
Planejamento de rota para VANTs em caso de
situação crítica: uma abordagem baseada em
segurança

Jesimar da Silva Arantes

SERVIÇO DE PÓS-GRADUAÇÃO DO ICMC-USP

Data de Depósito:

Assinatura: _____

Jesimar da Silva Arantes

Planejamento de rota para VANTs em caso de situação crítica: uma abordagem baseada em segurança¹

Dissertação apresentada ao Instituto de Ciências Matemáticas e de Computação – ICMC-USP, como parte dos requisitos para obtenção do título de Mestre em Ciências – Ciências de Computação e Matemática Computacional. *VERSÃO REVISADA*

Área de Concentração: Ciências de Computação e Matemática Computacional

Orientador: Prof. Dr. Claudio Fabiano Motta Toledo

USP – São Carlos
Maio de 2016

¹ Trabalho realizado com auxílio da FAPESP processo número 2014/12297-0

Ficha catalográfica elaborada pela Biblioteca Prof. Achille Bassi
e Seção Técnica de Informática, ICMC/USP,
com os dados fornecidos pelo(a) autor(a)

A662p Arantes, Jesimar da Silva
Planejamento de rota para VANTs em caso de
situação crítica: uma abordagem baseada em segurança
/ Jesimar da Silva Arantes; orientador Claudio
Fabiano Motta Toledo. - São Carlos - SP, 2016.
158 p.

Dissertação (Mestrado - Programa de Pós-Graduação
em Ciências de Computação e Matemática Computacional)
- Instituto de Ciências Matemáticas e de Computação,
Universidade de São Paulo, 2016.

1. Planejamento de Rota. 2. Veículo Aéreo Não
Tripulado. 3. Otimização. 4. Computação Evolutiva.
5. Pouso Emergencial. I. Toledo, Claudio Fabiano
Motta, orient. II. Título.

Jesimar da Silva Arantes

Route planning for UAVs with risk of critical failure: a security-based approach²

Master dissertation submitted to the Instituto de Ciências Matemáticas e de Computação – ICMC-USP, in partial fulfillment of the requirements for the degree of the Master Program in Computer Science and Computational Mathematics. *FINAL VERSION*

Concentration Area: Computer Science and Computational Mathematics

Advisor: Prof. Dr. Claudio Fabiano Motta Toledo

USP – São Carlos
May 2016

² Trabalho realizado com auxílio da FAPESP processo número 2014/12297-0

À minha família!

*Meu pai Jésus e minha mãe Sirley
que estiveram sempre presentes, me apoiando, incentivando e
ajudando nos momentos em que mais precisei.*

*Meu irmão Márcio pela amizade e ajuda,
e por saber que sempre terei com quem contar.*

AGRADECIMENTOS

À Universidade de São Paulo (USP) e ao Instituto de Ciências Matemáticas e de Computação (ICMC) por me proporcionarem um ambiente de estudo adequado à minha formação. Aos professores da USP, em especial, aos professores do ICMC pelos ensinamentos transmitidos.

Ao professor Claudio Toledo pela orientação, paciência, amizade e ensinamentos, que foram de grande relevância para a realização deste trabalho e para meu crescimento profissional. Ao grupo de pesquisa pelas inúmeras reuniões e apresentações que serviram para ampliação do meu conhecimento nas mais diversas áreas da ciência.

Ao André Pierre pelas muitas contribuições e ensinamentos a respeito de aeronáutica, alcançados através de muitas reuniões. Ao professor Onofre Trindade pelas inúmeras dicas e discussões a respeito de VANTs. Ao professor Eduardo Simões pelo grande auxílio nos experimentos envolvendo *drones*. Ao Marcelo Hossomi pela auxílio prestado no desenvolvimento deste trabalho.

À todos os integrantes do Laboratório de Computação Reconfigurável (LCR) e ao colega de apartamento Francisco de Assis pelas muitas discussões, que, por ventura, contribuíram e auxiliaram nas minhas pesquisas.

Agradeço, sobretudo, ao meu irmão Márcio pelas inumeráveis dicas e discussões, que auxiliaram a realização deste trabalho, além da constante amizade. E à minha namorada Vívian por fazer parte deste momento em minha vida e pelas muitas dicas e correções ao longo do texto.

Este trabalho somente foi possível, pois estive apoiado sobre os ombros desses gigantes. Muito obrigado por tudo!!!

“Se você tem uma maçã e eu tenho uma maçã e nós trocamos as maçãs, então você e eu ainda teremos uma maçã. Mas se você tem uma ideia e eu tenho uma ideia e nós trocamos essas ideias, então cada um de nós terá duas ideias.”

George Bernard Shaw

RESUMO

ARANTES, J. S.. **Planejamento de rota para VANTs em caso de situação crítica: uma abordagem baseada em segurança.** 2016. 158 f. Dissertação (Mestrado em Ciências – Ciências de Computação e Matemática Computacional) – Instituto de Ciências Matemáticas e de Computação (ICMC/USP), São Carlos – SP.

A segurança nos voos de Veículos Aéreos Não Tripulados (VANTs) é uma importante questão e vem ganhando destaque devido a uma série de acidentes com tais aeronaves. O aumento do número de aeronaves no espaço aéreo e a autonomia cada vez maior para realizar missões estão entre outros elementos que merecem destaque. No entanto, pouca atenção tem sido dada a autonomia da aeronave em casos emergenciais **[Contexto]**. Nesse contexto, o desenvolvimento de algoritmos que efetuam o planejamento de rotas na ocorrência de situações críticas é fundamental para obter maior segurança aérea. Eventuais situações de insegurança podem estar relacionadas a uma falha nos equipamentos do veículo aéreo que impede a continuação da missão em curso pela aeronave **[Lacuna]**. A presente pesquisa avança o estado da arte considerando um conceito chamado *In-Flight Awareness* (IFA), que estabelece consciência situacional em VANTs, visando maior segurança de voo. Os estudos também avançam na proposição de modelos matemáticos que representem o estado da aeronave avariada, viabilizando o pouso emergencial e minimizando possíveis danos **[Propósito]**. Este trabalho utiliza técnicas de computação evolutiva como Algoritmos Genéticos (AG) e Algoritmos Genéticos Multi-Populacional (AGMP), além de uma Heurística Gulosa (HG) e um modelo de Programação Linear Inteira Mista (PLIM) no tratamento de falhas críticas juntamente com o conceito de IFA **[Metodologia]**. As soluções obtidas foram avaliadas através de experimentos *offline* usando os modelos matemáticos desenvolvidos, além de validadas em um simulador de voo e em um voo real. De forma geral, o AG e AGMP obtiveram resultados equivalentes, salvando o VANT em aproximadamente 89% dos mapas. A HG conseguiu trazer a aeronave até uma região bonificadora em 77% dos mapas dentro de um tempo computacional abaixo de 1 segundo. No modelo PLIM, o tempo gasto foi de cerca de quatro minutos já que garantia a otimalidade da solução encontrada. Devido ao seu elevado tempo computacional, uma estratégia evolvendo rotas pré-calculadas foi definida a partir do PLIM, mostrando-se bastante promissora. Nos experimentos envolvendo simulador de voo foram testadas diferentes condições de vento e se verificou que mesmo sobre tais condições os métodos desenvolvidos conseguiram efetuar o pouso com segurança **[Resultado]**. O trabalho apresentado colabora com a segurança de Veículos Aéreos Não Tripulados e com a proposta de modelos matemáticos que representem a aeronave em caso de situações críticas. Os métodos, de forma geral, mostraram-se promissores na resolução do problema de pouso emergencial já que trouxeram a aeronave com segurança até regiões interessantes ao pouso em um baixo tempo computacional. Isso foi atestado pelos resultados obtidos a partir das simulações *offline*, em simulador de voo e em voo real **[Conclusão]**. As principais contribuições do trabalho são: modelagem de regiões

adequadas ao pouso, modelagem de falhas, arquitetura do sistema planejador de rotas e modelo linear para para pouso emergencial **[Contribuição]**.

Palavras-chave: Planejamento de Rota, Veículo Aéreo Não Tripulado, Otimização, Computação Evolutiva, Pouso Emergencial.

ABSTRACT

ARANTES, J. S.. **Planejamento de rota para VANTs em caso de situação crítica: uma abordagem baseada em segurança.** 2016. 158 f. Dissertação (Mestrado em Ciências – Ciências de Computação e Matemática Computacional) – Instituto de Ciências Matemáticas e de Computação (ICMC/USP), São Carlos – SP.

The security involved in flights of Unmanned Aerial Vehicles (UAVs) is an important issue and is achieving prominence due to a number of accidents involving such aircraft. Other elements that deserve highlights are the increase in the number of aircraft in the airspace and autonomy to perform missions, however, little attention has been given to the autonomy of the aircraft in emergency cases [**Context**]. In this context, the development of algorithms that contribute significantly to the path planning in the event of critical situations is essential for more air traffic. Possible situations of insecurity may be related to a failure in the equipment of vehicle that prevents the continuation of the current mission by aircraft [**Gap**]. The research advances the state of the art considering a concept called In-Flight Awareness (IFA), which provides situational awareness in UAVs aiming at greater flight safety. Advances also in the developing of mathematical models that represent the state of the damaged aircraft, with the purpose to execute the emergency landing by minimizing damages [**Purpose**]. Thus, this work applies evolutionary computation techniques such as Genetic Algorithms (GA) and Multi-Population Genetic Algorithms (MPGA), as well as a Greedy Heuristic (GH) and a Mixed Integer Linear Programming (MILP) model to deal with critical situations along with the concept of IFA [**Methodology**]. The solutions obtained were evaluated through offline experiments using the developed mathematical models, which were validated in a flight simulator and a real-world flight. In General, the GA and MPGA reached similar results by saving the UAV in approximately 89% of the maps, while the GH was able to bring the aircraft to a bonus region for 77% of maps within a feasible computational time lower than 1 second. In the MILP model, the time spent was about four minutes since it guarantees optimality of the solution found. Due to such high computational time, a strategy involving nearby routes pre-calculated was defined from the MILP which was very promising. In experiments involving flight simulator, different wind conditions were tested and it was found that even under such conditions the methods developed have managed to execute the landing safely [**Result**]. The work presented collaborates with the safety of Unmanned Aerial Vehicles and with the proposal of mathematical models that represent the aircraft under critical situations. The methods, in general, were promising since they brought the aircraft to execute a safe landing within a low computational time as shown by offline simulations, flight simulator and real flight [**Conclusion**]. The main contributions are: fault modeling, system architecture planner routes and linear model for emergency landing. [**Contribution**].

Key-words: Route Planning, Unmanned Aerial Vehicle, Optimization, Evolutionary Computation, Emergency Landing.

LISTA DE ILUSTRAÇÕES

Figura 1 – Relação desse trabalho com trabalhos do grupo de pesquisa.	33
Figura 2 – Veículos Aéreos Não Tripulados Arara M1, Tiriba e Ararinha.	38
Figura 3 – Eixos da aeronave e superfícies de controle da aeronave.	40
Figura 4 – Sistemas de coordenadas da aeronave e do mundo.	40
Figura 5 – Esquema da integração entre sensores, atuadores, IFA, AP e MOSA na aeronave.	43
Figura 6 – Simulador de voo FlightGear.	44
Figura 7 – Simulador de voo X-Plane.	44
Figura 8 – Simulador de voo Microsoft Flight Simulator.	45
Figura 9 – Cenário geral do problema abordado.	52
Figura 10 – Cenário para o planejamento de missão com replanejamento de rota em caso de situação crítica.	53
Figura 11 – Representação de regiões factíveis. Cada região factível foi aproximado por um triângulo e por um conjunto de restrições lineares. (a) região factível fora do obstáculo. (b) região factível dentro do obstáculo.	59
Figura 12 – Ilustração do modelo de planejamento com incremento da incerteza em cada instante de tempo.	60
Figura 13 – Formato dos limitantes das velocidades para o modelo matemático. Em (a), (b) e (c) são apresentados diferentes visões do mesmo poliedro/polígono das velocidades. Esse polígono é aproximado através de Γ faces.	65
Figura 14 – Formato dos limitantes das acelerações para o modelo matemático. Em (a), (b) e (c) são apresentados diferentes visões do mesmo poliedro/polígono das acelerações. Esse polígono é aproximado através de Γ faces.	66
Figura 15 – Exemplo da modelagem da curva efetuada a esquerda.	71
Figura 16 – Exemplo da modelagem da curva efetuada a direita.	72
Figura 17 – Codificação e decodificação dos controles na rota da aeronave utilizada na modelagem restrita.	76
Figura 18 – Rotas candidatas usando a estratégia Heurística Gulosa.	79
Figura 19 – Estrutura hierárquica em árvore usada no AGMP.	84
Figura 20 – Esquema das rotas emergenciais para a missão definida para as falhas Ψ^m , Ψ^b , Ψ^{s1} e Ψ^{s2} , respectivamente.	87
Figura 21 – Arquitetura do planejador de rota de pouso proposto.	88
Figura 22 – Estratégia para construção manual de mapas contendo regiões de interesse.	89
Figura 23 – Exemplos de mapas gerados automaticamente.	91

Figura 24 – Mapeamento dos cenários 2D para 3D. (a) mapa de origem 2D. (b) mapa de destino mapeado em 3D.	94
Figura 25 – Estratégia de pouso 2D para diferentes tempos gastos no pouso.	95
Figura 26 – Estratégia de pouso 3D em terreno plano	97
Figura 27 – Tipos de experimentos realizados.	99
Figura 28 – Análise das forças envolvidas na aeronave.	101
Figura 29 – Mudança de velocidade do VANT partindo do repouso e com aceleração constante.	101
Figura 30 – <i>Framework</i> ProOF utilizado no desenvolvimento dos métodos.	102
Figura 31 – Resultados da distribuição do <i>fitness</i> dos indivíduos, após a primeira geração. Os mapas foram escolhidos arbitrariamente.	107
Figura 32 – Resultados da distribuição do <i>fitness</i> dos indivíduos, após a última geração. Os mapas utilizados foram os mesmos do experimento anterior.	109
Figura 33 – Exemplos de rotas para cada uma das situações críticas.	112
Figura 34 – Exemplos de rotas obtidas pela HG.	114
Figura 35 – Exemplos de rotas obtidas pelo AG.	114
Figura 36 – Exemplos de rotas obtidas pelo AGMP.	117
Figura 37 – Simulação usando o FlightGear para seguir a rota gerada pelo AGMP. . . .	118
Figura 38 – Simulação no FlightGear usando o AGMP. Em verde tem as rotas percorridas pela Cessna e em vermelho as geradas pelo AGMP. (a) e (c) ventos de 10 knots. (b) e (d) ventos de 50 knots. Direção do vento: oeste.	119
Figura 39 – Todas as rotas obtidas pela HG. A melhor rota é retornada pela HG.	120
Figura 40 – Conjunto de rotas obtidas pelo AGMP. Estão plotados as 100 últimas rotas geradas até o número da avaliação indicado.	121
Figura 41 – Conjunto de todos os locais de pouso obtido pelo AGMP em 10.000 avaliações efetuadas.	122
Figura 42 – Análise do número de variáveis contínuas e binárias e número de restrições. .	124
Figura 43 – Evolução do objetivo em função do tempo para o problema no motor no modelo PLIM1.	127
Figura 44 – Evolução do objetivo em função do tempo para o problema na bateria no modelo PLIM1.	127
Figura 45 – Evolução do objetivo em função do tempo para o problema na superfície aerodinâmica tipo 1 no modelo PLIM1.	128
Figura 46 – Evolução do objetivo em função do tempo para o problema na superfície aerodinâmica tipo 2 no modelo PLIM1.	128
Figura 47 – Evolução do objetivo em função do tempo para as falhas Ψ^m , Ψ^b , Ψ^{s1} e Ψ^{s2} no modelo PLIM1.	129
Figura 48 – Evolução do objetivo em função do tempo para diferentes falhas no modelo PLIM2.	129

Figura 49 – Análise de distribuição do tempo.	131
Figura 50 – Análise de distribuição do objetivo.	131
Figura 51 – Rotas de pouso obtidas para o modelo PLIM2.	133
Figura 52 – Rota obtida na situação crítica Ψ^{s1} em ambiente tridimensional.	134
Figura 53 – Simulação usando o FlightGear seguindo uma rota gerada pelo método PLIM.	134
Figura 54 – Cálculo do erro horizontal da aeronave em relação a rota gerada.	136
Figura 55 – Distribuição do erro horizontal obtidos nas simulações.	137
Figura 56 – Exemplos de rotas obtidas nos experimentos no FlightGear. (a) e (b) rotas em diferentes mapas com falhas na bateria. Em (c) tem-se a rota (a) ampliada. Em (d) tem-se a rota (b) ampliada.	138
Figura 57 – Resultados das simulações pré-processadas em uma série de pontos-chave da rota.	139
Figura 58 – Exemplo de rotas obtidas pelo SCADE.	140
Figura 59 – Máquina de estados implementada usando o SCADE.	141
Figura 60 – Experimento com voo real usando um Hexacóptero.	142
Figura 61 – Síntese dos resultados para a modelagem restrita do problema.	144
Figura 62 – Síntese dos resultados para a modelagem completa do problema.	144

