .

Fonte: Elaborado pelo Autor.

No nosso caso, como existe a parada para captura da imagem, a relação entre o aumento de pontos gera o aumento do tempo total da missão.

Tabela 24 – Tabela Anova de pontos na reconstrução 3D com a variação da distância entre os pontos.

Tabela Anova					
	Df	Sum Square SS	Mean Square MS	Valor $F$	Pr(>F)
$D_{\min}$	4	2,08e10	5,20e9	325,462	6,82e-18
Resíduos	20	3,19e8	1,594e7		

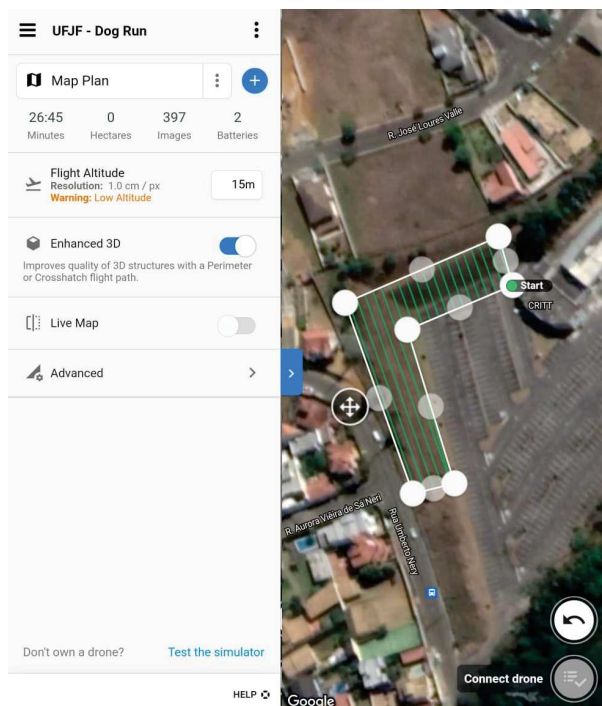
Com os resultados apresentados nas Figuras 64, 65 e 66, é possível notar que para a produção de uma missão adequada o VANT deve se executar mais transferências horizontais, o mais próximo possível da superfície. Além de pontos de captura de imagem próximos entre si. No entanto, este objetivo fotométrico para reconstrução 3D aumenta o tempo da missão, pois o VANT não cobrirá toda a superfície do alvo em uma missão. Desta forma, algoritmos meta-heurísticos equilibram esses objetivos, possibilitando uma reconstrução 3D com qualidade e adaptando todo o tempo de missão da superfície. Outro detalhe é a possibilidade

de cumprir o objetivo de distância à superfície em alguns locais críticos da missão, aumentando a qualidade da reconstrução 3D em pontos de interesse para o operador. Desta forma, a missão terá uma distância maior na superfície em geral e, em pequenas partes, terá uma aproximação com a superfície.

### 4.3 Comparação entre Algoritmos

Para comparação dos resultados encontrados, escolheram-se algoritmos de CPP disponíveis na literatura. O primeiro algoritmo é o *DroneDeploy* [DroneDeploy 2021]. *DroneDeploy* é a plataforma de *software* para VANTs comerciais. Com a confiança de usuários em uma variedade de setores, transformando a forma como as empresas coletam, gerenciam e interpretam dados coletados. Com uma interface amigável, é possível para empresas de qualquer tamanho melhorar seus fluxos de trabalho por meio de processamento de imagem escalável, armazenamento de dados e mapas de VANTs compartilháveis em tempo real e modelos 3D. Possui a geração de missões para captura de imagens de reconstrução 3D com controle das interseções entre as imagens, altura de voo e velocidade do VANT. Figura 67 mostra uma missão criada na região do CRITT-UFJF para captura de imagens. No *software*, pode-se configurar manualmente a altura de voo (*Flight Altitude*), altura em relação ao ponto de partida no qual o VANT realizará a missão de varredura, destacada em verde, para capturar imagens. A interseção padrão entregue pelo *software* é de 80% entre as imagens, tanto vertical quanto horizontalmente e velocidade fixa de 1 m/s, podendo ser modificada manualmente. Dessa forma com a altura de 15 metros, como indicado na Figura 67, o *software* criou uma missão de 26 minutos e 45 segundos onde irá capturar 397 imagens. Como o tempo de bateria fixo para 16 minutos, ele divide em duas missões.

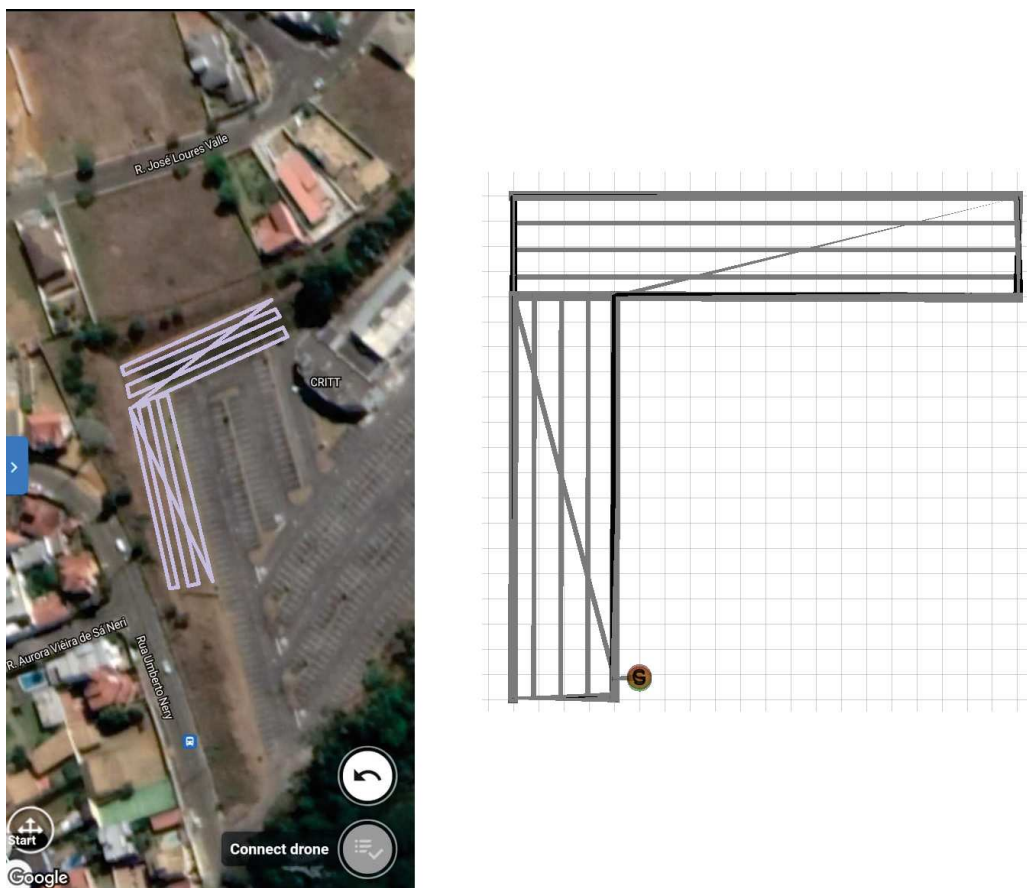
Figura 67 – Exemplo de missão criada utilizando *DroneDeploy*.