LISTA DE ALGORITMOS

Algoritmo 1 – Heurística Gulosa.	79
Algoritmo 2 – Algoritmo Genético.	81
Algoritmo 3 – Algoritmo Genético Multi-Populacional.	83
Algoritmo 4 – Algoritmo do Modelo PLIM.	86
Algoritmo 5 – Algoritmo Gerador Automático de Mapas.	92

LISTA DE TABELAS

Tabela 1 – Detalhes técnicos da aeronave Tiriba.	39
Tabela 2 – Detalhes de acidentes envolvendo VANT militares.	41
Tabela 3 – Comparações entre os simuladores de voo: FightGear, X-Plane e Microsoft Flight Simulator.	45
Tabela 4 – Configurações do gerador de mapas para cada tipo de categoria.	93
Tabela 5 – Notação utilizada em cada instância com sua descrição.	93
Tabela 6 – Parâmetros e valores das alturas das regiões no mapa 3D.	94
Tabela 7 – Configurações dos pesos de cada região do mapa.	99
Tabela 8 – Configurações do mapa, VANT, modelo e pesos das regiões usados.	106
Tabela 9 – Configurações do computador utilizado nos experimentos.	106
Tabela 10 – Análise de sensibilidade de parâmetros do AGMP e AG.	110
Tabela 11 – Resultado obtido para diferentes situações críticas enfrentadas pelo VANT. .	111
Tabela 12 – Resultado obtido para a HG, para todos as situações críticas e todas as instâncias.	113
Tabela 13 – Resultado obtido para o AG1 e AG2, para todos as situações críticas e todas as instâncias.	115
Tabela 14 – Resultado obtido para o AGMP1 e AGMP2, para todos as situações críticas e todas as instâncias.	116
Tabela 15 – Dados do local de pouso das 10.000 rotas gerados pelo AGMP.	121
Tabela 16 – Configurações do mapa, VANT, modelo, <i>solver</i> e pesos usados.	123
Tabela 17 – Configurações do computador utilizado nos experimentos.	124
Tabela 18 – Resultados obtidos para o modelo PLIM1 para todas as situações críticas e todas as instâncias.	125
Tabela 19 – Resultados obtidos para o modelo PLIM2 para todas as situações críticas e todas as instâncias.	126
Tabela 20 – Análises dos experimentos do PLIM2 para todas as situações críticas e todas a instâncias em relação ao tempo para a primeira e última solução factível. .	130
Tabela 21 – Análises dos experimentos do PLIM2 para todas as situações críticas e todas a instâncias em relação ao objetivo para a primeira e última solução factível.	132
Tabela 22 – Resultados obtidos nos experimentos utilizando simulador de voo com o modelo PLIM2.	135
Tabela 23 – Resultados obtidos para as rotas pré-processadas pelo modelo PLIM2. . .	138
Tabela 24 – Síntese dos resultados para todos os métodos desenvolvidos na modelagem restrita e completa.	143

LISTA DE ABREVIATURAS

2D	Duas Dimensões
3D	Três Dimensões
ACO	Otimização por Colônia de Formigas (<i>Ant Colony Optimization</i>)
AED	Algoritmos de Estimação de Distribuição (<i>Estimation of Distribution Algorithm</i>)
AG	Algoritmo Genético (<i>Genetic Algorithm</i>)
AGMP	Algoritmo Genético Multi-Populacional (<i>Multi-Population Genetic Algorithm</i>)
AIB	Conselho de Investigação de Acidentes (<i>Accident Investigation Board</i>)
ANAC	Agência Nacional de Aviação Civil
AP	Piloto Automático (Autopilot)
ARARA	Aeronave de Reconhecimento Autônoma e Remotamente Assistida
ATC	Controle de Tráfego Aéreo (<i>Air Traffic Control</i>)
B&B	(<i>Branch and Bound</i>)
B&C	(<i>Branch and Cut</i>)
CBT	Companhia Brasileira de Tratores
CENIPA	Centro de Investigação e Prevenção de Acidentes Aeronáuticos
CS	Consciência Situacional (<i>Situation Awareness</i>)
CTOL	Decolagem e Aterrissagem Convencional (<i>Conventional Take-off and Landing</i>)
ED	Evolução Diferencial (<i>Differential Evolution</i>)
EE	Estratégia Evolutiva (<i>Evolutionary Strategy</i>)
PPE	Planejador de Pouso de Emergência (<i>Emergency Landing Planner</i>)
EMBRAPA	Empresa Brasileira de Pesquisa Agropecuária
FAA	Administração Federal de Aviação (<i>Federal Aviation Administration</i>)
FG	FlighthGear
GISA	Grupo de Interesse em SisVANTs e Aplicações
GPL	Licença Pública Geral (<i>General Public License</i>)
GPS	Sistema de Posicionamento Global (<i>Global Positioning System</i>)
HG	Heurística Gulosa (<i>Greedy Heuristic</i>)
HIL	(<i>Hardware-In-the-Loop</i>).
ICAO	Organização de Aviação Civil Internacional (<i>International Civil Aviation Organization</i>)
ICMC	Instituto de Ciências Matemáticas e Computação

IE	Inteligência de Enxame (<i>Swarm Intelligence</i>)
IFA	Consciência em Voo (<i>In-Flight Awareness</i>)
IFA ² S	(<i>In-Flight Awareness Augmentation Systems</i>)
LCR	Laboratório de Computação Reconfigurável
MOSA	Arranjos de Sensores Orientados à Missão (<i>Mission Oriented Sensor Array</i>)
MSFS	Microsoft Flight Simulator
OEP	Otimização por Enxame de Partículas (<i>Particle Swarm Optimization</i>)
PE	Programação Evolutiva (<i>Evolutionary Programming</i>)
PG	Programação Genética (<i>Genetic Programming</i>)
PID	Proporcional Integral Derivativo (<i>Proportional Integral Derivative</i>)
PIM	Programação Inteira Mista (<i>Mixed-Integer Programming</i>)
PLIM	Programação Linear Inteira Mista (<i>Mixed-Integer Linear Programming</i>)
SEE	Sistemas Embarcados e Evolutivos
SisVANT	Sistema de Veículo Aéreo Não Tripulado
SSC	Departamento de Sistemas de Computação
S&A	Detectar e Evitar (<i>Sense and Avoid</i>)
USAF	Força Aérea dos Estados Unidos (<i>United States Air Force</i>)
USP	Universidade de São Paulo
VANT	Veículo Aéreo Não Tripulado (<i>Unmanned Aerial Vehicle</i>)
VTNT	Veículos Terrestres Não-Tripulados (<i>Unmanned Ground Vehicles</i>)
VTOL	Decolagem e Aterrissagem Vertical (<i>Vertical Take-off and Landing</i>)

LISTA DE SÍMBOLOS

Φ	Região do mapa cadastrada ($\Phi = \{\Phi^n, \Phi^p, \Phi^b, \Phi^r\}$)
Φ^n	Conjunto de regiões do tipo não navegável (proibido sobrevoar esta região)
Φ^p	Conjunto de regiões do tipo penalizadora (região péssima para pouso)
Φ^b	Conjunto de regiões do tipo bonificadora (região boa para pouso)
Φ^r	Conjunto de regiões do tipo remanescente (região não classificada)
Φ^*	Local do pouso emergencial da aeronave
Φ^j	Conjunto de regiões do mapa $\Phi^j = \{Z_{\Phi^j}^1, Z_{\Phi^j}^2, \dots, Z_{\Phi^j}^{ \Phi^j }\}$ com $j \in \{n, p, b, r\}$
$Z_{\Phi^j}^i$	i -ésima região do conjunto Φ^j
C_{Φ^j}	Custo de pousar no conjunto Φ^j
$ \Phi^j $	Número de regiões no conjunto Φ^j
Ψ	Situação crítica ocorrida na aeronave ($\Psi = \{\Psi^m, \Psi^b, \Psi^{s1}, \Psi^{s2}, \Psi^\emptyset\}$)
Ψ^m	Situação crítica para o problema no motor
Ψ^b	Situação crítica para o problema na bateria
Ψ^{s1}	Situação crítica para o problema nas superfícies aerodinâmicas tipo 1
Ψ^{s2}	Situação crítica para o problema nas superfícies aerodinâmicas tipo 2
Ψ^\emptyset	Condição de pouso normal sem situação crítica
Ψ^*	Situação crítica a que foi submetida a aeronave ($\Psi^* \in \Psi$)
F_Ψ	Função de transição de estados para falhas do tipo Ψ
Δ	Probabilidade do VANT violar uma região do conjunto Φ^n
T	Número de passos de tempo para pouso do VANT
K	Instante no tempo no qual a aeronave caiu ($0 < K \leq T$)
ΔT	Discretização do tempo de simulação com intervalo de tempo fixo
x_t	Estado da aeronave no tempo t no espaço 2D ($x_t = [p_t^x \ p_t^y \ v_t \ \alpha_t]$)
\bar{x}_t	Estado esperado da aeronave no tempo t no espaço 3D ($\bar{x}_t = [p_t^x \ p_t^y \ p_t^z \ v_t^x \ v_t^y \ v_t^z]$)
p^x	Posição no eixo x do VANT
p^y	Posição no eixo y do VANT
p^z	Posição no eixo z do VANT
v	Velocidade na horizontal do VANT
α	Ângulo (direção) na horizontal do VANT
v^x	Velocidade no eixo x do VANT
v^y	Velocidade no eixo y do VANT
v^z	Velocidade no eixo z do VANT

u_t	Controles da aeronave no tempo t no espaço 2D ($u_t = [a_t \ \boldsymbol{\varepsilon}_t]$)
\bar{u}_t	Controles esperados da aeronave no tempo t no espaço 3D ($\bar{u}_t = [a_t^x \ a_t^y \ a_t^z]$)
a	Aceleração na horizontal do VANT
$\boldsymbol{\varepsilon}$	Variação angular (rotação) do VANT em torno do eixo z
a^x	Aceleração no eixo x do VANT
a^y	Aceleração no eixo y do VANT
a^z	Aceleração no eixo z do VANT
ω_t	Perturbação independente do estado no instante t
Σ_{x_0}	Matriz da covariância associada a incerteza do VANT
Q	Matriz da covariância associada a incerteza do ambiente
F^d	Força do arrasto determinar a resistência do ar associada ao VANT
C_d	Coeficiente de arrasto específico para cada aeronave
ρ	Massa específica do fluido (ar) que a aeronave está inserida
\mathcal{A}	Área de referência do VANT quando imerso no fluido (ar)
K_d	Constante de arrasto associada a cada aeronave
m	Massa do VANT
a_{max}	Aceleração máxima que o VANT alcança
v_{term}	Velocidade terminal do VANT (velocidade que não se consegue ultrapassar)
Ω	Trajetória de pouso seguida pela aeronave (conjunto de <i>waypoints</i>)
\mathcal{W}	Objetivo ou <i>fitness</i> associada a solução
G	Conjunto de vetores normais que definem as arestas do polígono
A	Matriz 6×6 para o estado do VANT
B	Matriz 6×3 para o controle da dinâmica do VANT
Σ_t	Matriz 6×6 da covariância associada a posição do VANT
Γ	Número de lados do polígono usado na linearização
∇	Número de arestas dos polígonos usados nos mapas
η	Crescimento da incerteza a cada instante de tempo
τ^{min}	Tempo mínimo de voo
τ^{max}	Tempo máximo de voo
v_{max}^h	Velocidade horizontal máxima permitida para o VANT
v_{max}^v	Velocidade vertical máxima permitida para o VANT
v_{min}^h	Velocidade horizontal mínima permitida para o VANT
v_{min}^v	Velocidade vertical mínima permitida para o VANT
a_{max}^h	Aceleração horizontal máxima permitida para o VANT
a_{max}^v	Aceleração vertical máxima permitida para o VANT
σ^h	Incerteza associada a aeronave nas coordenadas (x, y)
σ^v	Incerteza associada a aeronave no eixo z

a_k	Vetor normal à aresta k
a_{ji}	Vetor normal à aresta i do polígono j
b_{ji}	Valor constante da aresta i do polígono j
G_{max}^v	Hiperplanos das faces externas do polígono das velocidades
G_{min}^v	Hiperplanos das faces internas do polígono das velocidades
G_{max}^a	Hiperplanos das faces externas do polígono dos controles
h^j	Altitude da região j
$h[v]$	Altitude de uma região qualquer do mapa, onde v representa a altura
$P_j^{\Phi^b}$	Probabilidade do VANT pousar na região bonificadora j
$L_j^{\Phi^b}$	Controla o pouso em apenas uma região bonificadora j
S_t	Possui o valor 1 apenas no instante exato que o VANT pousou
δ_{jt}^n	Risco associado à colisão com os lados da região não navegável j no instante t
δ_{jt}^p	Risco associado à colisão com os lados da região penalizadora j no instante t
δ_{jit}^b	Risco associado a violação do hiperplano i da região bonificadora j no instante t
H_t	1 indica VANT em voo, 0 indica pouso do VANT
W_{kt}	1 indica superar o limite inferior de velocidade no hiperplano k no instante t
$Y_{jt}^{\Phi^b}$	1 queda do VANT na j -ésima região de Φ^b no instante t
$Z_{jit}^{\Phi^n}$	1 indica estar fora do hiperplano i da j -ésima região de Φ^n no instante t
$Z_{jit}^{\Phi^p}$	1 indica estar fora do hiperplano i da j -ésima região de Φ^p no instante t
$\mathcal{M}_F^{25\%}$	Mapas com dificuldade fácil e com área de cobertura de 25%
$\mathcal{M}_F^{50\%}$	Mapas com dificuldade fácil e com área de cobertura de 50%
$\mathcal{M}_N^{25\%}$	Mapas com dificuldade normal e com área de cobertura de 25%
$\mathcal{M}_N^{50\%}$	Mapas com dificuldade normal e com área de cobertura de 50%
$\mathcal{M}_D^{25\%}$	Mapas com dificuldade difícil e com área de cobertura de 25%
$\mathcal{M}_D^{50\%}$	Mapas com dificuldade difícil e com área de cobertura de 50%
I1	Instância do grupo de mapas $\mathcal{M}_F^{25\%}$
I2	Instância do grupo de mapas $\mathcal{M}_F^{50\%}$
I3	Instância do grupo de mapas $\mathcal{M}_N^{25\%}$
I4	Instância do grupo de mapas $\mathcal{M}_N^{50\%}$
I5	Instância do grupo de mapas $\mathcal{M}_D^{25\%}$
I6	Instância do grupo de mapas $\mathcal{M}_D^{50\%}$

SUMÁRIO

1	INTRODUÇÃO	31
1.1	Contextualização	31
1.2	Motivação e Justificativas	33
1.3	Objetivos	34
1.4	Lacunas	34
1.5	Contribuições e Limitações	35
1.6	Organização	35
2	REVISÃO BIBLIOGRÁFICA	37
2.1	Introdução	37
2.2	Conceitos Importantes	37
2.2.1	<i>Veículos Aéreos Não Tripulados</i>	37
2.2.1.1	<i>Histórico de VANTs no ICMC</i>	38
2.2.1.2	<i>Componentes da Aeronave e Sistemas de Coordenadas</i>	39
2.2.1.3	<i>Situações Críticas em VANTs</i>	39
2.2.2	<i>In-Flight Awareness</i>	41
2.3	Simuladores de Voo	43
2.3.1	<i>FlightGear</i>	43
2.3.2	<i>X-Plane</i>	44
2.3.3	<i>Microsoft Flight Simulator</i>	44
2.3.4	<i>Comparação dos Simuladores</i>	45
2.4	Trabalhos Relacionados	46
2.5	Conclusão	49
3	PROBLEMA ABORDADO	51
3.1	Introdução	51
3.2	Descrição do Problema	51
3.2.1	<i>Conjunto de Regiões Modeladas</i>	52
3.2.2	<i>Situações Críticas Modeladas</i>	54
3.2.3	<i>Considerações Sobre as Condições de Voo e Ambiente</i>	55
3.3	Modelagem Restrita do Problema	55
3.4	Modelagem Completa do Problema	59
3.5	Conclusão	72

4	METODOLOGIA	75
4.1	<i>Introdução</i>	75
4.2	<i>Codificação do Problema</i>	75
4.2.1	<i>Função Objetivo</i>	77
4.3	<i>Heurística Gulosa</i>	78
4.4	<i>Algoritmo Genético</i>	80
4.5	<i>Algoritmo Genético Multi-Populacional</i>	82
4.6	<i>Programação Linear Inteira Mista</i>	84
4.7	<i>Arquitetura Proposta</i>	86
4.8	<i>Geração Manual de Mapas</i>	88
4.9	<i>Geração Automática de Mapas</i>	90
4.9.1	<i>Classificação dos Mapas</i>	90
4.9.2	<i>Algoritmo de Geração de Mapas</i>	91
4.9.3	<i>Configurações Usadas na Geração dos Mapas</i>	92
4.9.4	<i>Mapeamento de Cenários 2D para 3D</i>	93
4.10	<i>Estratégias de Pouso</i>	94
4.10.1	<i>Estratégia para Pouso em mapas 2D</i>	94
4.10.2	<i>Pouso 3D</i>	96
4.11	<i>Considerações Gerais</i>	96
4.11.1	<i>Considerações Sobre o VANT</i>	96
4.11.2	<i>Considerações Sobre os Experimentos Realizados</i>	98
4.11.3	<i>Considerações Sobre os Pesos das Regiões</i>	99
4.12	<i>Considerações Sobre a Constante de Arrasto</i>	100
4.13	<i>Framework ProOF</i>	102
4.14	<i>Conclusão</i>	102
5	RESULTADOS	105
5.1	<i>Introdução</i>	105
5.2	<i>Experimentos da Modelagem Restrita</i>	105
5.2.1	<i>Configurações Utilizadas</i>	105
5.2.2	<i>Análise da Distribuição do Fitness dos Indivíduos</i>	106
5.2.3	<i>Análise de Sensibilidade do AG e AGMP</i>	108
5.2.4	<i>Exemplo de Soluções para Situações Críticas</i>	110
5.2.5	<i>Heurística Gulosa</i>	111
5.2.6	<i>Algoritmo Genético</i>	113
5.2.7	<i>Algoritmo Genético Multi-Populacional</i>	115
5.2.8	<i>Experimentos Online</i>	117
5.2.9	<i>Estudo das Soluções do HG e AGMP</i>	118
5.3	<i>Experimentos com Modelagem Completa</i>	121
5.3.1	<i>Configurações Utilizadas</i>	122

5.3.2	<i>Análise do Modelo</i>	123
5.3.3	<i>Experimentos Offline</i>	124
5.3.4	<i>Experimentos Online</i>	132
5.3.5	<i>Experimentos com Rotas Pré-Processadas</i>	137
5.4	Experimentos Usando SCADE	139
5.5	Experimentos com Voo Real	142
5.6	Síntese dos Resultados	143
5.7	Conclusão	143
6	CONSIDERAÇÕES FINAIS	145
6.1	Conclusões	145
6.2	Artigos	147
6.3	Trabalhos Futuros	147
	REFERÊNCIAS	149
7	GLOSSÁRIO	157



INTRODUÇÃO

*“Não se espante com a altura do voo.
Quanto mais alto, mais longe do perigo.
Quanto mais você se eleva, mais tempo há
de reconhecer uma pane. É quando se está
próximo do solo que se deve desconfiar.”*

Santos Dumont

1.1 Contextualização

A presente dissertação estuda o planejamento de rota para Veículos Aéreos Não Tripulados (VANTs) após a ocorrência de uma situação crítica. Nesse caso, a missão desempenhada pela aeronave precisa ser abortada e um replanejamento da rota se faz necessário. O principal objetivo passa a ser pousar a aeronave sem riscos à segurança de pessoas, de propriedades em solo e da própria aeronave. A situação crítica pode ser representada por falhas em sensores, pane no motor, superaquecimento da bateria, entre outras situações que obriguem o VANT a realizar um pouso emergencial ou a retornar à base. Diversas falhas serão estudadas e algoritmos capazes de realizar o planejamento da rota de aterrissagem com segurança serão propostos.

A crescente utilização de VANTs gerou discussões e questionamentos sobre os riscos envolvidos em sua operação sobre regiões povoadas. A segurança de voo é definida pela Organização de Aviação Civil Internacional (ICAO) como “estado em que a possibilidade de danos a pessoas ou propriedades é reduzido a um nível aceitável, através de um processo contínuo de identificação de perigos e gestão de riscos de segurança” (ICAO, 2013, pág. 17). Para a integração dos mesmos ao espaço aéreo, a probabilidade de falha deve ser menor ou igual àquelas atualmente aceitas para a aviação geral (NATO, 2007; FAA, 2012).

Este trabalho faz uso de um conceito chamado *In-Flight Awareness* (IFA), definido em Rodrigues *et al.* (2011). O IFA amplia o conceito de *Sense and Avoid* (S&A) e estabelece consciência situacional em aeronaves não tripuladas. O S&A expressa a ideia de Detectar e Evitar que, segundo Mattei (2015), significa ter a capacidade de detectar riscos e tomar decisões

adequadas visando segurança. Uma vez que a segurança é o fator mais importante na integração dos VANTs ao espaço aéreo, torna-se necessário criar formas de trazer novamente a bordo os recursos eventualmente perdidos com a saída dos pilotos (MATTEI *et al.*, 2013; FONSECA; MATTEI; CUNHA, 2013; FIGUEIRA *et al.*, 2013).

O voo pode ser modelado como um sistema dinâmico e estocástico, influenciado por fatores internos e externos (LI, 2010; ONO; WILLIAMS; BLACKMORE, 2013; BLACKMORE; ONO; WILLIAMS, 2011; RICHARDS; HOW, 2002) à aeronave. O objetivo deste trabalho é maximizar a segurança dos voos através da elaboração de algoritmos que permitam conduzir o VANT em situação crítica até uma região segura para o pouso. Deve-se ressaltar que, mesmo durante situações críticas, o VANT pode ser capaz de decidir ativar procedimentos que permitam guiar a aeronave até um local seguro, antes de acionar mecanismos como paraquedas. Assim, os métodos desenvolvidos acionarão os procedimentos mais adequados de forma a privilegiar os aspectos de segurança.

As condições em que se efetuará o pouso forçado serão baseadas no seguinte cenário geral: um veículo aéreo está sobrevoando uma determinada região. Durante o voo, alguma situação crítica à operação da aeronave é detectada. A rota em curso será abortada e uma nova rota chamada de pouso emergencial será estabelecida.

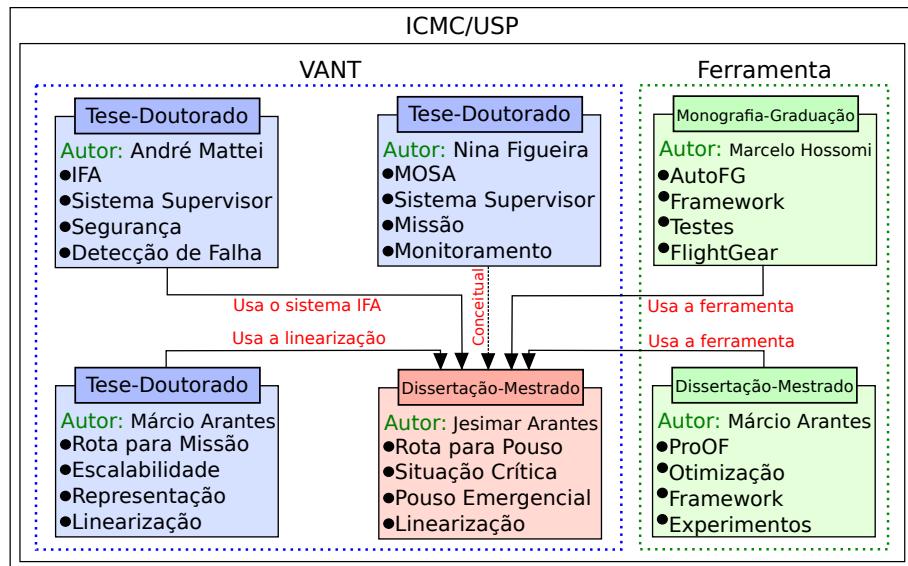
A presente dissertação trabalhará na etapa de estabelecimento do planejamento do voo em caso de situação crítica, propondo algoritmos que levem o VANT até um lugar considerado seguro para o pouso.

A metodologia de desenvolvimento utilizada neste trabalho está relacionada a técnicas de Computação Evolutiva (CE), utilizadas de forma promissora em planejamento de rotas para VANTs em trabalhos anteriores como Pehlivanoglu (2012), Tuncer e Yildirim (2012), Besada-Portas *et al.* (2013), Zhang e Duan (2015), Varela *et al.* (2014). A dissertação apresenta duas estratégias baseadas em CE que são: Algoritmos Genéticos (AGs) e Algoritmos Genéticos Multi-Populacionais (AGMPs). Um método heurístico utilizando uma estratégia gulosa, chamada Heurística Gulosa (HG), também foi desenvolvido. Por último, um método exato baseado em Programação Linear Inteira Mista (PLIM) foi proposto a partir da formulação matemática do problema estudado.

Essa dissertação converge diversos resultados obtidos recentemente nas pesquisas em VANTs conduzidas pelo grupo de Sistemas Embarcados e Evolutivos (SEE) do Departamento de Sistemas de Computação (SSC) no ICMC/USP. A Figura 1 apresenta um esquema da relação desta dissertação com alguns trabalhos do grupo. O presente trabalho se relaciona com o projeto de doutorado do ex-aluno André Pierre (MATTEI, 2015) ao utilizar o sistema IFA. Ele se relaciona também com o trabalho de doutorado do aluno Márcio Arantes (ARANTES, 2015) ao fazer a linearização do problema de planejamento de rotas com alocação de risco. Uma relação mais conceitual é estabelecida com o trabalho de doutorado da ex-aluna Nina Figueira (FIGUEIRA, 2015). Também foram utilizadas duas ferramentas desenvolvidas pelo grupo de

pesquisa. Uma ferramenta é o ambiente de simulação AutoFG desenvolvido pelo ex-aluno de graduação Marcelo Hossomi. A outra é o *framework* para otimização ProOf desenvolvido pelo ex-aluno de mestrado Márcio Arantes.

Figura 1 – Relação desse trabalho com trabalhos do grupo de pesquisa.



Fonte: Elaborada pelo autor.

1.2 Motivação e Justificativas

Uma motivação para este trabalho é o elevado índice de acidentes com VANTs comparados aos índices encontrados em aeronaves tripuladas. Uma parcela considerável desses acidentes é ocasionada pela tomada de decisões equivocadas dos pilotos localizados em terra. Desse modo, surge a necessidade de se construir sistemas planejadores de rotas capazes de oferecer alternativas ao piloto em solo ou executar o planejamento de uma nova rota de forma autônoma. Outra motivação é que os VANTs devem ser o tipo predominante de aeronaves em um futuro próximo, o que gera preocupação quanto a segurança do ambiente aéreo.

Uma série de acidentes envolvendo *drones*, usados para fins recreativos, vitimaram pessoas Smartdrones (2014c), Smartdrones (2014b), Smartdrones (2014a), Mortimer (2012). Há também registros de acidentes com VANTs em aplicações militares, revelando a ocorrência de mais de 400 acidentes em todo o mundo desde 2001 (OGLOBO, 2014).

Um elemento que justifica o estudo do replanejamento de rotas na ocorrência de situações críticas é a ausência de trabalhos a respeito desse tema. O comportamento da aeronave diante de cada tipo de problema exige um nível maior de compreensão quanto a forma de executar determinada rota de pouso, por exemplo, dado um certo tipo de falha no equipamento. De acordo com o Centro de Investigação e Prevenção de Acidentes Aeronáuticos (CENIPA), as principais avarias que levaram aeronaves civis à queda são: falhas no motor em voo 21,6%; perda do

controle em voo 19,7%; perda do controle em solo 11,5%; colisão em voo controlado com o terreno 9,6% ([CENIPA, 2012](#), pág. 17). Todas essas falhas somam mais de 60% dos acidentes envolvendo aeronaves tripuladas.

O entendimento de cada uma dessas falhas no contexto dos VANTs e o desenvolvimento de algoritmos capazes de lidar com cada uma delas são fatores relevantes que justificam o presente estudo. Estudos desse tipo poderão auxiliar tanto na maior autonomia de voo quanto na melhoria do nível de segurança.

1.3 Objetivos

Este trabalho visa promover maior segurança aos voos de VANTs, projetando algoritmos planejadores de rotas que levem em conta as principais situações críticas que podem ocorrer em tais aeronaves.

Nesta dissertação, foi estabelecida como meta atingir os seguintes objetivos específicos:

- Investigar e elencar as principais situações críticas que levam os VANTs a queda e suas causas;
- Desenvolver modelos matemáticos que representem os controles, os estados e a dinâmica da aeronave, considerando as situações críticas que podem ocorrer durante seu voo;
- Elaborar mapas para voo que contenham um conjunto de regiões diversificadas ao voo e pouso do VANT, através de um gerador automático de mapas;
- Avaliar e implementar uma série de algoritmos planejadores de rotas como Heurística Gulosa (HG), Algoritmo Genético (AG), Algoritmo Genético Multi-Populacional (AGMP) e Programação Linear Inteira Mista (PLIM) capazes de tratar as principais situações emergenciais consideradas;
- Avaliar o desempenho e comportamento dos algoritmos implementados em experimentos *offline* e *online*;
- Comparar e discutir os resultados obtidos para as diferentes estratégias de pouso nos mais diferentes cenários.

1.4 Lacunas

A pesquisa efetuada durante a dissertação mostrou que existem vários trabalhos que tratam de planejamento de caminho como [Blackmore, Ono e Williams \(2011\)](#) e planejamento de missão como [Ono, Williams e Blackmore \(2013\)](#), [Li \(2010\)](#) para VANTs, no entanto, esses trabalhos geralmente, apresentam o planejamento sem considerar avaria na aeronave. Existem

algumas pesquisas que lidam com o problema de pouso em caso de emergência como Meuleau *et al.* (2009), Meuleau *et al.* (2011), mas lidam com aeronaves tripuladas e também não tratam falhas que levem a queda imediata. Existem poucos trabalhos estudando o pouso emergencial de VANTs como Li, Chen e Li (2014) e Li (2013), contudo, os mesmos apresentam algumas limitações nos experimentos realizados ao não definirem bem as falhas a que as aeronaves estariam sujeitas.

Sendo assim, a pequena quantidade de trabalhos que lida com situações críticas e planejamento de rotas de pouso para VANTs é uma lacuna a ser preenchida e que motivou o desenvolvimento deste trabalho.

1.5 Contribuições e Limitações

Espera-se contribuir com o desenvolvimento de modelos matemáticos que representem a dinâmica da aeronave avariada e determinem uma rota de pouso adequada para cada situação crítica. Além disso, métodos heurísticos e metaheurísticos serão propostos visando a obtenção de soluções de boa qualidade dentro de um tempo computacional razoável. Deseja-se contribuir na modelagem e definição de regiões de interesse e de não interesse ao pouso do VANT. Uma possível contribuição prática é o desenvolvimento de um sistema autônomo de planejamento de rotas que minimize danos em caso de situação crítica e que seja capaz de reduzir o grau de dependência de operadores humanos nas tomadas de decisões.

Este trabalho apresenta também alguns pontos que podem limitar o seu desenvolvimento. O VANT precisa conhecer o cenário de voo, já que qualquer eventual mudança no cenário pode afetar os resultados. Um sistema embarcado que possibilite acesso a tais informações precisa ser estabelecido para a aplicação em cenários reais. Uma segunda limitação está na complexidade dos modelos matemáticos capazes de representar a dinâmica da aeronave no contexto considerado. Devido a alta complexidade de tais modelos, a obtenção de solução ótima em tempo real se torna inviável. Isso limita o uso embarcado de *solvers* como CPLEX que são capazes de garantir uma solução ótima. Por último, outra limitação é a dificuldade inerente à realização de experimentos reais em campo, pois, trata-se de uma tarefa complexa que demanda tempo e recursos financeiros.

1.6 Organização

A organização deste documento foi subdividida da seguinte forma:

- **Capítulo 1:** apresentou a contextualização do problema, trazendo as motivações, justificativas, objetivos, lacunas, contribuições e limitações do presente trabalho;

- **Capítulo 2:** expõe uma revisão bibliográfica dos principais conceitos envolvidos em VANTs, AGs e IFA, além dos principais trabalhos relacionados;
- **Capítulo 3:** define o problema a ser estudado, apresenta a descrição, a modelagem matemática e um conjunto de situações críticas a serem avaliadas;
- **Capítulo 4:** apresenta a metodologia a ser utilizada, detalhando os algoritmos utilizados e a arquitetura a desenvolvida;
- **Capítulo 5:** mostra os resultados obtidos nesta dissertação como: gerador automático de mapas; modelo de dinâmica do VANT e os experimentos das simulações dos métodos propostos em experimentos *offline* e *online*;
- **Capítulo 6:** encerra esta dissertação mostrando as conclusões, contribuições e possíveis linhas para pesquisas futuras.



REVISÃO BIBLIOGRÁFICA

“Se fui capaz de ver mais longe, é porque me apoiei em ombros de gigantes.”