Fonte: Elaborado pelo autor.

Outro algoritmo para comparação é o *polygon coverage planning* (PCP) [Bähnemann et al. 2021], algoritmo que utiliza de decomposição em polígonos para planejamento de voos em baixas altitudes com a presença de obstáculos, atualizando o planejamento da cobertura do boustrophedon e otimizando as diferentes combinações de varredura para encontrar o caminho de varredura ideal e considera os obstáculos durante os voos de transição entre as células. O algoritmo cria um planejamento 2D utilizando obstáculos como áreas de impossibilidade de voo (*no-fly-zones*). Figura 68 mostra a missão gerada para a mesma região do CRITT-UFJF pelo algoritmo. Percebe-se que o mesmo segmentou a região em duas partes, sendo a primeira à esquerda do mapa e a segunda acima. Além disso, realizou o caminho em cada uma delas, conectando no final as duas partes. A direita está a saída do algoritmo, enquanto a esquerda é a projeção no mapa.

Figura 68 – Exemplo de missão criada utilizando *polygon coverage planning*.

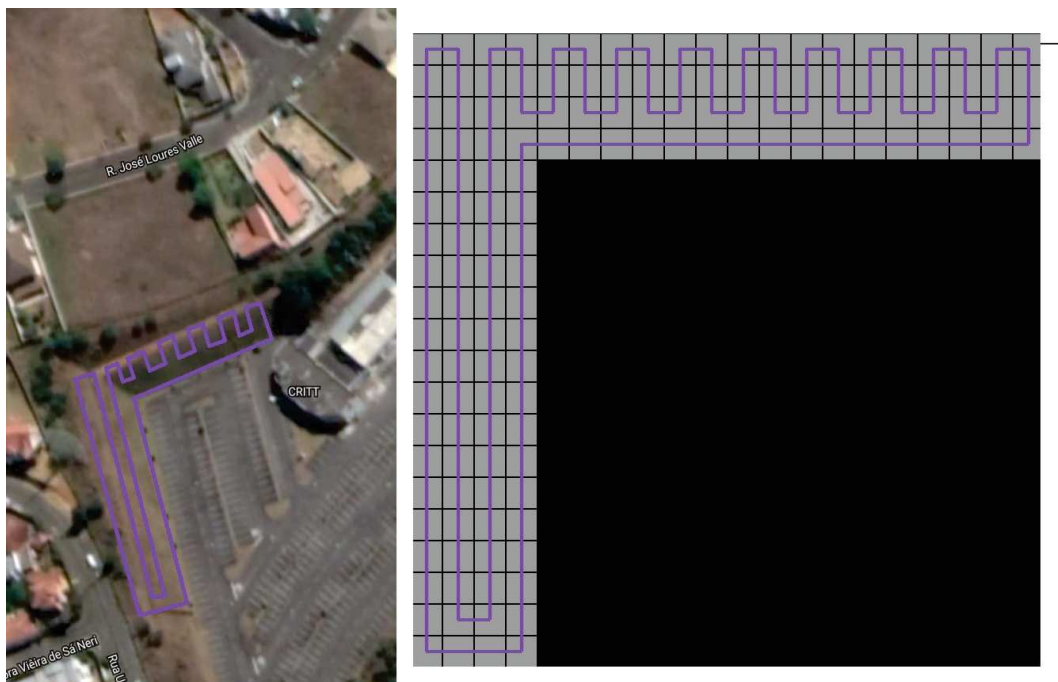


Fonte: Elaborado pelo autor.

O último algoritmo é o DARP (*Divide areas algorithm for optimal multi-robot coverage path planning*) [Kapoutsis e Chatzichristofis 2017] que é um planejador de caminhos para múltiplos robôs que teve seus parâmetros de entrada modificado para somente uma aeronave. O algoritmo transforma o problema de múltiplos robôs em diversos problemas com somente um equipamento para encontrar a solução ótima, onde todos os robôs deslocarão à mesma distância. O algoritmo utiliza *Spanning Tree Coverage* que subdivide a região de trabalho em células disjuntas correspondentes, então segue uma árvore de abrangência do gráfico induzida pelas células, enquanto cobre cada ponto precisamente uma vez. A Figura 69 mostra a saída do algoritmo (a direita) e a sua projeção no mapa da região do CRITT (a

esquerda). O algoritmo tem como entrada o tamanho de cada célula e a posição inicial dos VANTs. Dessa forma, foi gerado um mapa de 20x20 para que no centro de cada célula ocorra a captura das imagens.

Figura 69 – Exemplo de missão criada utilizando DARP.



Fonte: Elaborado pelo autor.

Todos os três algoritmos tem como entrada a altura que será realizada a navegação, pois são feitas em planos bidimensionais com altura fixa. A comparação é com o método proposto que tem altura variada sempre equidistante a superfície. Para a comparação foram realizados voos com cada um dos algoritmos. Os voos foram às 8 horas dos dias 20, 21 e 22 de outubro de 2021. Para o algoritmo *DroneDeploy* temos dois voos: o primeiro, do dia 20, foi com as configurações padrões do algoritmo (80% de ambas as interseções e 1 m/s de velocidade do VANT), porém com altura de 30 metros em relação ao solo. Esse voo teve duração de 15 minutos e 58 segundos e 1 bateria com a captura de 124 imagens. Já o segundo voo, do dia 21, foi realizado com altura de 15 metros, porém com as interseções em 60% e a velocidade de 1 m/s e com duração de 15 minutos e 42

segundos e a captura de 122 imagens.

Para os algoritmos PCP e DARP teve-se que utilizar um software de suporte para a navegação com a passagem dos pontos de latitude, longitude e altura. O *software* Litchi [Ltd 2021] foi escolhido pela sua capacidade de controle do VANT e a captura de imagens em pontos específicos. O VANT é o DJI Phantom Pro com resolução das imagens em FULL HD ( $1920 \times 1080 \text{ pixels}$ ). Ambos os algoritmos não escolhem o tempo de voo, então

Abaixo, é possível verificar os valores da saída do algoritmo proposto para essa área. A missão gerada tem tempo de execução de 8 minutos e 14 segundos e cobertura horizontal de 74% e vertical de 63% sem a identificação dinâmica de objetivos, que os outros algoritmos não têm disponível. O voo aconteceu no dia 22 de Outubro de 2021. Os resultados foram resumidos na Tabela 25.

- $D_{\min} = 2.28$  metros
- $Dist_H = 12.4$  metros
- $Num_{\text{way}} = 3$

Tabela 25 – Resultados comparativos entre metodologias em mundo real.

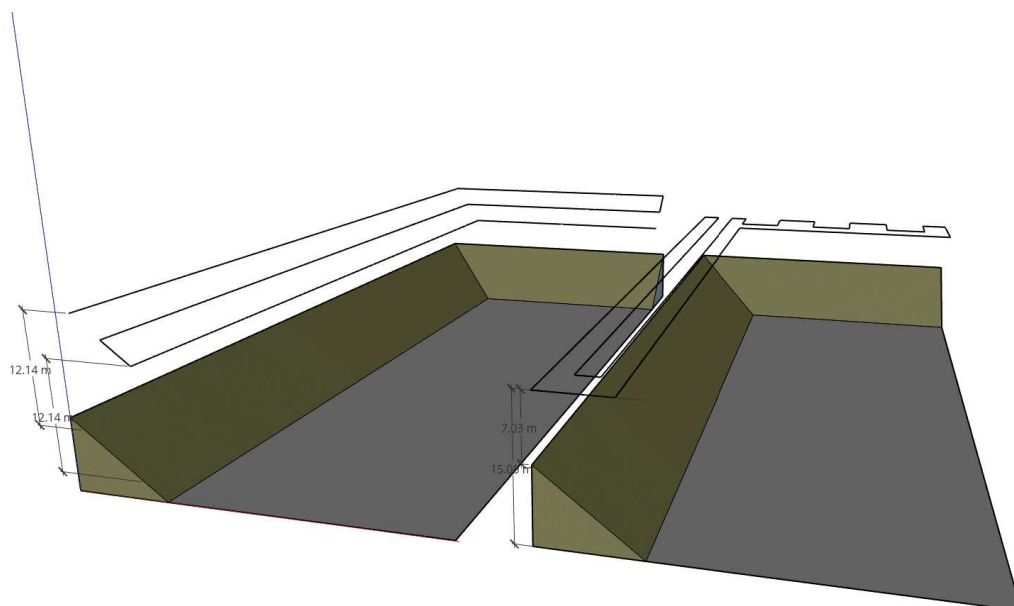
Método	Tempo de Voo [Min:Seg]	Int. Hor. [%]	In. Vert. [%]	Altura do Voo [m]	Imagens Capt.	Quantidade de Pontos na Recons. 3D
DroneDeploy	15:42	60	60	15	122	104212
DroneDeploy	15:58	80	80	30	124	98320
DARP	15:20	65,4	65,4	15	144	112403
DARP	15:40	85,6	85,6	30	144	104882
PCP	18:15	73,8	73,8	15	170	115384
PCP	18:35	90,1	90,1	30	170	108310
Metodologia Proposta	12:14	74,0	63,3	12,4	112	110215

A primeira consideração é que todos os algoritmos produziram resultados utilizando somente uma bateria, sendo que as alturas e interseções foram ajustadas para atender o máximo de 20 minutos de missão. Percebe-se que em relação ao

tempo de missão a metodologia proposta teve o melhor resultado, já que a mesma tem como objetivo a redução do tempo da missão. Os algoritmos DARP e PCP mantiveram as mesmas missões adicionando somente o tempo para chegar na altura desejada e retornar ao solo. Já o *DroneDeploy*, modificou a missão alterando a interseção e os pontos de captura de imagem.

Sobre o aumento da altura de voo, percebe-se que diminuiu a qualidade da reconstrução 3D, justificando a inserção do ganho da Equação 3.20 na metodologia. Mesmo o aumento da interseção não compensou o aumento da distância a superfície. As missões do *DroneDeploy*, DARP e PCC tem o cálculo das interseções em relação ao solo, porém como a superfície não tem altura fixa temos uma variação dessa interseção ao decorrer da execução da missão, como mostrado na Figura 70, onde a altura varia de 15 metros no solo e 7 metros no ponto mais alto. Sobre o ângulo da câmera e a superfície, a câmera dos outros métodos sempre são voltadas para baixo mesmo nos pontos inclinados da missão. Já a metodologia proposta ajusta altura para permanecer constante em 12,04 metros em toda a missão, sendo a interseção entre as imagens sempre fixa e o ângulo da câmera sempre seguindo a normal da superfície. O número de imagens capturadas também é menor reduzindo a redundância e, em equipamentos onde o tempo de parada para captura de imagens seja maior, a missão também terá seu tempo reduzido.

Figura 70 – Comparação entre missão proposta e dos outros algoritmos.



Fonte: Elaborado pelo autor.

Apesar de não ter a melhor quantidade de pontos na reconstrução 3D, a metodologia proposta conseguiu alcançar bons resultados visando seus dois objetivos. Outro fator que pode melhorar a qualidade da reconstrução é o relaxamento do Ganho de tempo ( $G_{Tempo}$  - Equação 3.19) já que o tempo disponível ainda é alto em relação ao tempo máximo da missão.

#### 4.4 Conclusões Parciais

Os resultados apresentados nesta seção mostraram que a metodologia atingiu o objetivo de vistoriar uma área com requisitos de segurança e fotométricos. O algoritmo seguiu os critérios para um CPP definidos por Cao et al. [Cao e Huang 1988], além de adicionar as mini-missões, que aumentaram a qualidade das reconstruções realizadas.

Os resultados simulados tiveram com objetivo explorar a capacidade do algoritmo antes dos testes reais. A variedade das meta-heurísticas e das populações criaram milhares de missões que foram analisadas usando o método ANOVA para



descobrir a população de indivíduos ideal e logo após a meta-heurística que teve melhor resultado para o problema proposto. Além disso, diversas estruturas de superfícies foram analisadas com entrada pelos sensores *Laser* e por DEM. Dessa forma foram selecionados 300 indivíduos para o algoritmo de morcegos.

Após a decisão relacionada a meta-heurística foi mostrada a realização da identificação de pontos de interesse e seu impacto nas missões dinâmicas. Como resultado percebeu que a variação entre o tempo planejado de execução e o tempo de fato realizado poderia acarretar problemas de segurança da missão. Neste caso o replanejamento constante diminui a possibilidade desse problema, além de permitir que, mesmo com uma perda grande de tempo por motivo não planejado, a missão seja executada até o final otimizando o tempo disponível com a superfície a ser vistoriada.

Sobre as modificações em relação à função de *fitness* observou que o algoritmo proposto consegue modificar a missão em relação ao tempo máximo, sendo um aumento da restrição de tempo. Sobre a relação dos ganhos temos como resultado que ao se adicionar um ganho favorável a fotometria as missões ficaram muito próximas ao tempo máximo, até mesmo passando deste limite. Essa escolha pode ocasionar quedas por falta de bateria no final da missão. Já o objetivo de tempo como objetivo maximizado teve-se uma queda nas coberturas, o que ocasiona perda de qualidade e disponibilidade de tempo extra não utilizado no VANT. Esses dois objetivos opostos devem ser equilibrados de acordo com a proposta do operador.