Isaac Newton

2.1 Introdução

Este capítulo apresentará uma revisão dos principais conceitos utilizados neste trabalho como: Veículos Aéreos Não Tripulados (VANTs), Algoritmos Genéticos (AGs) e *In-Flight Awareness* (IFA). Posteriormente, serão apresentados os principais simuladores de voo estudados como: FightGear, X-Plane, Microsoft Flight Simulator. Por fim, os principais trabalhos relacionados com essa dissertação serão revisados.

2.2 Conceitos Importantes

A presente dissertação definirá alguns conceitos relacionados aos VANTs para facilitar o entendimento do objeto estudado. No desenvolvimento dos métodos propostos, conceitos a respeito de AGs e IFA também precisam ser devidamente estabelecidos.

2.2.1 Veículos Aéreos Não Tripulados

A Agência Nacional de Aviação Civil (ANAC) define VANT ou *drone* como sendo “Aeronave projetada para operar sem piloto a bordo e que não seja utilizada para fins meramente recreativos” (ANAC, 2012a, pág. 3). Essas aeronaves podem ser controladas a distância por meios eletrônicos e computacionais, por pessoas e por piloto automático (LEITE, 2013, pág. 4).

Segundo ANAC (2012b, pág. 3), os VANTs necessitam de infraestrutura remota para sua operação por não possuírem piloto a bordo. Essa infraestrutura compõe o que se denomina Sistema de Veículo Aéreo Não Tripulado (SisVANT). O SisVANT é formado pelo veículo aéreo, os componentes necessários à decolagem, voo e pouso do veículo, os meios utilizados na

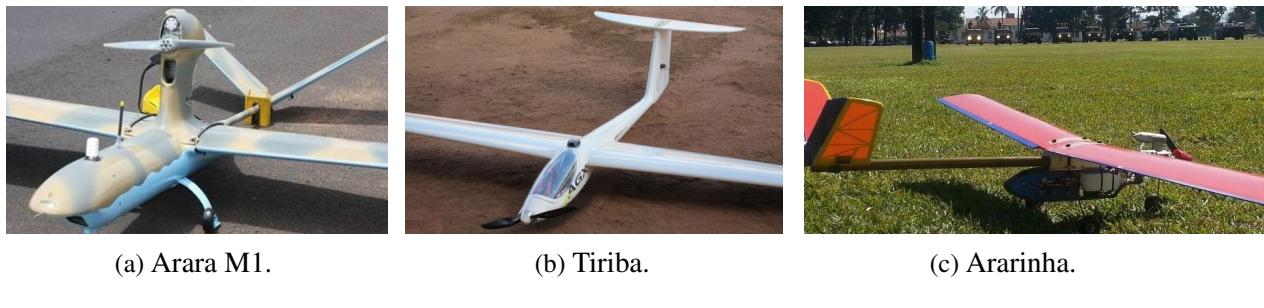
realização da missão, a estação de pilotagem remota, os meios para comunicações e controle, entre outros. Este trabalho se encaixa na categoria de SisVANT, onde serão desenvolvidos componentes que permitam o replanejamento da rota na ocorrência de uma situação crítica.

2.2.1.1 Histórico de VANTs no ICMC

O Instituto de Ciências Matemáticas e Computação da Universidade de São Paulo (ICMC-USP) apresenta um histórico de pesquisas envolvendo VANTs. A seguir são apresentados alguns desses projetos de pesquisa. O projeto Aeronave de Reconhecimento Autônoma e Remotamente Assistida (ARARA) foi desenvolvido numa parceria entre o ICMC-USP e a Empresa Brasileira de Pesquisa Agropecuária (EMBRAPA). Trata-se de um projeto voltado especialmente para utilização do VANT em agricultura de precisão. Tal projeto deu origem, em 2005, ao primeiro *drone* de asa fixa desenvolvido com tecnologia brasileira ([WIKIPÉDIA, 2014c](#)). O Arara M1 é uma aeronave proveniente dessa parceria, utilizada no monitoramento, reconhecimento e vigilância para operações militares ([AGX, 2014](#)). A Figura 2 (a) apresenta o Arara M1.

O projeto da aeronave Tiriba, desenvolvido em 2011 pela empresa AGX em parceria com o ICMC-USP, é mostrado na Figura 2 (b). O resultado desse projeto foi o desenvolvimento do primeiro VANT de propulsão elétrica com tecnologia nacional, voltado para aplicações em mapeamento, monitoramento, levantamento ambiental e controle agrícola ([AGX, 2014](#); [WIKIPÉDIA, 2014c](#)).

Figura 2 – Veículos Aéreos Não Tripulados Arara M1, Tiriba e Ararinha.



Fonte: (a) Foto de [AGX \(2014\)](#). (b) Foto de [AGX \(2014\)](#). (c) Foto de [GISA \(2015\)](#).

Um projeto iniciado em 2013, totalmente nacional, é o Ararinha. Trata-se de uma iniciativa acadêmica, aberta e de baixo custo para o desenvolvimento de sistemas aéreos não tripulados. A Figura 2 (c) mostra essa aeronave. Ela foi desenvolvida pelo Grupo de Interesse em SisVANTs e Aplicações (GISA), localizado no ICMC-USP, cujo objetivo é disseminar conhecimento sobre a área e promover aplicações que envolvam a academia, indústria e iniciativas amadoras. Mais informações sobre o projeto são encontradas em [GISA \(2015\)](#).

A Tabela 1 apresenta as principais características técnicas do Tiriba, que será tomado como base neste trabalho para a modelagem da dinâmica de um VANT diante de uma situação crítica.

Tabela 1 – Detalhes técnicos da aeronave Tiriba.

Atributo	Valor
Decolagem	Manual / Catapulta
Pouso	Automático / Paraquedas
Propulsão	Elétrica ($1,2kW$)
Autonomia	$40min$ a $1h60min$
Envergadura	$2m$
Peso	$3kg$
Carga útil	$0,7kg$
Velocidade de cruzeiro	$86km/h$ ou $24m/s$
Velocidade terminal considerada	$110km/h$ ou $30,5m/s$
Velocidade de sustentação considerada	$40km/h$ ou $11,1m/s$
Aceleração máxima considerada	$2m/s^2$

Fonte: [Branco et al. \(2011\)](#), [Braga et al. \(2011\)](#).

2.2.1.2 Componentes da Aeronave e Sistemas de Coordenadas

A aeronave possui um conjunto de componentes, controles e movimentos associados. A seguir, será ilustrado como eles estão organizados na presente dissertação para um avião de asa fixa. A Figura 3 mostra um avião com suas principais superfícies de controle (*aileron*, leme e profundor) e sistemas de coordenadas (eixo longitudinal, eixo lateral e eixo vertical) com os respectivos movimentos (*roll*, *pitch*, e *yaw*) ([RODRIGUES, 2011](#)).

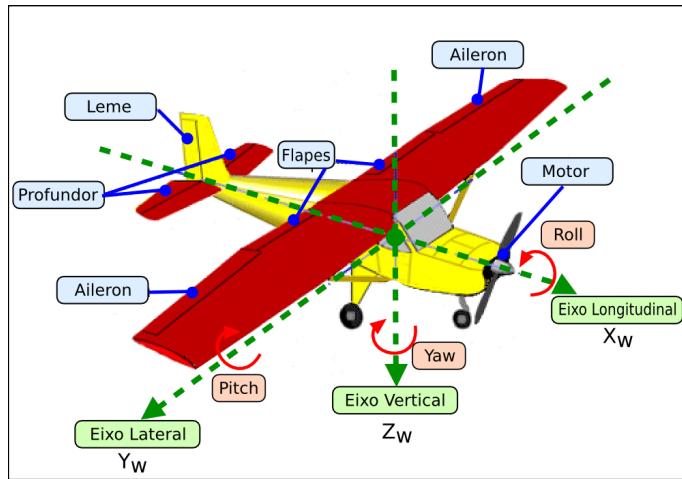
De forma geral, a aeronave possui três eixos principais que são: eixo longitudinal (X_w), eixo lateral (Y_w) e eixo vertical (Z_w). Sobre esses eixos existem três formas de rotacioná-la que são: *roll* ou rolamento que representa a rotação do veículo ao redor do eixo X_w , *pitch* ou arfagem rotação sobre o eixo Y_w e *yaw* ou guinada rotação ao redor do eixo Z_w . Devido a maior popularidade, as terminologias em inglês *roll*, *pitch* e *yaw* serão utilizadas. Os principais elementos de controle da aeronave são: *aileron*, profundor, leme e motor, responsáveis pelas rotações de *roll*, *pitch*, *yaw* e movimento do veículo aéreo. Os flaps também são utilizados no controle, principalmente, no momento da decolagem e do pouso.

Uma descrição de como se relacionam os sistemas de coordenadas da aeronave e do mundo é importante. A Figura 4 mostra esses dois sistemas, no qual as coordenadas do mundo permanecem invariáveis em relação a posição da aeronave. Neste trabalho, é considerado que as equações de movimento do VANT e toda a matemática envolvida estarão relacionada ao sistema de coordenadas do mundo.

2.2.1.3 Situações Críticas em VANTs

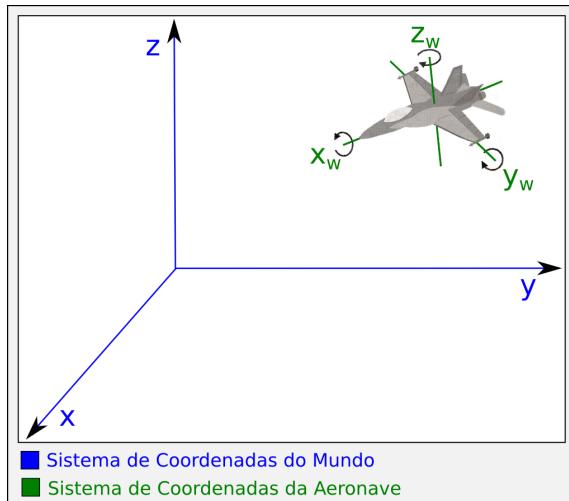
Existem poucas fontes que divulgam as causas de acidentes envolvendo aeronaves não tripuladas. A Tabela 2 apresenta um conjunto de problemas que ocorrem em VANTs organizado pelo Conselho de Investigação de Acidentes (AIB) da Força Aérea dos Estados Unidos (USAF)

Figura 3 – Eixos da aeronave e superfícies de controle da aeronave.



Fonte: Adaptada de Rodrigues (2011).

Figura 4 – Sistemas de coordenadas da aeronave e do mundo.



Fonte: Elaborada pelo autor.

(DRONEWARS, 2016).

As falhas mencionadas nessa tabela servirão de base para os estudos dessa dissertação, onde as situações críticas consideradas serão detalhadas no Capítulo 3. A Tabela 2 apresenta as falhas levantadas na pesquisa. Dada a ausência do piloto, a falha com Id:3 não ocorreria no sistema proposto neste trabalho. A falha Id:8 é complexa e restrita ao ambiente militar, dessa forma, não será tratada neste estudo. As situações id:4, 6, 10, 13 e 14 também não serão consideradas por serem específicas e de difícil modelagem. As falhas consideradas neste trabalho são: id:1 e 12 tratadas aqui como falhas no motor; id:9 e 11 serão assumidas como problemas na bateria (combustível); id:2, 5 e 7 serão falhas que impedem o VANT, por exemplo, de virar para a esquerda ou para a direita. Esses dois últimos tipos de problemas ilustram casos em que a aeronave não consegue ativar o leme e/ou aileron, responsáveis pelas curvas à esquerda e à

Tabela 2 – Detalhes de acidentes envolvendo VANT militares.

Id	Identificação da falha	Ano de ocorrência (Quantidade)									
		2007	2008	2009	2010	2011	2012	2013	2014	2015	Total
1	Falha do motor	6	2	1		3	6	1	1		20
2	Falha mecânica		3	6	1	1	3	2	3		19
3	Erro do piloto		2	1	7	1	3		2		16
4	Perda de comunicação	2	3	2	2		3	3			15
5	Falha elétrica		2	2			1		1		6
6	Condições meteorológicas				1	2		2			5
7	Falha eletrônica							1		2	3
8	VANT abatido									2	2
9	Vazamento de combustível/óleo			1					1		2
10	Bater na pista de decolagem						2				2
11	Falha no sistema de alimentação	1									1
12	Problemas nas hélices							1			1
13	Falha mecânica e meteorológica								1		1
14	Asa esquerda caiu	1									1
15	Falha no controle da asa direita								1		1
16	Desconhecida	5	14	14	11	23	18	19	8	16	128
	Total de acidentes	15	26	27	22	30	36	29	18	20	223

Fonte: Adaptada de [DroneWars \(2016\)](#).

direita.

Na Tabela 2, há um total de 95 falhas conhecidas de um total de 223. As situações críticas consideradas nesta dissertação correspondem a 53 ocorrências de um total de 95. Dessa maneira, caso os métodos propostos aqui consigam abordar todas as situações críticas mencionadas, um tratamento satisfatório será dado a 55,7% das avarias conhecidas, representando 23,7% do total.

2.2.2 In-Flight Awareness

O conceito chamado *In-Flight Awareness* (IFA) está relacionado à ideia de Consciência Situacional (CS) em aeronaves não tripuladas, de acordo com [Mattei \(2015\)](#). O IFA é um sistema supervisor que monitora todos os aspectos relacionados à segurança da aeronave ([MATTEI, 2015](#)). Esse monitoramento é feito através de um conjunto de sensores que verificam o funcionamento dos principais componentes da aeronave. O sensoriamento realizado permite processar as informações, visando mitigar acidentes ou simplesmente atualizar o plano de voo, mediante um algoritmo de planejamento de rota. Os autores em [Mattei et al. \(2013\)](#) apresentam uma taxonomia de fatores internos (como o estado de funcionamento da aeronave obtido através de sensores *onboard*) e externos (como condições meteorológicas obtidas via enlace de comunicação com a estação de base) ao VANT. Em caso de condições não favoráveis ao voo, uma estratégia baseada em árvore de decisão é utilizada para atualizar o plano de voo. Uma proposta de implementação do sistema IFA é descrita em [Mattei et al. \(2015\)](#), chamada *In-Flight Awareness Augmentation Systems* (IFA²S), apresentando como seria sua integração a uma aeronave do mundo real. Nesse mesmo trabalho, um estudo de caso foi conduzido considerando uma simulação que integra o Labview (linguagem de programação baseada em fluxo de dados) ao simulador de voo X-Plane.

A definição original para o termo Consciência Situacional (CS) foi dada por [Endsley](#)

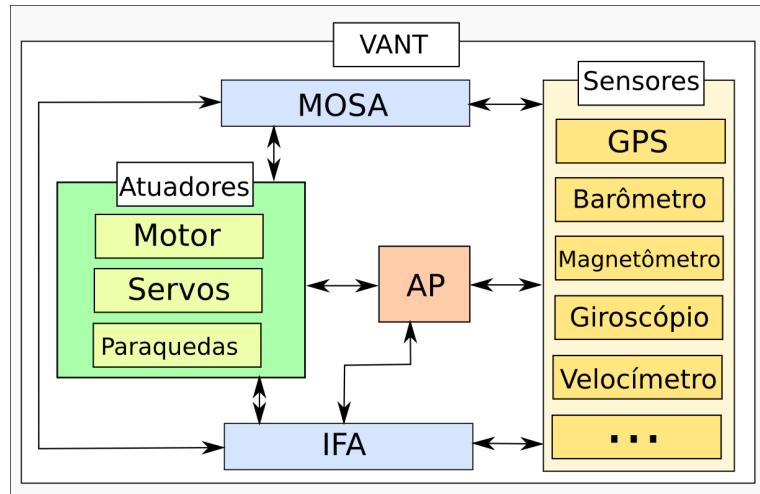
(1988): “a percepção de elementos no ambiente dentro de um volume de tempo e de espaço, a compreensão de seu significado e a projeção de seu status no futuro próximo”. O trabalho de Drury, Riek e Rackliffe (2006) usa a definição de Endsley (1988), mas a altera de forma a tornar mais fácil o reconhecimento da presença de grupos de pessoas e de VANTs durante uma operação. O trabalho de Drury, Riek e Rackliffe (2006) também analisou alguns dos fatores que afetam um voo e, consequentemente, possíveis de serem considerados como insumo para IFA. Elementos como o estado dos sistemas da aeronave, condições climáticas, o estado da carga útil, o conhecimento do operador sobre as capacidades da plataforma e as dinâmicas foram mencionados para serem considerados durante a decomposição do problema. A percepção dos elementos mencionados é fundamental para que a aeronave possa cumprir sua missão.

Em um trabalho posterior, os autores em Endsley e Jones (2006) apresentaram outra definição para CS: “entendimento compartilhado do subconjunto de informações necessário para cada um de seus objetivos”. Essa definição para CS é utilizada em Seppänen *et al.* (2013.) sob a ótica da gestão de desastres, onde se pode encontrar uma situação particular na qual a coordenação, a informação compartilhada e a resposta rápida são figuras-chave para a eficiência. Aqui CS é um conceito importante para o cumprimento da missão e foi desenvolvido para ser utilizado como uma ferramenta para aumentar a consciência da situação dos VANT em ação. Para situações emergenciais, as informações necessárias devem estar disponíveis no momento certo e no formato adequado.

Os sistemas para evitar acidentes envolvem radares *Air Traffic Control* (ATC), sistemas *Sense and Avoid* (S&A) e os pilotos a bordo como uma medida final. Levando em consideração que os VANTs são uma plataforma robótica, projetada para transportar cargas úteis com o objetivo de realizar missões em um ambiente em constante mudança, é fundamental estabelecer formas de melhorar a consciência situacional para além da percepção de possíveis colisões. Considerando também que segurança é um fator importante para viabilizar a integração dos VANTs ao espaço aéreo, é preciso criar formas de trazer novamente a bordo o que foi eventualmente perdido com a saída dos pilotos.

O IFA está integrado a aeronave como um componente dela, da mesma forma que o Piloto Automático (AP) e o *Mission Oriented Sensor Array* (MOSA). A Figura 5 mostra essa relação. A principal tarefa do MOSA é o comprimento da missão, ele possui acesso a um conjunto de sensores do veículo aéreo (FIGUEIRA *et al.*, 2013). Uma outra tarefa relacionada à missão é o controle da carga útil e também a trajetória de voo. O VANT navega usando um AP para manter um plano de voo pré-definido para realizar a missão. Todas as responsabilidades ligadas às questões de segurança são destinadas ao IFA, assim, ele pode utilizar as informações provindas de uma série de sensores da aeronave para garantir a segurança. Esta dissertação visa integrar ao IFA um planejador de rotas que leve o VANT até uma região considerada segura após a ocorrência de uma situação crítica.

Figura 5 – Esquema da integração entre sensores, atuadores, IFA, AP e MOSA na aeronave.



Fonte: Elaborada pelo autor.

2.3 Simuladores de Voo

Simulação é a imitação da operação de um processo do mundo real ou do sistema ao longo do tempo Banks (2000). Um simulador de voo é um software que tenta recriar a realidade existente no voo de uma aeronave em função do tempo de voo. Conseguir simular o mundo da maneira mais realística possível é uma importante tarefa a ser realizada por um simulador de voo. Pensando nisso, muitos simuladores de voo incluem turbulências, mudança do clima, transição do tempo, entre vários outros fatores externos.

Esta seção apresenta uma breve revisão sobre algumas características de três principais simuladores de voo encontrados: FlightGear, X-Plane e Microsoft Flight Simulator.

2.3.1 *FlightGear*

FlightGear (FG) é um software de simulação de voo multiplataforma, utilizado em pesquisa acadêmica, educação e para entretenimento. Trata-se de uma ferramenta computacional lançada de forma livre e com código aberto, sob a licença GPL (FLIGHTGEAR, 2014b). A Figura 6 apresenta uma tela do simulador. O FlightGear se caracteriza pelo seu realismo e por possuir uma grande variedade de aeronaves, aeroportos e cenários disponíveis para *download*. O desenvolvimento desse simulador tem como foco principal o realismo e a possibilidade do seu uso para pesquisas, não sendo voltado para jogos. A ferramenta exige um conhecimento mínimo de pilotagem e lidar com seus controles pode gerar uma dificuldade inicial (WIKIPÉDIA, 2014a).

O motor de simulação SimGear é utilizado pelo FG, sendo um software para simulações independentes do FG (FLIGHTGEAR, 2014d). Também é utilizado o TerraGear, que é um programa de dados de terrenos capaz de produzir efeitos climáticos que incluem nuvens 3D, efeitos de iluminação e hora do dia (FLIGHTGEAR, 2014e). Esse simulador utiliza a biblioteca

Figura 6 – Simulador de voo FlightGear.



Fonte: [FlightGear \(2014a\)](#).

OpenGL para desenhar rotinas gráficas 3D ([FLIGHTGEAR, 2014c](#)).

2.3.2 X-Plane

O simulador X-Plane, entre todos os simuladores de voo, é o único simulador para computador credenciado pela Administração Federal de Aviação (FAA) para treinamento de pilotos ([X-PLANE, 2014a](#)). Esse simulador possui uma alta precisão no modelo de voo, uma grande base de dados de aeronaves e terrenos, cenários de navegação de todo o mundo ([ASTUTI et al., 2007](#)). A Figura 7 mostra a tela de interface do simulador X-Plane.

Figura 7 – Simulador de voo X-Plane.



Fonte: [X-Plane \(2014c\)](#).

X-Plane conta com um editor de mapas chamado World Editor que é capaz de criar cenários e aeroportos 2D ([X-PLANE, 2014b](#)). Um outro recurso importante deste simulador é o Plane Maker que permite aos usuários personalizar ou criar aeronaves ([X-PLANE, 2013](#)).

2.3.3 Microsoft Flight Simulator

Microsoft Flight Simulator (MSFS) é um jogo de computador e também um simulador de voo ([MICROSOFT, 2014](#)). Na Figura 8, é apresentada uma tela do simulador de voo Flight Simulator. Ele é distribuído na forma *Free-to-Play*, ou seja, o jogo é gratuito, porém o usuário paga por conteúdo extra, como aeronaves ou paisagens. A partir de 2010 o simulador mudou de

nome para Microsoft Flight e permaneceu em desenvolvimento até 2012, quando foi cancelado ([WIKIPÉDIA, 2014b](#)).

Figura 8 – Simulador de voo Microsoft Flight Simulator.



Fonte: [Microsoft \(2014\)](#).

2.3.4 Comparação dos Simuladores

A Tabela 3 traz um resumo de algumas características dos simuladores de voos estudados. Entre essas características destacam: o ano de lançamento, o ano da última versão, a portabilidade para diferentes sistemas operacionais, o modo de licença, a distribuição e qual a linguagem de programação em que foi desenvolvido.

Tabela 3 – Comparações entre os simuladores de voo: FightGear, X-Plane e Microsoft Flight Simulator.

Simulador	Lançamento	Última Versão	Sistema Operacional	Licença	Distribuição	Linguagem
FG	1997	2015	Windows, Linux, Mac OS, FreeBSD, Solaris, IRIX	GPL	Grátis	C/C++
X-Plane	1993	2015	Windows, Linux, Mac OS	Proprietário	Pago	C/C++
MSFS	1982	2012	Windows	Proprietário	Free-to-Play	C/C++/C#

Fonte: Elaborada pelo autor.

Nesta dissertação, dois tipos de experimentos serão efetuados: experimentos *offline* e *online*. Chama-se aqui experimentos *offline*, as simulações onde os algoritmos responsáveis pelo pouso da aeronave não estão sendo executados juntamente com o simulador de voo. Já experimentos *online* ocorrem quando o algoritmo planejador de rotas está sendo executado junto a um simulador de voo. No segundo caso, toda a dinâmica do veículo aéreo e do ambiente herdados do simulador de voo são considerados, tornando a simulação mais realista. Utilizou-se o simulador FlightGear na etapa de experimentos *online*, pois ele suporta múltiplas plataformas, sua distribuição é gratuita e se trata de um sistema em constante atualização.

2.4 Trabalhos Relacionados

O planejamento de missão para veículos autônomos foi abordado em [Li \(2010\)](#), [Blackmore, Ono e Williams \(2011\)](#), [Ono, Williams e Blackmore \(2013\)](#), onde foi modelado como um problema de Programação Inteira Mista (PIM). Em [Li \(2010\)](#), é proposto um sistema capaz de planejar uma sequência de ações discretas e controles contínuos, aplicados em veículos autônomos aéreos e submarinos. O sistema planejador fornece ao veículo um nível de autonomia que permite a tomada de decisão ótima durante o cumprimento de sua missão. Esse trabalho apresenta duas inovações. Primeiro, utiliza uma representação compacta para todos os planos factíveis, sendo o primeiro trabalho a tratar o planejamento para ações contínuas e discretas em conjunto. Segundo, apresenta uma formulação para definir em alto nível o conjunto das ações que o VANT pode tomar e o *scheduling* dos objetivos da missão. Entretanto, o sistema proposto não considerava a alocação de risco na tomada de decisão.

Os autores em [Blackmore, Ono e Williams \(2011\)](#) utilizam alocação de risco durante o planejamento de rotas para VANTs, onde o risco de colisão com obstáculos deve estar dentro de uma margem de segurança. O problema de planejamento de rotas é descrito usando modelos estocásticos. O problema se torna não convexo ao incorporar o desvio de obstáculos, assim, os autores propuseram relaxações e aproximações para resolvê-lo. No entanto, o planejamento de missão não é considerado pelos autores, eles tratam apenas o planejamento de caminho de um ponto origem até um destino com desvio de obstáculos e alocação de riscos.

O planejamento de missão é estudado em [Ono, Williams e Blackmore \(2013\)](#), onde um novo sistema planejador é proposto e melhorias no tratamento de alocação de risco em relação a [Blackmore, Ono e Williams \(2011\)](#) são introduzidas. O sistema incorpora *scheduling* para as tarefas a serem executadas, semelhantemente ao trabalho de [Li \(2010\)](#), planejando o tempo em que o veículo deverá atingir uma série de objetivos da missão. Assim, a missão pode ser descrita como um conjunto de episódios que ocorrem entre dois eventos. Os episódios especificam objetivos a serem cumpridos pelo VANT e cada episódio tem um risco associado. Esse trabalho foi validado em dois cenários: Sistema de Transporte Pessoal e Sistema de Transporte de Carga Aeroespacial. O algoritmo utilizado pelos autores é baseado na técnica *Branch and Bound* (B&B).

O replanejamento de rota a partir de uma situação crítica é tratado em [Meuleau et al. \(2009\)](#), onde um sistema ajuda os pilotos de aviões tripulados a determinar o melhor local de pouso de aeronaves avariadas. Após a falha no avião, o piloto deve primeiro recuperar o controle da aeronave e, em seguida, o objetivo é encontrar o melhor local para o pouso de emergência. Vários fatores são avaliados para determinar o local de queda como: condições atuais de controle da aeronave (recursos disponíveis), distância para o local de pouso, condições meteorológicas da rota, características da trajetória de aproximação, características do local de pouso, equipamentos de emergência disponíveis no local e perigo à população ao longo da trajetória de pouso. O sistema proposto, chamado de Planejador de Pouso de Emergência (PPE), avalia esses fatores e

propõe rotas, bem como possíveis locais de pouso para a aeronave. O PPE utiliza um algoritmo A* híbrido.

Em um trabalho posterior, os autores em [Meuleau et al. \(2011\)](#) analisam o comportamento do algoritmo A* proposto em [Meuleau et al. \(2009\)](#) através da sua aplicação em um cenário real. Os autores realizaram testes em um simulador de voo para aeronave de grande porte, onde o A* híbrido é utilizado para determinar o local de pouso. A heurística de estimativa utilizada para guiar o A* combina a distância euclidiana de cada pista de pouso possível e os riscos associados à aproximação e aterrissagem naquela pista. O PPE projetado considera apenas aeroportos e pistas oficialmente reconhecidos. A simulação considera panes que ocorrem entre 1 a 3 minutos de voo, onde o algoritmo PPE é acionado para auxiliar os pilotos. O algoritmo exibe as possíveis rotas de pouso de emergência da aeronave para o piloto da aeronave.

Existem poucos trabalhos enfatizando o pouso de aeronaves em situação crítica como reportado em [Meuleau et al. \(2011\)](#), [Meuleau et al. \(2009\)](#). Nos dois primeiros trabalhos, conforme descrito, considera-se o pouso emergencial de aeronaves tripuladas de grande porte. Trabalhos tratando o pouso emergencial de VANTs são também escassos como reportado em [Li, Chen e Li \(2014\)](#), [Li \(2013\)](#). Nesses trabalhos, as falhas não são bem definidas e não são apresentados experimentos rigorosos considerando tais falhas.

Há uma grande quantidade de trabalhos abordando o pouso de VANTs como [Hermansson et al. \(2010\)](#), [Kim et al. \(2013\)](#), [Li, Chen e Li \(2014\)](#), [Li \(2013\)](#), [Saripalli, Montgomery e Sukhatme \(2002\)](#), [Zhao e Pei \(2012\)](#). A grande maioria trata o pouso autônomo de VANTs com asa rotativa [Hermansson et al. \(2010\)](#), [Li \(2013\)](#), [Saripalli, Montgomery e Sukhatme \(2002\)](#), [Zhao e Pei \(2012\)](#). Efetuar o pouso de veículo aéreo de asa rotativa tem como vantagem a baixa velocidade no momento do pouso devido a sua capacidade de flutuar. Os autores em [Hermansson et al. \(2010\)](#), [Kim et al. \(2013\)](#), [Saripalli, Montgomery e Sukhatme \(2002\)](#), [Zhao e Pei \(2012\)](#) efetuam o pouso considerando estações de pouso estacionárias com cenários de testes menores que $20m \times 20m$. A presente dissertação lida com cenários de $1000m \times 1000m$.

Algoritmos evolutivos tem sido utilizados no planejamento de rotas para VANTs. O autor [Pehlivanoglu \(2012\)](#) utilizou AG com diagrama de Voronoi para planejamento de rotas de voos autônomos. Nesse trabalho, é dada ênfase na nova estratégia de mutação criada, que foi separada em diversidade aleatória global e local. O diagrama de Voronoi é utilizado na fase inicial do processo de geração da população. A técnica proposta levou a melhoria na qualidade dos indivíduos da população o que acelerou a convergência do método.

No trabalho de [Tuncer e Yildirim \(2012\)](#) foi utilizado AG no planejamento de rotas para Veículos Terrestres Não-Tripulados (VTNT). Um novo operador de mutação, aplicado ao planejamento de caminho em ambientes dinâmicos e com obstáculos, é proposto. Esse operador evita a convergência prematura do método e consegue encontrar o caminho ideal várias vezes. O ambiente a ser percorrido no trabalho é representado por uma malha (*grid*).

Métricas para comparar planejadores de caminho para VANTs foram proposta em [Besada-Portas et al. \(2013\)](#), onde são levadas em consideração a complexidade e peculiaridades do problema tratado. As métricas criadas analisam o desempenho dos métodos de forma gráfica. Os autores avaliaram o desempenho de três técnicas: Algoritmo Genético (AG), Otimização por Enxame de Partículas (OEP) e Evolução Diferencial (ED). Os resultados indicaram que o AG foi o melhor método, seguido pelo OEP e ED respectivamente.

ED é aplicada em [Zhang e Duan \(2015\)](#) no planejamento de rota para VANT em ambientes tridimensionais. A rota é projetada para ter curta duração e baixa altitude de voo. A aeronave deve evitar regiões não navegáveis, regiões com radares, mísseis e armas antiaéreas. Os resultados foram obtidos a partir de dois diferentes cenários aplicados a ambientes de guerra e se mostrou, em geral, superior aos onze algoritmos da literatura comparados em termos de robustez e velocidade de convergência.

Os autores em [Varela et al. \(2014\)](#) também utilizaram ED para cenários de busca e resgate em tempo real com VANTs. Esse trabalho lida com um conjunto de aeronaves autônomas utilizado para tratar a tarefa de resgate em uma grande área. Os resultados são comparados com uma estratégia baseada em Inteligência de Enxame (IE). A abordagem evolutiva mostrou-se melhor que a IE, pois, a IE diminui o seu desempenho com o aumento do número de alvos de resgate, o que não ocorre com a ED.

No trabalho de [Nikolos, Tsourveloudis e Valavanis \(2007\)](#), o planejamento de rotas para múltiplos VANTs é baseado em ED. As rotas planejadas são representadas por curvas 3D usando a técnica B-Spline. O cenário é representado por uma malha tridimensional. Dois casos de testes são validados: *offline*, onde rotas são geradas em ambiente conhecido e com restrições de voo conhecidas, e *online*, onde as rotas rotas são criadas em ambiente estáticos e desconhecidos, usando as informações dos sensores da aeronave.

O problema de pouso emergencial, tratado nessa dissertação, poderia ser resolvido por uma grande quantidade de métodos, heurísticas e metaheurísticas, além das descritas acima como: Multi-Start, Simulated Annealing, Busca Tabu, GRASP, Algoritmos Meméticos, Algoritmo de Estimação de Distribuição (AED), Campos Potenciais, A*, Dijkstra, Otimização por Colônia de Formigas (ACO) e Otimização por Enxame de Partículas (OEP). Algumas dessas técnicas, necessitam de uma maior ou menor adaptação do problema ao método, por exemplo, o algoritmo A* necessita de uma discretização do espaço de busca já o algoritmo de Dijkstra não leva em conta a dinâmica associada ao veículo de asa fixa.