Os resultados em ambiente real mostraram que o objetivo de maximizar a cobertura aumentam a qualidade da reconstrução 3D, apesar do aumento de tempo. Isso se mostrou com a variação das variáveis de decisão em um mesmo ambiente. Já sobre a comparação com os outros algoritmos, a metodologia proposta conseguiu competir com algoritmos bem conceituados de mercado tendo a vantagem de manter coberturas constantes em toda a missão em contraponto das outras metodologias de altura fixa. Um fator contra a metodologia é a necessidade de informações do ambiente mais detalhadas que as outras metodologias, que só precisam da área a ser vistoriada, não levando em conta diferentes características de relevo.

## 5 Conclusões e Propostas Futuras

A metodologia apresentada nesta tese de doutorado tem como principal objetivo maximizar a qualidade da reconstrução 3D e otimizar o tempo total de voo do VANT em problemas de CPP. Além disso, o algoritmo desenvolvido é capaz de perceber pontos de interesse durante o voo, modificando a missão enquanto mantém os critérios de CPP, e otimiza novamente a missão para adequar às variações que não foram previstas.

Os objetivos propostos por essa tese foram alcançados com sucesso, tendo a comparação com as mais diversas meta-heurísticas, além de outros algoritmos presentes na literatura. Aplicou-se a metodologia em diversos tipos de superfícies, tendo encontrado missões capazes de atender a qualidade das fotos e a segurança de uma missão sem colisão com a superfície por manobras ou quedas por falta de bateria.

A proposta de aumentar a qualidade das reconstruções 3D adicionando objetivos fotométricos na otimização da missão é um desafio promissor, aumentando a qualidade dos dados disponíveis na nuvem de pontos, permitindo a utilizando desse ambiente para estudos mais complexos, tais como comparação entre reconstruções e análises de tensões na superfície. Outros critérios podem ser estudados e adicionados a construção da função de aptidão, afetando as missões.

Neste caso a contribuição entregue por essa tese é uma metodologia de otimização para planejamento de caminhos por cobertura equidistantes a superfícies que otimiza dois objetivos conflitantes: tempo e fotometria para aumentar a qualidade de reconstrução 3D junto a critérios de segurança da missão.

Sobre as meta-heurísticas investigadas, apesar da boa performance do algoritmo de morcegos, é necessário investigar a melhor escolha de cada hiper-parâmetro envolvido. Todas as meta-heurísticas propostas tiveram resultados que atenderam ao aumento da qualidade nas imagens e ao tempo de missão, podendo ser aplicado nessa metodologia. Sobre o tamanho da população, teve-se uma investigação sobre o tamanho ideal para o algoritmo de forma *online*, porém em casos utilizando *point cloud* e DEM pode-se realizar uma otimização com população maior com o VANT em solo e utilizar esse conhecimento prévio para populações

menores durante o voo. Assim permitindo melhores resultados *offline* sem afetar o tempo de processamento durante a execução da missão.

Na pesquisa deste tema algumas dificuldades foram encontradas. O primeiro desafio é na escolha de toda a estrutura presente em simuladores para a escolha dos VANT's e a necessidade de equipamentos reais que tenham sua estrutura de navegação e controle próximos ao equipamentos reais para possibilitar testes confiáveis em ambiente simulado que não geram prejuízos em ambiente real.

O segundo desafio é ligado à implementação dos algoritmos no VANT. Os equipamentos escolhidos têm sua arquitetura fechada pelo fabricante, dificultando a aplicação dos algoritmos no VANT. A escolha de algoritmos de terceiros como o Litchi [Ltd 2021] possibilitou a interação com o programa desenvolvido e o equipamento, mas para a criação de um software completo é necessário o desenvolvimento a partir de SDK (*Software Development Kit*) disponíveis pela empresa. Cada missão era testada pelo menos 5 vezes nos simuladores antes dos testes reais, o que gerou mais segurança para as missões, nas quais não ocorreu nenhuma colisão ou queda do equipamento.

Como proposta para continuação da pesquisa, é esperada a modificação do algoritmo para introdução de outros algoritmos de CPP para otimizar os caminhos, sendo que a metodologia otimizaria o conjunto de *waypoints* visitados e a conexão entre eles otimizada por outros algoritmos, tais como *nearest neighbour*, *tabu search*, *variable neighborhood search* e *greedy randomized adaptive search procedure* e *iterated local search*. Outra proposta é a utilização de grafos no processo de conexão entre os *waypoints*. Essa proposta aumenta a otimização dos caminhos propostos entre os pontos e a otimização fotométrica de captura de imagens, podendo garantir um caminho ainda mais otimizado, e podendo adicionar outros critérios tais como distância a obstáculos, capacidade de manobra e até mesmo múltiplos VANTs. Esse processo pode ser investigado tanto de forma *online* com a identificação de objetivos dinâmico e a reprogramação da missão, quanto de forma *offline* onde somente uma missão é gerada com informações através de DEM ou *point cloud*.

Outra proposta é a captura de informações do ambiente. No caso temos a utilização de sensores *Laser* bidimensionais, DEM e *point cloud*. Outra questão é a adição de *Laser* 3D e de câmeras estéreo para a captura do formato do objeto a ser

inspecionado, analisando o impacto da aplicação dessas duas técnicas no algoritmo. Além disso, a introdução de mais pontos de interesse situacional, como rachaduras, canos e outros aspectos de inspeção que podem ser do interesse do operador.

Os algoritmos de otimização escolhidos são meta-heurísticas. Como outra proposta futura é a possibilidade de introdução de outros métodos como redes neurais, sistemas evolutivos ou não paramétricos, permitindo a comparação com as meta-heurísticas neste tipo de problema, e analisando o impacto em relação ao processo de otimização e as variáveis de decisão.

Como finalização para a tese irá ser adicionado ao github do grupo de robótica da UFJF o algoritmo em formato de *client* e *server* do ROS da metodologia proposta. Além disso o seguinte vídeo mostra a utilização do algoritmo em simulador: <https://youtu.be/mJ6M-Pt-BC0>. No github estará disponível os dados utilizados para futuras pesquisas.

**REFERÊNCIAS**

- [Abdechiri e Meybodi 2013]ABDECHIRI, M.; MEYBODI, M. R. Gases brownian motion optimization: an algorithm for optimization (gbmo). *Applied Soft Computing*, Elsevier, v. 13, n. 5, p. 2932–2946, 2013.
- [Abedinpourshotorban et al. 2016]ABEDINPOURSHOTORBAN, H. et al. Electromagnetic field optimization: a physics-inspired metaheuristic optimization algorithm. *Swarm and Evolutionary Computation*, Elsevier, v. 26, p. 8–22, 2016.
- [Acar et al. 2002]ACAR, E. U. et al. Morse decompositions for coverage tasks. *The international journal of robotics research*, SAGE Publications Sage UK: London, England, v. 21, n. 4, p. 331–344, 2002.
- [Aggarwal e Kumar 2020]AGGARWAL, S.; KUMAR, N. Path planning techniques for unmanned aerial vehicles: A review, solutions, and challenges. *Computer Communications*, Elsevier, v. 149, p. 270–299, 2020.
- [Arkin e Fekete 2000]ARKIN, E. M.; FEKETE, S. P. Approximation algorithms for lawn mowing and milling. *Computational Geometry*, Elsevier, v. 17, n. 1-2, p. 25–50, 2000.
- [Atkar et al. 2001]ATKAR, P. N. et al. Exact cellular decomposition of closed orientable surfaces embedded in  $\mathbb{R}^3$ . In: IEEE. *Proceedings 2001 ICRA. IEEE International Conference on Robotics and Automation (Cat. No. 01CH37164)*. [S.l.], 2001. v. 1, p. 699–704.
- [Azpúrua et al. 2018]AZPÚRUA, H. et al. Multi-robot coverage path planning using hexagonal segmentation for geophysical surveys. *Robotica*, Cambridge University Press, v. 36, n. 8, p. 1144–1166, 2018.
- [Bähнемann et al. 2021]BÄHNEMANN, R. et al. Revisiting boustrophedon coverage path planning as a generalized traveling salesman problem. In: SPRINGER. *Field and Service Robotics*. [S.l.], 2021. p. 277–290.
- [Baras e Dasygenis 2019]BARAS, N.; DASYGENIS, M. Multi-robot coverage path planning in 3-dimensional environments. In: IEEE. *2019 8th International Con-*

- ference on Modern Circuits and Systems Technologies (MOCAST)*. [S.l.], 2019. p. 1–4.
- [Basset e Abdel 2018]BASSET, M. A.; ABDEL, L. F. Metaheuristic algorithms: A comprehensive review. *Computational intelligence for multimedia big data on the cloud with engineering applications*, Elsevier, p. 185–231, 2018.
- [Bean 1960]BEAN, W. B. The clonal selection theory of acquired immunity. *AMA Archives of Internal Medicine*, American Medical Association, v. 105, n. 6, p. 973–974, 1960.
- [Benjamini 1988]BENJAMINI, Y. Opening the box of a boxplot. *The American Statistician*, Taylor & Francis, v. 42, n. 4, p. 257–262, 1988.
- [Besada et al. 2018]BESADA, J. A. et al. Drone mission definition and implementation for automated infrastructure inspection using airborne sensors. *Sensors*, Multidisciplinary Digital Publishing Institute, v. 18, n. 4, p. 1170, 2018.
- [Biundini et al. 2019]BIUNDINI, I. Z. et al. Coverage path planning optimization for slopes and dams inspection. In: SPRINGER. *Iberian Robotics conference*. [S.l.], 2019. p. 513–523.
- [Biundini et al. 2020]BIUNDINI, I. Z. et al. Coverage path planning optimization for slopes and dams inspection. In: SILVA, M. F. et al. (Ed.). *Robot 2019: Fourth Iberian Robotics Conference*. Cham: Springer International Publishing, 2020. p. 513–523.
- [Blum e Roli 2003]BLUM, C.; ROLI, A. Metaheuristics in combinatorial optimization: Overview and conceptual comparison. *ACM computing surveys (CSUR)*, Acm New York, NY, USA, v. 35, n. 3, p. 268–308, 2003.
- [Bo et al. 2015]BO, X. et al. College of engineering, china agricultural university; beijing key laboratory of intelligent equipment technology for agriculture; beijing research center of intelligent equipment for agriculture;; route planning algorithm and verification based on uav operation path angle in irregular area [j]. , v. 11, 2015.

- [Buehler e Iagnemma 2009]BUEHLER, M.; IAGNEMMA, K. *The DARPA urban challenge: autonomous vehicles in city traffic*. [S.l.]: springer, 2009. v. 56.
- [Buffi et al. 2017]BUFFI, G. et al. Survey of the ridracoli dam: Uav-based photogrammetry and traditional topographic techniques in the inspection of vertical structures. *Geomatics, Natural Hazards and Risk*, Taylor & Francis, v. 8, n. 2, p. 1562–1579, 2017.
- [Butler e Rizzi 1999]BUTLER, Z. J.; RIZZI, A. A. Contact sensor-based coverage of rectilinear environments. In: IEEE. *Proceedings of the 1999 IEEE International Symposium on Intelligent Control Intelligent Systems and Semiotics (Cat. No. 99CH37014)*. [S.l.], 1999. p. 266–271.
- [Cao e Huang 1988]CAO, Z. L.; HUANG, Y. Region filling operations with random obstacle avoidance for mobile robots. *Journal of Robotic systems*, Wiley Online Library, v. 5, n. 2, p. 87–102, 1988.
- [Chien e Wang 2010]CHIEN, S.-Y.; WANG, H. Human vs. algorithmic path planning for search and rescue by robot teams. In: SAGE PUBLICATIONS SAGE CA: LOS ANGELES, CA. *Proceedings of the Human Factors and Ergonomics Society Annual Meeting*. [S.l.], 2010. v. 54, n. 4, p. 379–383.
- [Choi e Lee 2021]CHOI, D.; LEE, E. M. Online 3d coverage path planning using surface vector. In: IEEE. *2021 18th International Conference on Ubiquitous Robots (UR)*. [S.l.], 2021. p. 392–396.
- [Choset 2001]CHOSSET, H. Coverage for robotics—a survey of recent results. *Annals of mathematics and artificial intelligence*, Springer, v. 31, n. 1-4, p. 113–126, 2001.
- [Choset et al. 2000]CHOSSET, H. et al. Exact cellular decompositions in terms of critical points of morse functions. In: IEEE. *Proceedings 2000 ICRA. Millennium Conference. IEEE International Conference on Robotics and Automation. Symposia Proceedings (Cat. No. 00CH37065)*. [S.l.], 2000. v. 3, p. 2270–2277.
- [Choset e Pignon 1998]CHOSSET, H.; PIGNON, P. Coverage path planning: The boustrophedon cellular decomposition. In: SPRINGER. *Field and service robotics*. [S.l.], 1998. p. 203–209.