O presente trabalho propõe o uso de Algoritmo Genético Multi-Populacional (AGMP), onde indivíduos são hierarquicamente estruturados em árvores. Essa abordagem foi introduzida por [Franca, Mendes e Moscato \(2001\)](#) com indivíduos seguindo uma hierarquia baseada no seu valor de *fitness* em uma árvore ternária. Tal abordagem tem sido aplicada para resolver problemas em diferentes contextos como programação da produção na indústria de embalagem de vidro ([Toledo et al. \(2013\)](#)), células de manufatura ([Tavakkoli-Moghaddam, Gholipour-Kanani e](#)

[Cheraghalizadeh \(2008\)](#)), problema do caixeiro viajante ([Buriol, Franca e Moscato \(2004\)](#)), ordenação de dados ([Moscato, Mendes e Berretta \(2007\)](#)), entre outros, com resultados relevantes alcançados.

Os estudos conduzidos em [Li \(2010\)](#), [Blackmore, Ono e Williams \(2011\)](#), [Ono, Williams e Blackmore \(2013\)](#) tiveram avanços no tratamento de risco para planejamento e garantem que a missão planejada seja executada dentro de uma margem de segurança, considerando as imprecisões dos equipamentos e do ambiente. Entretanto, não é considerado o aparecimento de situações críticas. O planejamento assume que os equipamentos tenham imprecisões conhecidas e, por isso, os trabalhos tratam a alocação de risco. Não é considerado que possa haver uma falha total em um equipamento.

Além das heurísticas e metaheurísticas propostas, esta dissertação apresenta uma formulação matemática para o problema de replanejamento de caminho baseada no modelo descrito em [Blackmore, Ono e Williams \(2011\)](#) para planejamento de caminho com alocação de risco. Porém, a presente proposta planeja uma rota de pouso válida, dada uma falha e um ponto origem. Quando se considera falhas em aeronaves, o presente estudo se assemelha ao descrito em [Meuleau et al. \(2009\)](#). Todavia, difere por lidar com VANTs, possuir um conjunto diferente de falhas associadas e por apresentar falhas severas que levam a queda da aeronave de forma imediata. Em [Meuleau et al. \(2009\)](#), a aeronave era capaz de executar o voo mesmo com a falha. Em relação às técnicas aplicadas na solução do problema, este trabalho se assemelha com [Toledo, Lima e Arantes \(2012\)](#), [Toledo et al. \(2013\)](#) ao aplicar técnicas de computação evolutiva.

2.5 Conclusão

Este capítulo apresentou os principais conceitos envolvidos com VANTs; AGs e IFA. A relação entre IFA, AP e MOSA com os sensores e atuadores da aeronave foi estabelecida. Uma breve descrição de alguns simuladores de voo existentes também foi apresentada. Todos esses conceitos são importantes e dão suporte para a compreensão dos capítulos seguintes.

Ao fim desse capítulo, foram descritos trabalhos relacionados a presente dissertação. Esses trabalhos serviram de base para o desenvolvimento da modelagem matemática e dos métodos para a solução do problema de pouso emergencial abordado.



PROBLEMA ABORDADO

*“Para quê preocuparmo-nos com a morte?
A vida tem tantos problemas que temos de
resolver primeiro.”*

Confúcio

3.1 Introdução

O problema abordado será descrito inicialmente em linhas gerais e ilustrado através de um exemplo. Em seguida, uma formulação matemática é proposta para estabelecer os objetivos e restrições considerados neste trabalho.

3.2 Descrição do Problema

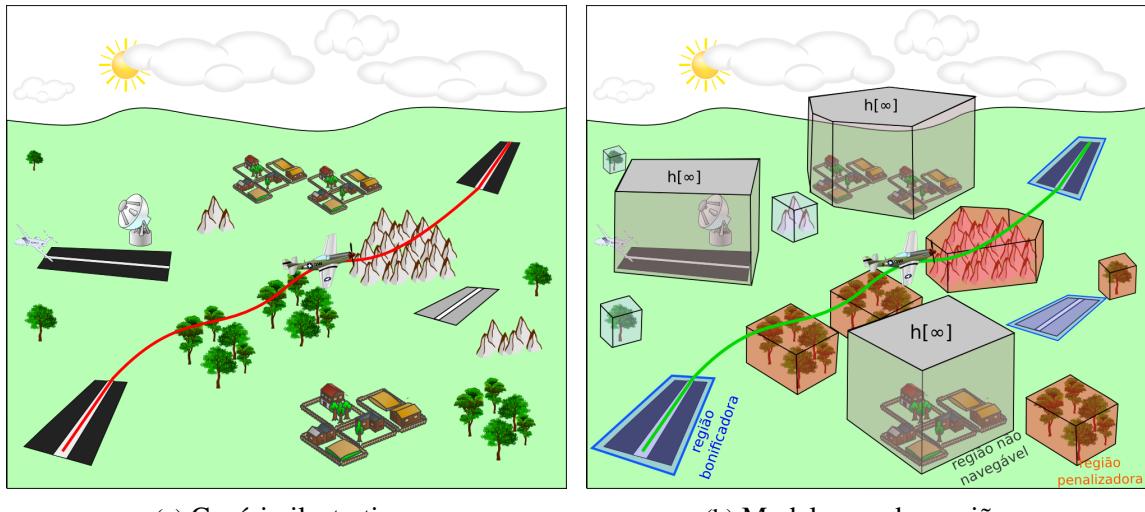
As características do problema abordado serão apresentadas a partir de um exemplo descrito a seguir e ilustrado nas Figuras 9 e 10.

Missão: *O Veículo Aéreo Não Tripulado (VANT) inicia seu voo na pista A para executar tarefas em uma missão previamente definida. O mesmo deve sobrevoar as florestas, desviando das casas e do aeroporto. O destino de pouso é a pista B. Ele deve permanecer dentro da região segura ao voo em toda a trajetória.*

Planejamento de pouso emergencial: *Durante a execução da missão, a aeronave apresenta problemas que são detectados pelo sistema supervisor In-Flight Awareness (IFA). O sistema supervisor decide abortar a trajetória atual e executa um replanejamento da rota visando um pouso emergencial. A nova rota será obtida através da minimização de possíveis danos a pessoas, propriedades e a própria aeronave. Nesse exemplo, a nova rota leva o VANT a pousar na pista C.*

A Figura 9 (a) apresenta as áreas dentro do cenário (mapa) considerado: duas áreas povoadas, florestas, árvores, montanhas, um aeroporto para aeronaves tripuladas e três pistas para VANTs. A Figura 9 (b) ilustra um possível tratamento das regiões como obstáculos, representados

Figura 9 – Cenário geral do problema abordado.



Fonte: Elaborada pelo autor.

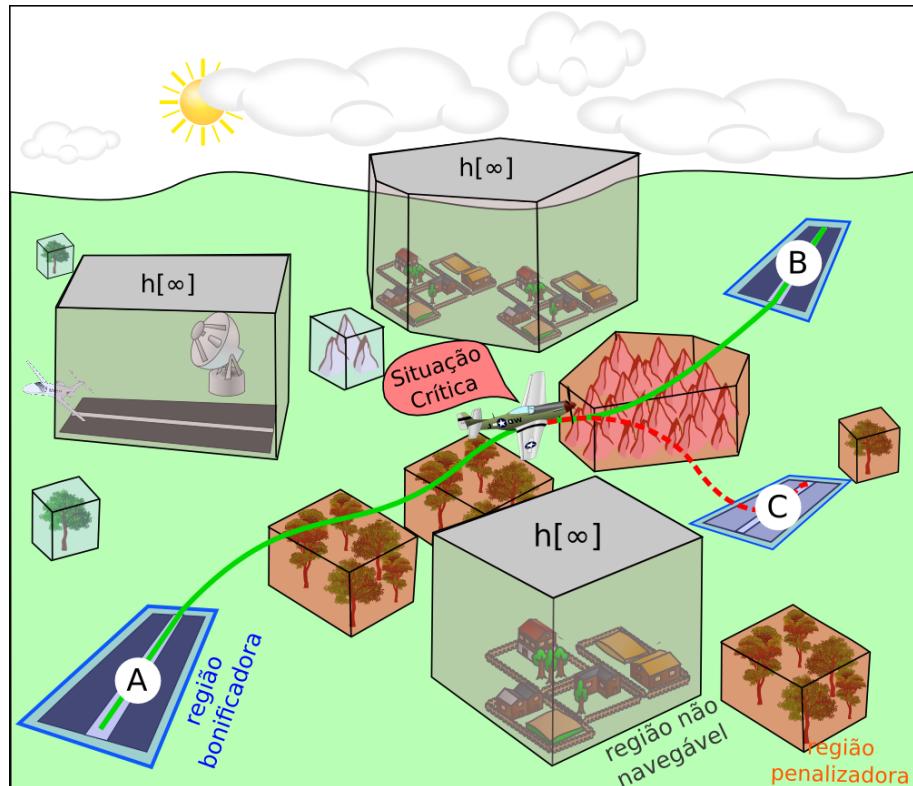
por poliedros no espaço tridimensional (polígonos no espaço bidimensional). Por exemplo, O VANT não pode sobrevoar regiões povoadas e aeroportos. Logo, os respectivos poliedros na Figura 9 (b) tem uma altura infinita ($h[\infty]$), representando assim a impossibilidade de voo. Florestas e montanhas podem ser sobrevoadas, logo, o poliedro assume a altura da maior árvore ou do maior pico naquelas regiões. A Figura 10 apresenta uma possível rota replanejada, onde o VANT é forçado a pouso na região C.

3.2.1 Conjunto de Regiões Modeladas

Várias regiões podem ser mapeadas previamente para a execução da missão. Isso permite que sejam separadas em conjuntos de acordo com suas características. Esses conjuntos, representados por Φ , foram estabelecidos e definidos de acordo com a probabilidade desejada de pouso da aeronave nas regiões. Um total de quatro conjuntos foram definidos neste trabalho.

1. **Conjunto Não Navegável (Φ^n):** A aeronave não pode sobrevoar e pousar nas regiões deste conjunto. As regiões dentro deste conjunto podem ser representadas por aeroportos, bases militares, região povoadas e outras áreas com restrições ao voo. A penalização por sobrevoar essas regiões é altíssima.
2. **Conjunto Navegável com Penalização (Φ^p):** O VANT pode sobrevoar regiões deste conjunto, mas não é desejável que o mesmo pouse sobre elas. As regiões deste conjunto podem representar florestas, montanhas, lagos, entre outras, onde o voo de aeronaves é permitido, mas o seu pouso deve ser evitado. As consequências devido à queda nessas regiões são relevantes, podendo acarretar prejuízos financeiros ou acidentes envolvendo

Figura 10 – Cenário para o planejamento de missão com replanejamento de rota em caso de situação crítica.



Fonte: Elaborada pelo autor.

pessoas. Por isso, haverá uma penalização considerável no caso de pouso nessas regiões. As regiões desse conjunto serão ao longo do texto chamadas de região penalizadora.

3. **Conjunto Navegável e Bonificador (Φ^b):** O VANT pode sobrevoar e é desejado que ele pouse numa das regiões deste conjunto. As regiões são planas e adequadas ao pouso como campos gramados ou áreas com plantações rasteiras. A pista de origem e destino da aeronave também pertencem a este conjunto. O pouso em uma dessas regiões representa a grande chance do veículo aéreo ser salvo. Neste caso, há uma bonificação ao invés de penalização. As regiões desse conjunto serão ao longo do texto chamadas de região bonificadora.
4. **Conjunto Remanescente (Φ^r):** A aeronave pode sobrevoar e pousar nessa região. Este conjunto representa as áreas restantes que não foram classificadas para o pouso. Não há restrições ao voo ou pouso nessas regiões, mas não há informação suficiente para classificá-los em um dos conjuntos Φ^p e Φ^b .

Os três primeiros tipos de regiões dos conjuntos definidos acima (Φ^n , Φ^p e Φ^b) foram modelados através de polígonos convexos. A cada uma dessas regiões foi atribuído um peso associado à penalização/bonificação do pouso forçado do VANT em tais lugares. A região Φ^r

descreve um espaço não convexo. Neste trabalho, a região onde a aeronave pousou será denotada por Φ^* com $\Phi^* \in \{\Phi^n, \Phi^p, \Phi^b, \Phi^r\}$.

3.2.2 Situações Críticas Modeladas

Algumas situações críticas à operação da aeronave foram definidas com base no levantamento de dados descrito na Seção 2.2.1.3. O algoritmo de replanejamento deve ser informado sobre o tipo de limitação imposta à aeronave pelo problema detectado. Neste trabalho, algumas situações críticas foram consideradas, representadas por Ψ , e são descritas a seguir:

1. **Problema no Motor (Ψ^m):** O motor m apresenta problemas e para de funcionar. Neste caso, uma região adequada para o pouso tem que ser encontrada enquanto a aeronave paira. Foi assumido uma perda total do motor, ou seja, não é possível acelerá-la. Logo, o VANT deve pousar rapidamente.
2. **Problema na Bateria (Ψ^b):** A bateria b apresenta um superaquecimento. Todos os controles continuam funcionando, mas o VANT deve pousar o mais rápido possível. Caso o pouso demore, a bateria pode incendiar ou os componentes eletrônicos podem parar, levando a aeronave a uma queda imediata. Um outro tipo de problema associado a bateria é a perda de potência, sendo assim, o VANT deve pousar rapidamente.
3. **Problema nas Superfícies Aerodinâmicas tipo 1 (Ψ^{s1}):** A aeronave apresenta uma falha em uma semi-asa $s1$ que a torna capaz de virar apenas para a esquerda. A principal causa geralmente está associada a danos nas superfícies de controle *aileron*, leme ou respectivos atuadores. Essa falha é caracterizada pela perda do controle látero-direcional com incapacidade de apontamento e realização de curvas para direita.
4. **Problema nas Superfícies Aerodinâmicas tipo 2 (Ψ^{s2}):** A aeronave apresenta uma falha em uma semi-asa $s2$ que a torna capaz de virar apenas para a direita. A principal causa disto geralmente está associada a danos nas superfícies de controle *aileron*, leme ou respectivos atuadores. Essa falha é caracterizada pela perda do controle látero-direcional com incapacidade de apontamento e realização de curvas para esquerda.
5. **Nenhum Problema (Ψ^\emptyset):** A aeronave não apresenta nenhuma falha, logo, todos os controles funcionam. Dessa maneira, o problema se transforma em uma simples rota de pouso sem nenhuma avaria.

A situação crítica a que foi submetida a aeronave, durante o voo, será representada por Ψ^* , onde $\Psi^* \in \{\Psi^m, \Psi^b, \Psi^{s1}, \Psi^{s2}, \Psi^\emptyset\}$.

3.2.3 Considerações Sobre as Condições de Voo e Ambiente

Vamos assumir um cenário no qual o VANT inicia seu voo e ganha certa altitude a uma taxa constante, passando a executar a trajetória em determinada velocidade de cruzeiro. As situações críticas, por hipótese, irão ocorrer somente quando o voo está em cruzeiro e serão consideradas apenas aquelas descritas na Seção 3.2.2.

Duas modelagem diferentes são apresentadas: uma modelagem restrita e outra completa. A modelagem restrita efetua apenas o planejamento horizontal e assume uma taxa constante de descida para o VANT. A modelagem completa efetua todo o planejamento da aeronave num ambiente tridimensional. Após o VANT atingir uma altitude ideal ao final da rota de pouso, será considerado que o paraquedas foi aberto e seu deslocamento horizontal será nulo. Também pode ser assumido que, ao final do último *waypoint* da rota de pouso, um algoritmo assume o controle e aterrissa a aeronave.

Algumas outras condições assumidas neste trabalho são: a velocidade do vento (ar) é igual a zero; a densidade do ar não muda com a altitude; a constante de resistência do ar não muda com o ângulo da aeronave; todos os cenários (mapas) são carregados na memória do VANT; os mapas são todos estáticos, ou seja, não mudam com o tempo.

Os algoritmos envolvidos na etapa de decolagem não são tratados neste trabalho. A missão envolvida no momento do voo cruzeiro também não é tratada, ou seja, o problema de pouso emergencial é independente de missão. A presente dissertação possui como foco o tratamento do pouso a partir do momento em que houve alguma falha na aeronave. Todas as hipóteses consideradas tem como objetivo definir o escopo do problema considerado.

3.3 Modelagem Restrita do Problema

Uma formulação matemática restrita é apresentada para descrever de forma resumida os aspectos do problema abordado. A modelagem é restrita no sentido de que apenas é modelado o planejamento horizontal do VANT. A dimensão altitude é determinada através de uma queda constante. Essa formulação vem de uma modelagem semelhante, descrita em [Blackmore, Ono e Williams \(2011\)](#), para o planejamento de caminho com alocação de riscos. A hipótese de decaimento linear da altitude será utilizada como base no desenvolvimento dos métodos: Heurística Gulosa (HG), Algoritmo Genético (AG) e Algoritmo Genético Multi-Populacional (AGMP). A seguir, encontram-se alguns parâmetros e variáveis de decisão utilizados na formulação matemática.