- [Choset et al. 2005]CHOSSET, H. M. et al. *Principles of robot motion: theory, algorithms, and implementation*. [S.l.]: MIT press, 2005.
- [Chu e Tsai 2006]CHU, S.-C.; TSAI, P.-W. Cat swarm optimization. In: SPRINGER. *Pacific Rim international conference on artificial intelligence*. [S.l.], 2006. p. 854–858.
- [Dai e Feng 2014]DAI, F.; FENG, Y. Photogrammetric error sources and impacts on modeling and surveying in construction engineering applications. *Visualization in Engineering*, Springer, v. 2, n. 1, p. 2, 2014.
- [Dai et al. 2019]DAI, X. et al. Mobile robot path planning based on ant colony algorithm with a\* heuristic method. *Frontiers in neurorobotics*, Frontiers, v. 13, p. 15, 2019.
- [Dorigo e Birattari 2006]DORIGO, M.; BIRATTARI, M. Ant colony optimization. *IEEE computational intelligence magazine*, IEEE, v. 1, n. 4, p. 28–39, 2006.
- [DroneDeploy 2021]DRONEDEPLOY. *DroneDeploy*. 2021. Disponível em: <dronedeploy.com>.
- [Ellefsen e Lepikson 2017]ELLEFSEN, K. O.; LEPIKSON, H. A. Multiobjective coverage path planning: Enabling automated inspection of complex, real-world structures. *Applied Soft Computing*, Elsevier, v. 61, p. 264–282, 2017.
- [Fogel 1998]FOGEL, D. B. *Artificial intelligence through simulated evolution*. [S.l.]: Wiley-IEEE Press, 1998.
- [Franco e Buttazzo 2015]FRANCO, C. D.; BUTTAZZO, G. Energy-aware coverage path planning of uavs. In: *2015 IEEE International Conference on Autonomous Robot Systems and Competitions*. [S.l.]: IEEE, 2015. p. 111–117.
- [Franco 2016]FRANCO, G. *Como testar se uma distribuição é Normal - SOS Estatística*. 2016. <https://sosestatistica.com.br/como-testar-se-uma-distribuicao-e-normal/>. (Accessed on 23/03/2022).



- [Freitas et al. 2019]FREITAS, C. M. d. et al. Da samarco em mariana à vale em brumadinho: desastres em barragens de mineração e saúde coletiva. *Cadernos de Saúde Pública*, SciELO Brasil, v. 35, 2019.
- [Gabriely e Rimon 2002]GABRIELY, Y.; RIMON, E. Spiral-stc: An on-line coverage algorithm of grid environments by a mobile robot. In: IEEE. *Proceedings 2002 IEEE International Conference on Robotics and Automation (Cat. No. 02CH37292)*. [S.l.], 2002. v. 1, p. 954–960.
- [Galceran e Carreras 2012]GALCERAN, E.; CARRERAS, M. Efficient seabed coverage path planning for asvs and auvs. In: IEEE. *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*. [S.l.], 2012. p. 88–93.
- [Galceran e Carreras 2013]GALCERAN, E.; CARRERAS, M. A survey on coverage path planning for robotics. *Robotics and Autonomous systems*, Elsevier, v. 61, n. 12, p. 1258–1276, 2013.
- [Gao e Ding 2020]GAO, S.; DING, Y. A frontier-based coverage path planning algorithm for robot exploration in unknown environment. In: IEEE. *2020 39th Chinese Control Conference (CCC)*. [S.l.], 2020. p. 3920–3925.
- [Geem e Kim 2001]GEEM, Z. W.; KIM, J. H. A new heuristic optimization algorithm: harmony search. *simulation*, Sage Publications Sage CA: Thousand Oaks, CA, v. 76, n. 2, p. 60–68, 2001.
- [Globo 2021]GLOBO. *As Histórias na GloboNews 7: desastres de Mariana e Brumadinho*. 2021.
- [Glover 1986]GLOVER, F. Future paths for integer programming and links to artificial intelligence. *Computers & operations research*, Elsevier, v. 13, n. 5, p. 533–549, 1986.
- [Glover e McMillan 1986]GLOVER, F.; MCMILLAN, C. The general employee scheduling problem. an integration of ms and ai. *Computers & operations research*, Elsevier, v. 13, n. 5, p. 563–573, 1986.

- [Hayat et al. 2017]HAYAT, S. et al. Multi-objective uav path planning for search and rescue. In: IEEE. *2017 IEEE International Conference on Robotics and Automation (ICRA)*. [S.l.], 2017. p. 5569–5574.
- [Hert e Tiwari 1996]HERT, S.; TIWARI, S. A terrain-covering algorithm for an auv. In: *Underwater Robots*. [S.l.]: Springer, 1996. p. 17–45.
- [Holst e Kuhlmann 2016]HOLST, C.; KUHLMANN, H. Challenges and present fields of action at laser scanner based deformation analyses. *Journal of applied geodesy*, De Gruyter, v. 10, n. 1, p. 17–25, 2016.
- [Jing et al. 2019]JING, W. et al. Coverage path planning using path primitive sampling and primitive coverage graph for visual inspection. *arXiv preprint arXiv:1908.02901*, 2019.
- [John 1992]JOHN, H. Holland. genetic algorithms. *Scientific american*, v. 267, n. 1, p. 44–50, 1992.
- [Jr et al. 2019]JR, J. W. K. et al. *System and method with automatic radius crossing notification for global positioning system (GPS) tracker*. [S.l.]: Google Patents, 2019. US Patent 10,448,196.
- [Kalinina et al. 2018]KALININA, D. et al. Computing concave hull with closed curve smoothing: performance, concaveness measure and applications. *Procedia Computer Science*, Elsevier, v. 136, p. 479–488, 2018.
- [Kapoutsis e Chatzichristofis 2017]KAPOUTSIS, A. C.; CHATZICHRISTOFIS, S. A. Darp: divide areas algorithm for optimal multi-robot coverage path planning. *Journal of Intelligent & Robotic Systems*, Springer, v. 86, n. 3-4, p. 663–680, 2017.
- [Katoch e Chauhan 2021]KATOCH, S.; CHAUHAN, S. S. A review on genetic algorithm: past, present, and future. *Multimedia Tools and Applications*, Springer, v. 80, n. 5, p. 8091–8126, 2021.
- [Khaloo et al. 2018]KHALOO, A. et al. Utilizing uav and 3d computer vision for visual inspection of a large gravity dam. *Frontiers in Built Environment*, Frontiers, v. 4, p. 31, 2018.