Parâmetros:

- $\Phi^l = \{Z_{\Phi^l}^1, Z_{\Phi^l}^2 \dots Z_{\Phi^l}^{|\Phi^l|}\}$: Conjunto de regiões com $l \in \{n, p, b, r\}$;
- $Z_{\Phi^l}^i$: i -ésima região do conjunto Φ^l ;

- $|\Phi^l|$: Número de regiões no conjunto Φ^l ;
- C_{Φ^l} : Custo de pouso no conjunto Φ^l ;
- T : Número de passos de tempo para pouso do VANT;
- Δ : Probabilidade do VANT violar uma região no conjunto Φ^n ;
- F_Ψ : Função de transição de estados para um determinado tipo de falha Ψ ;
- ω_t : Perturbação independente do estado, no instante t ;
- Q : Matriz 4×4 da covariância associada à incerteza do ambiente;
- Σ_{x_0} : Matriz 4×4 da covariância associada à incerteza do VANT;
- F^d : Resistência do ar ou força do arrasto;
- ΔT : Discretização do tempo com intervalo de tempo fixo.

Variáveis de Decisão:

- x_t : Conjunto de estados do VANT ($x_t = [p_t^x \ p_t^y \ v_t \ \alpha_t]'$);
- p_t^x : Posição no eixo x da aeronave no instante t ;
- p_t^y : Posição no eixo y da aeronave no instante t ;
- v_t : Velocidade horizontal do VANT no instante t ;
- α_t : Ângulo horizontal do VANT no instante t ;
- u_t : Conjunto de controles do VANT ($u_t = [a_t \ \varepsilon_t]'$);
- a_t : Aceleração na horizontal do VANT no instante t ;
- ε_t : Variação angular na horizontal do VANT no instante t .

Formulação Matemática:

$$\text{Minimizar} \sum_{i=1}^{|\Phi^p|} (C_{\Phi^p} \cdot Pr(x_T \in Z_{\Phi^p}^i)) - \sum_{i=1}^{|\Phi^b|} (C_{\Phi^b} \cdot Pr(x_T \in Z_{\Phi^b}^i)) \quad (3.1)$$

sujeito a:

$$x_{t+1} = F_\Psi(x_t, u_t) + \omega_t \quad \forall t = 0, 1, 2, \dots, T \quad (3.2)$$

$$Pr \left(\bigwedge_{t=0}^T \bigwedge_{i=1}^{|\Phi^n|} x_t \notin Z_{\Phi^n}^i \right) \geq 1 - \Delta \quad (3.3)$$

Uma discretização do horizonte de planejamento T é assumida com $t = 0, 1, 2 \dots T$. A variável de decisão x_t representa o estado do VANT no instante t que é dado por sua coordenada no plano cartesiano (p_t^x, p_t^y), velocidade (v_t) e direção angular nesse plano (α_t). A variável u_t representa os controles aplicados na aeronave em t , onde temos a aceleração (a_t) e a variação angular (ε_t).

A função objetivo (3.1) minimiza danos ao executar um pouso de emergência. Ela é formada por duas partes: penalização e bonificação. A primeira parte tem as penalidades (C_{Φ^p}) para uma rota que pousa a aeronave no último passo de tempo ($t = T$) dentro do conjunto de regiões penalizadoras ($x_T \in Z_{\Phi^p}^i$). A segunda parte tem as recompensas (C_{Φ^b}) para uma rota que pousa a aeronave dentro das regiões bonificadoras ($x_T \in Z_{\Phi^b}^i$). A função $Pr(\cdot)$ utilizada nesta formulação representa a probabilidade da aeronave no estado x_t pertencer ou não a uma região particular em Φ^l .

Assim, como proposto em [Blackmore, Ono e Williams \(2011\)](#), é assumido que as incertezas e pertubações (ω_t) seguem uma distribuição Gaussiana com ruído branco $\omega_t \sim \mathcal{N}(0, Q)$ com média zero e matriz de covariância Q . Nessa modelagem, considera-se a incerteza da posição como sendo constante ao longo do tempo. Também é assumido que o estado inicial segue uma distribuição Gaussiana com média \hat{x}_0 e matriz de covariância Σ_{x_0} , então $x_0 \sim \mathcal{N}(\hat{x}_0, \Sigma_{x_0})$. Além disso, o estado futuro também segue uma distribuição Gaussiana e x_t é uma variável aleatória.

As restrições (3.2) descrevem as transições de estados do VANT, onde a dinâmica dos estados do VANT no tempo $t + 1$ é definida a partir de posições e controles aplicados no instante t mais a incerteza ω_t causada por fatores externos. A função F_Ψ depende de cada tipo de situação crítica ocorrida (Ψ). Os modelos de dinâmica desenvolvidos consideram um VANT de asa fixa com sistema de decolagem e aterrissagem convencional sobre pistas (CTOL). A expressão geral que representa as mudanças de estados do VANT é dada por:

$$x_{t+1} = F_\Psi(x_t, u_t) \Leftrightarrow \begin{bmatrix} p_{t+1}^x \\ p_{t+1}^y \\ v_{t+1} \\ \alpha_{t+1} \end{bmatrix} = \begin{bmatrix} p_t^x + v_t \cdot \cos(\alpha_t) \cdot \Delta T + a_t \cdot \cos(\alpha_t) \cdot (\Delta T)^2 / 2 \\ p_t^y + v_t \cdot \sin(\alpha_t) \cdot \Delta T + a_t \cdot \sin(\alpha_t) \cdot (\Delta T)^2 / 2 \\ v_t + a_t \cdot \Delta T - \frac{F_t^d}{m} \cdot \Delta T \\ \alpha_t + \varepsilon_t \cdot \Delta T \end{bmatrix} \quad (3.4)$$

Esse sistema não linear $F_\Psi(x_t, u_t)$ define as transições de estados. As novas posições (p_{t+1}^x, p_{t+1}^y) dependem apenas de dados do estado corrente ($p_t^x, p_t^y, v_t, \alpha_t$) e do controle de aceleração (a_t). O novo valor da velocidade será dado pela velocidade do estado corrente (v_t), pelo controle de aceleração do estado atual (a_t) e pela resistência do ar. Isso está representado na variável F^d (veja a Equação (3.5)), também conhecida como equação do arrasto. Por fim, o novo

valor da direção apontada pelo aeronave será dado pelo valor do estado corrente (α_t) e por uma nova variação angular dada por (ε_t).

A equação do arrasto (F_t^d) em um dado tempo é apresentada na Equação (3.5) que possui como parâmetros: coeficiente de arrasto (C_d), que é específico para cada aeronave, massa específica do fluido (ρ) e área de referência (\mathcal{A}) (STUDART; DAHMEN, 2006).

$$F_t^d = \frac{1}{2} \cdot C_d \cdot \rho \cdot \mathcal{A} \cdot v_t^2 \quad (3.5)$$

A seguir são apresentadas as funções de transição de estado de acordo com cada uma das situações críticas.

$$x_{t+1} = F_{\Psi^m}(x_t, u_t) \Leftrightarrow \begin{bmatrix} p_{t+1}^x \\ p_{t+1}^y \\ v_{t+1} \\ \alpha_{t+1} \end{bmatrix} = \begin{bmatrix} p_t^x + v_t \cdot \cos(\alpha_t) \cdot \Delta T \\ p_t^y + v_t \cdot \sin(\alpha_t) \cdot \Delta T \\ v_t - \frac{F_t^d}{m} \cdot \Delta T \\ \alpha_t + \varepsilon_t \cdot \Delta T \end{bmatrix} \quad (3.6)$$

$$x_{t+1} = F_{\Psi^{s1}}(x_t, u_t) \Leftrightarrow \begin{bmatrix} p_{t+1}^x \\ p_{t+1}^y \\ v_{t+1} \\ \alpha_{t+1} \end{bmatrix} = \begin{bmatrix} p_t^x + v_t \cdot \cos(\alpha_t) \cdot \Delta T + a_t \cdot \cos(\alpha_t) \cdot (\Delta T)^2 / 2 \\ p_t^y + v_t \cdot \sin(\alpha_t) \cdot \Delta T + a_t \cdot \sin(\alpha_t) \cdot (\Delta T)^2 / 2 \\ v_t + a_t \cdot \Delta T - \frac{F_t^d}{m} \cdot \Delta T \\ \alpha_t + \max\{0, \varepsilon_t\} \cdot \Delta T \end{bmatrix} \quad (3.7)$$

$$x_{t+1} = F_{\Psi^{s2}}(x_t, u_t) \Leftrightarrow \begin{bmatrix} p_{t+1}^x \\ p_{t+1}^y \\ v_{t+1} \\ \alpha_{t+1} \end{bmatrix} = \begin{bmatrix} p_t^x + v_t \cdot \cos(\alpha_t) \cdot \Delta T + a_t \cdot \cos(\alpha_t) \cdot (\Delta T)^2 / 2 \\ p_t^y + v_t \cdot \sin(\alpha_t) \cdot \Delta T + a_t \cdot \sin(\alpha_t) \cdot (\Delta T)^2 / 2 \\ v_t + a_t \cdot \Delta T - \frac{F_t^d}{m} \cdot \Delta T \\ \alpha_t + \min\{0, \varepsilon_t\} \cdot \Delta T \end{bmatrix} \quad (3.8)$$

A primeira modelagem apresentada F_Ψ (Equação (3.4)) é equivalente a aeronave voando sem falha associada F_{Ψ^0} . O modelo que representa o problema no motor F_{Ψ^m} (Equação (3.6)) não possui o componente do controle de aceleração, onde foi assumido que o motor deixou de funcionar. No caso de problemas relacionados à superfície aerodinâmica 1 $F_{\Psi^{s1}}$ (Equação (3.7)), o VANT não pode efetuar manobra à direita. No problema da superfície aerodinâmica 2 $F_{\Psi^{s2}}$ (Equação (3.8)), o veículo aéreo não pode efetuar manobra à esquerda. A expressão do problema na bateria não é apresentada já que durante tal falha todos os componentes funcionam (função de transição completa). Porém, a aeronave é penalizada proporcionalmente ao seu tempo de voo, caso continue a voar com problema na bateria.

Devido as incertezas associadas a localização do VANT e ao risco de se desviar de sua rota, atingindo uma região não navegável, um limite para esse risco deve ser considerado durante o planejamento de trajetória. Logo, a Restrição (3.3) descreve a probabilidade $(1 - \Delta)$ do VANT

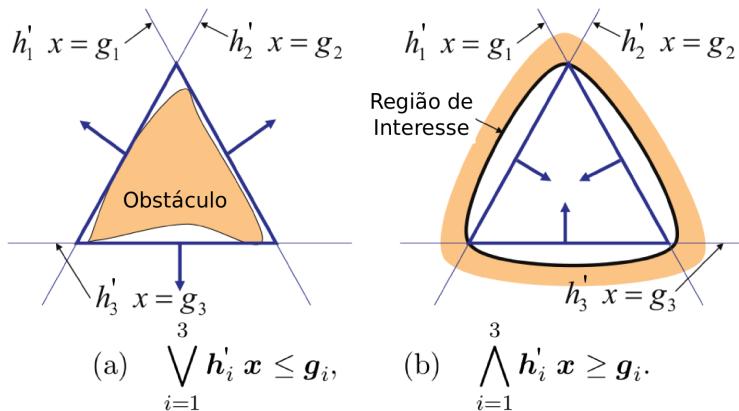
estar fora de regiões que pertencem ao conjunto Φ^n . Dessa forma, Δ representa a probabilidade do VANT violar regiões não navegáveis.

Ao executar um método que minimize a Equação (3.1), uma solução contendo uma rota e um local de pouso será obtida. A trajetória final será retornada, representada por Ω , contendo um conjunto de *waypoints* para a rota de pouso emergencial (veja Equação (3.9)). O local de pouso da aeronave é representado por Φ^* conforme descrito na Equação (3.10).

$$\Omega = \{(p_0^x, p_0^y), (p_1^x, p_1^y), (p_2^x, p_2^y), \dots, (p_T^x, p_T^y)\} \quad (3.9)$$

$$\Phi^* = \{\Phi \in \{\Phi^n, \Phi^p, \Phi^b, \Phi^r\} \wedge i \in \{1, 2, 3, \dots, |\Phi|\} \mid (p_T^x, p_T^y) \in Z_\Phi^i\} \quad (3.10)$$

Figura 11 – Representação de regiões factíveis. Cada região factível foi aproximada por um triângulo e por um conjunto de restrições lineares. (a) região factível fora do obstáculo. (b) região factível dentro do obstáculo.



Fonte: Adaptada de [Ono, Williams e Blackmore \(2013\)](#).

A Figura 11 mostra dois exemplos simples de como tratar as questões de pertencer ou não a um determinado polígono. A situação de estar fora de um obstáculo Figura 11 (a), utiliza o operador \vee que indica o OR lógico, onde a aeronave deve estar fora de uma ou mais retas. Assim, ela estará fora do obstáculo. Uma outra situação possível é permanecer dentro de uma área passível de voo Figura 11(b), onde o operador \wedge representa o AND lógico, e para que a aeronave esteja dentro dessa região então deve estar dentro de todas as retas. Ambas as situações podem ser descritas como regiões factíveis aproximadas através de restrições lineares.

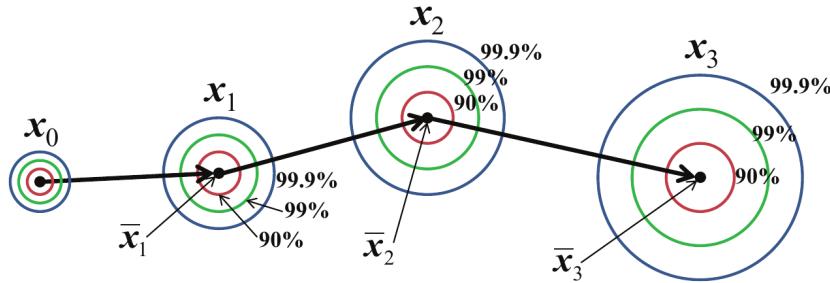
3.4 Modelagem Completa do Problema

Nesta seção, uma formulação matemática para sumarizar todos os aspectos do problema abordado é apresentada, usando Programação Linear Inteira Mista (PLIM). Essa formulação é similar a modelagem descrita em [Blackmore, Ono e Williams \(2011\)](#), porém o modelo aqui

proposto incorpora o pouso de VANTs diante de uma situação crítica. A variável de decisão x_t também representa o estado do VANT no momento t , mas fornecendo agora sua coordenada no plano cartesiano (p_t^x, p_t^y, p_t^z) e sua velocidade (v_t^x, v_t^y, v_t^z). A variável de decisão u_t define os controles aplicados sobre a aeronave em t , representado agora pelas acelerações (a_t^x, a_t^y, a_t^z).

Na presente modelagem, a incerteza da posição aumenta através do horizonte de tempo como descrito em [Ono, Williams e Blackmore \(2013\)](#) e ilustrado na Figura 12. As perturbações são adicionadas em cada instante de tempo.

Figura 12 – Ilustração do modelo de planejamento com incremento da incerteza em cada instante de tempo.



Fonte: [Ono, Williams e Blackmore \(2013\)](#).

A seguir são definidos os índices, parâmetros e variáveis de decisão utilizados no modelo.

Índices:

- $t = 0, 1, 2 \dots T$: Índice do horizonte de planejamento;
- $l \in \{n, p, b, r\}$: Índice para os tipos de regiões do mapa;
- $j = 1, 2, 3 \dots |\Phi^l|$: Índice para as regiões do tipo Φ^l ;
- $i = 1, 2, 3 \dots |\Phi_j^l|$: Índice para as arestas da região Φ_j^l ;
- $k = 1, 2, 3 \dots |G|$: Índice para as arestas do polígono G .

Parâmetros:

- \hat{x}_0 : Estado inicial esperado do VANT, onde $\hat{x}_0 := [p_0^x \ p_0^y \ p_0^z \ v_0^x \ v_0^y \ v_0^z]'$;
- A : Matriz 6×6 para a mudança do estado do VANT;
- B : Matriz 6×3 para os controles associados ao VANT;
- Σ_t : Matriz 6×6 da covariância associada a incerteza do VANT;
- ∇ : Número de arestas das regiões dos mapas;
- τ^{min} : Delimitação do tempo mínimo de voo;

- τ^{max} : Delimitação do tempo máximo de voo;
- v_{max}^h : Velocidade horizontal máxima permitida para o VANT;
- v_{max}^v : Velocidade vertical máxima permitida para o VANT;
- v_{min}^h : Velocidade horizontal mínima permitida para o VANT;
- v_{min}^v : Velocidade vertical mínima permitida para o VANT;
- a_{max}^h : Aceleração horizontal máxima permitida para o VANT;
- a_{max}^v : Aceleração vertical máxima permitida para o VANT;
- σ^h : Incerteza associada a posição da aeronave na horizontal;
- σ^v : Incerteza associada a posição da aeronave na vertical;
- η : Fator de crescimento da incerteza a cada instante de tempo;
- a_k : Vetor normal à aresta k ;
- a_{ji} : Vetor normal à aresta i do polígono j ;
- b_{ji} : Valor constante da aresta i do polígono j ;
- G : Poliedro/polígono usado na linearização;
- Γ : Número de arestas do polígono G usado na linearização;
- G_{max}^v : Hiperplanos das faces externas do polígono das velocidades;
- G_{min}^v : Hiperplanos das faces internas do polígono das velocidades;
- G_{max}^a : Hiperplanos das faces externas do polígono dos controles;
- h^j : Altitude da região j ;
- $h[v]$: Altitude de uma região qualquer do mapa, onde v representa a altura.