- [Kim 2017]KIM, T. K. Understanding one-way anova using conceptual figures. *Korean journal of anesthesiology*, Korean Society of Anesthesiologists, v. 70, n. 1, p. 22, 2017.
- [Kirkpatrick e Gelatt 1983]KIRKPATRICK, S.; GELATT, C. D. Optimization by simulated annealing. *science*, American association for the advancement of science, v. 220, n. 4598, p. 671–680, 1983.
- [Klancar et al. 2017]KLANCAR, G. et al. *Wheeled mobile robotics: from fundamentals towards autonomous systems*. [S.l.]: Butterworth-Heinemann, 2017.
- [Koenig e Hsu 2013]KOENIG, N.; HSU, J. The many faces of simulation: Use cases for a general purpose simulator. In: *Proc. of the ICRA*. [S.l.: s.n.], 2013. v. 13, p. 10–11.
- [Koenig e Likhachev 2002]KOENIG, S.; LIKHACHEV, M. Improved fast replanning for robot navigation in unknown terrain. In: *IEEE. Proceedings 2002 IEEE International Conference on Robotics and Automation (Cat. No. 02CH37292)*. [S.l.], 2002. v. 1, p. 968–975.
- [Koza 1990]KOZA, J. R. A paradigm for genetically breeding populations of computer programs to solve problems. *Computer Science Dept., Stanford Univ., Margaret Jacks Hall, Stanford, Calif*, 1990.
- [Kuyu e Vatansever 2021]KUYU, Y. Ç.; VATANSEVER, F. Advanced metaheuristic algorithms on solving multimodal functions: Experimental analyses and performance evaluations. *Archives of Computational Methods in Engineering*, Springer, p. 1–13, 2021.
- [Lakshmanan et al. 2020]LAKSHMANAN, A. K. et al. Complete coverage path planning using reinforcement learning for tetromino based cleaning and maintenance robot. *Automation in Construction*, Elsevier, v. 112, p. 103078, 2020.
- [Lam e Li 2009]LAM, A. Y.; LI, V. O. Chemical-reaction-inspired metaheuristic for optimization. *IEEE transactions on evolutionary computation*, IEEE, v. 14, n. 3, p. 381–399, 2009.

- [Latombe 1990]LATOMBE, J.-C. Robot motion planning. 1991. *Kluwer Academic Publishers*, v. 25, p. 100–107, 1990.
- [LaValle 1998]LAVALLE, S. M. Rapidly-exploring random trees: A new tool for path planning. Citeseer, 1998.
- [LaValle 2006]LAVALLE, S. M. *Planning algorithms*. [S.l.]: Cambridge university press, 2006.
- [Lee e Pippin 2008]LEE, J.; PIPPIN, C. Cost based planning with rrt in outdoor environments. In: IEEE. *2008 IEEE/RSJ International Conference on Intelligent Robots and Systems*. [S.l.], 2008. p. 684–689.
- [Ltd 2021]LTD, V. T. *Litchi for DJI Mavic/ Phantom/ Inspire/Spark*. 2021. Disponível em: <<https://flylitchi.com/>>.
- [Łukasik e Żak 2009]ŁUKASIK, S.; ŻAK, S. Firefly algorithm for continuous constrained optimization tasks. In: SPRINGER. *International conference on computational collective intelligence*. [S.l.], 2009. p. 97–106.
- [Madridano et al. 2020]MADRIDANO, Á. et al. 3d trajectory planning method for uavs swarm in building emergencies. *Sensors*, Multidisciplinary Digital Publishing Institute, v. 20, n. 3, p. 642, 2020.
- [Martino e Read 2021]MARTINO, L.; READ, J. A joint introduction to gaussian processes and relevance vector machines with connections to kalman filtering and other kernel smoothers. *Information Fusion*, Elsevier, v. 74, p. 17–38, 2021.
- [Meng et al. 2016]MENG, X.-B. et al. A new bio-inspired optimisation algorithm: Bird swarm algorithm. *Journal of Experimental & Theoretical Artificial Intelligence*, Taylor & Francis, v. 28, n. 4, p. 673–687, 2016.
- [Mescheder et al. 2019]MESCHEDER, L. et al. Occupancy networks: Learning 3d reconstruction in function space. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. [S.l.: s.n.], 2019. p. 4460–4470.

- [Metropolis et al. 1953]METROPOLIS, N. et al. Equation of state calculations by fast computing machines. *The journal of chemical physics*, American Institute of Physics, v. 21, n. 6, p. 1087–1092, 1953.
- [Meyer et al. 2012]MEYER, J. et al. Comprehensive simulation of quadrotor uavs using ros and gazebo. In: *3rd Int. Conf. on Simulation, Modeling and Programming for Autonomous Robots (SIMPAN)*. [S.l.: s.n.], 2012. p. to appear.
- [Mirjalili e Lewis 2016]MIRJALILI, S.; LEWIS, A. The whale optimization algorithm. *Advances in engineering software*, Elsevier, v. 95, p. 51–67, 2016.
- [Mohsin et al. 2020]MOHSIN, I. et al. Optimization of the polishing efficiency and torque by using taguchi method and anova in robotic polishing. *Applied Sciences*, Multidisciplinary Digital Publishing Institute, v. 10, n. 3, p. 824, 2020.
- [Mora-Gutiérrez e Ramírez-Rodríguez 2014]MORA-GUTIÉRREZ, R. A.; RAMÍREZ-RODRÍGUEZ, J. An optimization algorithm inspired by musical composition. *Artificial Intelligence Review*, Springer, v. 41, n. 3, p. 301–315, 2014.
- [Moravec e Elfes 1985]MORAVEC, H.; ELFES, A. High resolution maps from wide angle sonar. In: IEEE. *Proceedings. 1985 IEEE international conference on robotics and automation*. [S.l.], 1985. v. 2, p. 116–121.
- [Moses e Olafenwa 2018]MOSES; OLAFENWA, J. *ImageAI, an open source python library built to empower developers to build applications and systems with self-contained Computer Vision capabilities*. 2018. Disponível em: <<https://github.com/OlafenwaMoses/ImageAI>>.
- [Murray e Raj 2020]MURRAY, C. C.; RAJ, R. The multiple flying sidekicks traveling salesman problem: Parcel delivery with multiple drones. *Transportation Research Part C: Emerging Technologies*, Elsevier, v. 110, p. 368–398, 2020.
- [Nguyen et al. 2019]NGUYEN, L. K. et al. Quickhulldisk: A faster convex hull algorithm for disks. *Applied Mathematics and Computation*, Elsevier, v. 363, p. 124626, 2019.

- [Nguyen et al. 2019]NGUYEN, T. et al. Efficient time-series forecasting using neural network and opposition-based coral reefs optimization. *International Journal of Computational Intelligence Systems*, Atlantis Press, v. 12, n. 2, p. 1144–1161, 2019.
- [Oliveira 2021]OLIVEIRA, B. *Análise de variância (ANOVA): Suas Aplicações E Resultados*. Bruno Oliveira [https://operdata.com.br/wp-content/uploads/2019/07/logo\\_right-185x156.png](https://operdata.com.br/wp-content/uploads/2019/07/logo_right-185x156.png), 2021. Disponível em :<<https://operdata.com.br/blog/como-interpretar-analise-de-variancia-anova/>>.
- [Orozco-Rosas e Montiel 2019]OROZCO-ROSAS, U.; MONTIEL, O. Mobile robot path planning using membrane evolutionary artificial potential field. *Applied Soft Computing*, Elsevier, v. 77, p. 236–251, 2019.
- [Palacín et al. 2005]PALACÍN, J. et al. Measuring coverage performances of a floor cleaning mobile robot using a vision system. In: IEEE. *Proceedings of the 2005 IEEE International Conference on Robotics and Automation*. [S.l.], 2005. p. 4236–4241.
- [Pandiri e Singh 2020]PANDIRI, V.; SINGH, A. Two hybrid metaheuristic approaches for the covering salesman problem. *Neural Computing and Applications*, Springer, p. 1–21, 2020.
- [Peppia et al. 2019]PEPPA, M. et al. Photogrammetric assessment and comparison of dji phantom 4 pro and phantom 4 rtk small unmanned aircraft systems. *International Archives of the Photogrammetry, Remote Sensing & Spatial Information Sciences*, 2019.
- [Pham 2015]PHAM, D. T. A comparative study of the bees algorithm as a tool for function optimisation. *Cogent Engineering*, Taylor & Francis, v. 2, n. 1, p. 1091540, 2015.
- [Pinto et al. 2020]PINTO, M. F. et al. Arcog: An aerial robotics cognitive architecture. *Robotica*, Cambridge University Press, p. 1–20, 2020.

- [Pinto et al. 2019]PINTO, M. F. et al. A framework for analyzing fog-cloud computing cooperation applied to information processing of uavs. *Wireless Communications and Mobile Computing*, Hindawi, v. 2019, 2019.
- [Quigley et al.]QUIGLEY, M. et al. Ros: an open-source robot operating system. In: . [S.l.: s.n.].
- [Rahman e Farrell 2019]RAHMAN, F.; FARRELL, J. A. Earth-centered earth-fixed (ecef) vehicle state estimation performance. In: IEEE. *2019 IEEE Conference on Control Technology and Applications (CCTA)*. [S.l.], 2019. p. 27–32.
- [95]RAO, R. V.; SAVSANI, V. J.; VAKHARIA, D. Teaching–learning-based optimization: a novel method for constrained mechanical design optimization problems. *Computer-Aided Design*, Elsevier, v. 43, n. 3, p. 303–315, 2011.
- [Rashedi e Nezamabadi-Pour 2009]RASHEDI, E.; NEZAMABADI-POUR, H. Gsa: a gravitational search algorithm. *Information sciences*, Elsevier, v. 179, n. 13, p. 2232–2248, 2009.
- [Rechenberg 1973]RECHENBERG, I. Evolution strategy: Optimization of technical systems by means of biological evolution. *Fromman-Holzboog, Stuttgart*, v. 104, p. 15–16, 1973.
- [Rodrigues 2021]RODRIGUES, L. *Entenda, de uma vez por todas, o que é boxplot e como analisar esse gráfico!* 2021. Disponível em: <<https://www.voitto.com.br/blog/artigo/boxplot>>.
- [Song et al. 2019]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)*. [S.l.], 2019. p. 123–126.
- [Theile et al. 2020]THEILE, M. et al. Uav coverage path planning under varying power constraints using deep reinforcement learning. In: IEEE. *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. [S.l.], 2020. p. 1444–1449.

- [Thieu 2020]THIEU, N. V. *A collection of the state-of-the-art Meta-heuristics ALgorithms in PYthon: Mealpy*. Zenodo, 2020. Disponível em: <<https://doi.org/10.5281/zenodo.3711948>>.
- [Thrun 1998]THRUN, S. Learning metric-topological maps for indoor mobile robot navigation. *Artificial Intelligence*, Elsevier, v. 99, n. 1, p. 21–71, 1998.
- [Vemprala e Saripalli 2018]VEMPRALA, S.; SARIPALLI, S. Vision based collaborative path planning for micro aerial vehicles. In: IEEE. *2018 IEEE International Conference on Robotics and Automation (ICRA)*. [S.l.], 2018. p. 3889–3895.
- [Vemula e Muelling 2016]VEMULA, A.; MUELLING, K. Path planning in dynamic environments with adaptive dimensionality. In: *Ninth Annual Symposium on Combinatorial Search*. [S.l.: s.n.], 2016.
- [Wei et al. 2019]WEI, Z. et al. Nuclear reaction optimization: A novel and powerful physics-based algorithm for global optimization. *IEEE Access*, IEEE, v. 7, p. 66084–66109, 2019.
- [Wolpert e Macready 1997]WOLPERT, D. H.; MACREADY, W. G. No free lunch theorems for optimization. *IEEE transactions on evolutionary computation*, IEEE, v. 1, n. 1, p. 67–82, 1997.
- [Wong e MacDonald 2003]WONG, S. C.; MACDONALD, B. A. A topological coverage algorithm for mobile robots. In: IEEE. *Proceedings 2003 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2003)(Cat. No. 03CH37453)*. [S.l.], 2003. v. 2, p. 1685–1690.
- [Wong e MacDonald 2004]WONG, S. C.; MACDONALD, B. A. Complete coverage by mobile robots using slice decomposition based on natural landmarks. In: SPRINGER. *Pacific Rim International Conference on Artificial Intelligence*. [S.l.], 2004. p. 683–692.
- [Yang 2012]YANG, X.-S. Flower pollination algorithm for global optimization. In: SPRINGER. *International conference on unconventional computing and natural computation*. [S.l.], 2012. p. 240–249.



- [Yang e Gandomi 2012]YANG, X.-S.; GANDOMI, A. H. Bat algorithm: a novel approach for global engineering optimization. *Engineering Computations*, Emerald Group Publishing Limited, v. 29, n. 5, p. 464–483, 2012.
- [Yang e Karamanoglu 2012]YANG, X.-S.; KARAMANOGLU, M. Bat algorithm for topology optimization in microelectronic applications. In: IEEE. *The First International Conference on Future Generation Communication Technologies*. [S.l.], 2012. p. 150–155.
- [Yang et al. 2021]YANG, Y. et al. Hunger games search: Visions, conception, implementation, deep analysis, perspectives, and towards performance shifts. *Expert Systems with Applications*, Elsevier, v. 177, p. 114864, 2021.
- [Zelinsky et al. 1993]ZELINSKY, A. et al. Planning paths of complete coverage of an unstructured environment by a mobile robot. In: *Proceedings of international conference on advanced robotics*. [S.l.: s.n.], 1993. v. 13, p. 533–538.
- [Zhang e Montgomery 1994]ZHANG, W.; MONTGOMERY, D. R. Digital elevation model grid size, landscape representation, and hydrologic simulations. *Water resources research*, Wiley Online Library, v. 30, n. 4, p. 1019–1028, 1994.
- [Zhang et al. 2020]ZHANG, W. et al. Object color recognition and sorting robot based on opencv and machine vision. In: IEEE. *2020 IEEE 11th International Conference on Mechanical and Intelligent Manufacturing Technologies (ICMIMT)*. [S.l.], 2020. p. 125–129.
- [Zhou e Gong 2015]ZHOU, Z.; GONG, J. Image-based 3d reconstruction for posthurricane residential building damage assessment. *Journal of Computing in Civil Engineering*, American Society of Civil Engineers, v. 30, n. 2, p. 04015015, 2015.