Variáveis de decisão [Contínuas]:

- \bar{x}_t : Estado esperado do VANT no instante t , onde $\bar{x}_t := [p_t^x \ p_t^y \ p_t^z \ v_t^x \ v_t^y \ v_t^z]'$;
- p_t^x : Posição no eixo x do VANT no instante t ;
- p_t^y : Posição no eixo y do VANT no instante t ;
- p_t^z : Posição no eixo z do VANT no instante t ;

- v_t^x : Velocidade no eixo x do VANT no instante t ;
- v_t^y : Velocidade no eixo y do VANT no instante t ;
- v_t^z : Velocidade no eixo z do VANT no instante t ;
- \bar{u}_t : Controles esperados do VANT, no instante t , onde $\bar{u}_t := [a_t^x \ a_t^y \ a_t^z]'$;
- a_t^x : Aceleração no eixo x do VANT no instante t ;
- a_t^y : Aceleração no eixo y do VANT no instante t ;
- a_t^z : Aceleração no eixo z do VANT no instante t ;
- $P_j^{\Phi^b}$: Probabilidade do VANT poussar na região bonificadora j ;
- $L_j^{\Phi^b}$: Controla o pouso em apenas uma região bonificadora j ;
- S_t : Possui o valor 1 apenas no instante exato que o VANT pousou;
- δ_{jt}^n : Risco associado à colisão com os lados da região não navegável j no instante t ;
- δ_{jt}^p : Risco associado à colisão com os lados da região penalizadora j no instante t ;
- δ_{jii}^b : Risco associado à violação do hiperplano i da região bonificadora j no instante t .

Variáveis de decisão [Binárias]:

- H_t : 1 indica VANT em voo, 0 indica pouso do VANT;
- W_{kt} : 1 indica superar o limite inferior de velocidade no hiperplano k no instante t ;
- $Y_{jt}^{\Phi^b}$: 1 queda do VANT na j -ésima região de Φ^b no instante t ;
- $Z_{jii}^{\Phi^n}$: 1 indica estar fora do hiperplano i da j -ésima região de Φ^n no instante t ;
- $Z_{jii}^{\Phi^p}$: 1 indica estar fora do hiperplano i da j -ésima região de Φ^p no instante t .

Matrizes da dinâmica e covariância:

$$A = \begin{bmatrix} 1 & 0 & 0 & \Delta T & 0 & 0 \\ 0 & 1 & 0 & 0 & \Delta T & 0 \\ 0 & 0 & 1 & 0 & 0 & \Delta T \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

$$B = \begin{bmatrix} \frac{1}{2}\Delta T^2 & 0 & 0 \\ 0 & \frac{1}{2}\Delta T^2 & 0 \\ 0 & 0 & \frac{1}{2}\Delta T^2 \\ \Delta T & 0 & 0 \\ 0 & \Delta T & 0 \\ 0 & 0 & \Delta T \end{bmatrix}$$

$$\Sigma_t = \begin{bmatrix} (\sigma_t^h)^2 & 0 & 0 & 0 & 0 & 0 \\ 0 & (\sigma_t^h)^2 & 0 & 0 & 0 & 0 \\ 0 & 0 & (\sigma_t^v)^2 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

Função objetivo:

A função objetivo (3.11) minimiza o quadrado dos controles da aeronave. Devido ao fato de $\bar{u}'_t \cdot \bar{u}_t$ ser não linear, esse valor será aproximado através de um polígono de Γ faces. Uma bonificação será dada ao VANT que conseguir pousar em alguma das regiões bonificadoras. Uma segunda função objetivo (3.12) foi definida e é semelhante a primeira, no entanto, penaliza por tempo de voo e, em geral, faz com que soluções mais rápidas sejam retornadas. As duas versões de funções objetivos serão testadas e comparadas nesse trabalho.

$$\text{Minimizar} \sum_{t=0}^{T-1} (\bar{u}'_t \cdot \bar{u}_t) - C_{\Phi^b} \cdot \sum_{j=1}^{|\Phi^b|} P_j^{\Phi^b} \quad (3.11)$$

$$\text{Minimizar} \sum_{t=0}^{T-1} u'_t \cdot u_t \cdot 2^{\frac{t-\tau^{max}}{10}} - C_{\Phi^b} \sum_{j=1}^{|\Phi^b|} P_j^{\Phi^b} + C_{\Phi^b} \sum_{t=1}^T S_t \cdot 2^{\frac{t-\tau^{max}}{10}} \quad (3.12)$$

Restrições:

A seguir é apresentado um conjunto de restrições que descrevem as diversas características consideradas neste problema.

Equação de dinâmica

A dinâmica de voo seguida pela aeronave é dada pelas restrições (3.13), sendo ativada quando a variável H_t ocorre (VANT em voo). Essas restrições estabelecem uma dinâmica linear, enquanto a Equação 3.4 do modelo restrito apresentava uma dinâmica não linear.

$$H_t \Rightarrow \bar{x}_{t+1} = A\bar{x}_t + B\bar{u}_t \quad \forall t = 0, 1, 2, \dots, T-1 \quad (3.13)$$

Estado inicial

A trajetória de pouso é determinada a partir do estado inicial como descrito na Restrição (3.14). O parâmetro \hat{x}_0 caracteriza a posição e as velocidades desempenhadas pela aeronave no instante da situação crítica. No contexto deste problema, tais informações seriam fornecidas pelo sistema IFA.

$$\bar{x}_0 = \hat{x}_0 \quad (3.14)$$

Definindo o horizonte de planejamento

O conjunto de restrições (3.15)-(3.20) determina o comportamento da variável que define o estado do VANT no horizonte de planejamento. A aeronave parte inicialmente do estado voando e deve estar no solo ao final do horizonte de planejamento. Os parâmetros τ^{\min} e τ^{\max} delimitam, respectivamente, o tempo mínimo e máximo de voo, sendo assim, a aeronave não poderá atingir o solo antes de τ^{\min} e nem depois de τ^{\max} . A notação $\forall t$ omite o intervalo, assim, será considerado que t varia no intervalo $t = 0, 1, 2, \dots, T$.

$$H_t = 1 \quad (\text{caso } t = 0) \quad (3.15)$$

$$H_t = 0 \quad (\text{caso } t = T) \quad (3.16)$$

$$H_t \leq H_{t-1} \quad \forall t \quad (3.17)$$

$$H_t = 1 \quad \forall t < \tau^{\min} \quad (3.18)$$

$$H_t = 0 \quad \forall t > \tau^{\max} \quad (3.19)$$

$$0 \leq H_t \leq 1 \quad \forall \tau^{\min} \leq t \leq \tau^{\max} \quad (3.20)$$

Limits superiores de velocidade

As restrições (3.22), (3.23) e (3.24) limitam superiormente a velocidade da aeronave. Nota-se que as restrições de velocidade somente se aplicam caso a aeronave esteja voando (H_t ativado). Os valores assumidos por a_k nas equações (3.21) determinam um polígono de Γ faces. Os três primeiros valores são iguais a zero (0) em a_k , indicando que no produto vetorial com o vetor \bar{x}_t apenas as componentes das velocidade em x e y importam. Assim, as restrições (3.22) limitam a velocidade do VANT a estar dentro desse polígono de velocidades. A Figura 13 ilustra como é a limitação superior da velocidade na aeronave. Na restrições citadas, G_{\max}^v representa apenas as faces externas do poliedro da Figura 13 ("circulo" mais externo). A

expressão $a'_k \cdot \bar{x}_t \leq v_{max}^h$ garante que a velocidade horizontal seja sempre menor que o valor limite v_{max}^h definido.

$$a'_k = \left[0 \ 0 \ 0 \cos\left(\frac{2\pi k}{\Gamma}\right) \ \sin\left(\frac{2\pi k}{\Gamma}\right) \ 0 \right] \quad \forall k \in G_{max}^v \quad (3.21)$$

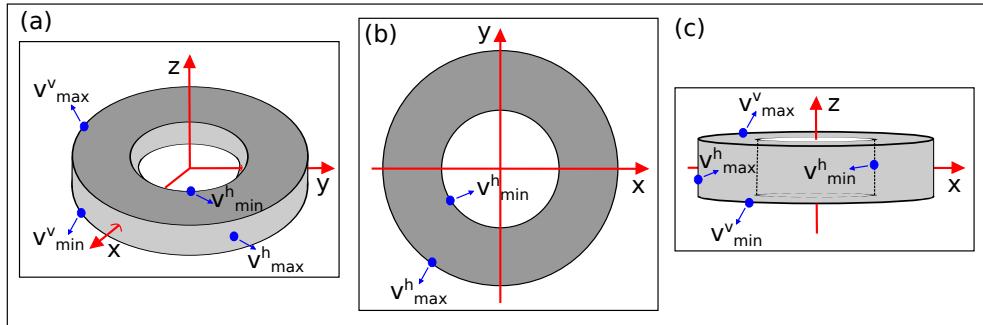
$$H_t \Rightarrow a'_k \cdot \bar{x}_t \leq v_{max}^h \quad \forall t, k \in G_{max}^v \quad (3.22)$$

$$H_t \Rightarrow \bar{x}_t^z \leq v_{max}^y \quad \forall t \quad (3.23)$$

$$H_t \Rightarrow \bar{x}_t^z \geq v_{min}^y \quad \forall t \quad (3.24)$$

Existem outras formas de fazer o cálculo da velocidade horizontal desenvolvida pela aeronave como $\sqrt{(\bar{x}.v_t^x)^2 + (\bar{x}.v_t^y)^2} \leq v_{max}^h$, onde $\bar{x}.v_t^x$ representa a velocidade no eixo x e $\bar{x}.v_t^y$ representa a velocidade no eixo y ambos no instante t , no entanto, ao usar tal expressão teríamos um modelo não linear. Por se tratar de um modelo baseado em PLIM todas as expressões desenvolvidas são lineares ou foram linearizadas através de algum método.

Figura 13 – Formato dos limitantes das velocidades para o modelo matemático. Em (a), (b) e (c) são apresentados diferentes visões do mesmo poliedro/polígono das velocidades. Esse polígono é aproximado através de Γ faces.



Fonte: Elaborada pelo autor.

Limits inferiores de velocidade

As restrições (3.26) definem um limitante inferior para a velocidade da aeronave. A variável W_{kt} garante que a aeronave esteja fora de um dos hiperplanos em todos os instantes de voo estabelecidos nas restrições (3.27). Os valores assumidos por a_k nas equações (3.25) definem um polígono de Γ faces, usado para definir a disjunção no espaço. A Figura 13 apresenta o formato dos limitantes superiores e inferiores da velocidade da aeronave. Nessas restrições, G_{min}^v representa apenas as faces internas do poliedro da Figura 13 ("circulo" mais interno).

$$a'_k = \left[0 \ 0 \ 0 \cos\left(\frac{2\pi k}{\Gamma}\right) \ \sin\left(\frac{2\pi k}{\Gamma}\right) \ 0 \right] \quad \forall k \in G_{min}^v \quad (3.25)$$

$$W_{kt} \Rightarrow a'_k \cdot \bar{x}_t \geq v_{min}^h \quad \forall t, k \in G_{min}^v \quad (3.26)$$

$$\sum_{k \in G_{min}^v} W_{kt} \geq H_t \quad \forall t \quad (3.27)$$

Limits dos controles

As restrições (3.29), (3.30) e (3.31) estabelecem os limites aplicados aos controles do VANT. Tais limites também são aproximados por um polígono de Γ faces. A Equação (3.28) descreve os hiperplanos do polígono. A Figura 14 mostra o formato dos limitantes superiores dos controles (acelerações). Observe que não existem limites inferiores. Nas restrições citadas, G_{max}^a representa as faces externas do poliedro da Figura 14 ("circulo" mais externo).

$$a'_k = \left[\cos\left(\frac{2\pi k}{\Gamma}\right) \sin\left(\frac{2\pi k}{\Gamma}\right) 0 \right] \quad \forall k \in G_{max}^a \quad (3.28)$$

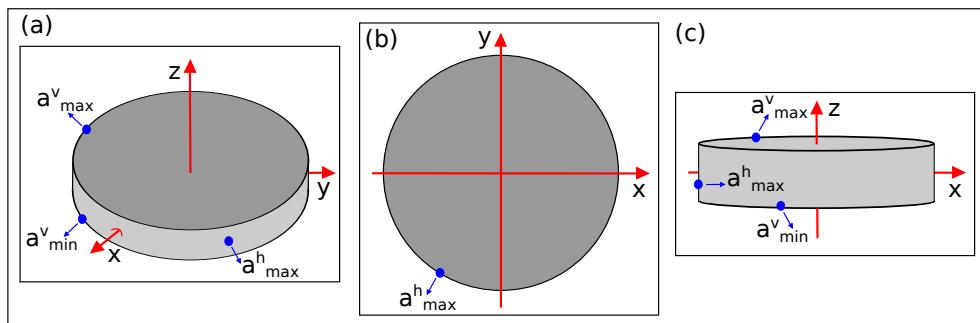
$$H_t \Rightarrow a'_k \cdot \bar{u}_t \leq a_{max}^h \quad \forall t = 0, 1, 2 \dots T-1, k \in G_{max}^a \quad (3.29)$$

$$H_t \Rightarrow \bar{u}_t^z \leq a_{max}^v \quad \forall t = 0, 1, 2 \dots T-1 \quad (3.30)$$

$$H_t \Rightarrow \bar{u}_t^z \geq a_{min}^v \quad \forall t = 0, 1, 2 \dots T-1 \quad (3.31)$$

O fato de definirmos um limite inferior para a velocidade horizontal indica que o VANT não pode tentar voar abaixo desse valor, caso contrário a aeronave cairia. O fato de não haver um limite inferior para as acelerações horizontais indica que o VANT pode voar sem acelerações, desde que, tenha velocidade horizontal maior que o valor mínimo.

Figura 14 – Formato dos limitantes das acelerações para o modelo matemático. Em (a), (b) e (c) são apresentados diferentes visões do mesmo poliedro/polígonos das acelerações. Esse polígonos é aproximado através de Γ faces.



Fonte: Elaborada pelo autor.

Capturando instante de pouso

A aeronave irá efetuar o pouso no instante em que existir diferença entre os estados como definido pelas restrições (3.32). As restrições (3.33) fazem com que, se o VANT pousou (S_t ativado), a variável binária $Y_{jt}^{\Phi^b}$ seja ativada em apenas um instante e numa única região bonificadora.

As restrições (3.34) garantem que o avião, ao pousar sobre uma região bonificadora, não atinja o topo da mesma e mantenha uma margem de segurança. A notação \neg indica a negação da variável binária H_t . A função $d_t(\cdot)$ retorna uma distância de segurança do topo da região bonificadora, onde a aeronave deverá permanecer a cada instante t com 10% do valor do risco (Δ). Essa distância garante que as soluções do modelo não irão colidir com o topo das regiões bonificadoras.

$$S_t = H_{t-1} - H_t \quad \forall t \quad (3.32)$$

$$S_t \geq \sum_{j=1}^{|\Phi^b|} Y_{jt}^{\Phi^b} \quad \forall t \quad (3.33)$$

$$\neg H_t \Rightarrow \bar{x}_t^z - \sum_{j=1}^{|\Phi^b|} h^j \cdot Y_{jt}^{\Phi^b} = d_t(0, 1 \cdot \Delta) \quad \forall t \quad (3.34)$$

Função de alocação de risco

Na Equação (3.35), temos a função $c_{jit}(\xi)$ que aloca o risco de colisão na variável ξ no instante t com o hiperplano i do obstáculo j . Apesar da função inversa do erro ($erf^{-1}(\cdot)$) ser não linear, uma aproximação linear por partes é utilizada como definido em [Arantes, Toledo e Williams \(2016\)](#).

$$c_{jit}(\xi) = erf^{-1}(1 - 2\xi) \sqrt{2a'_{ji} \cdot \Sigma_t \cdot a_{ji}} \quad (3.35)$$

A Expressão (3.38) descreve o parâmetro a'_{ji} que representa o vetor normal ao hiperplano i do obstáculo j . A matriz Σ_t descreve a incerteza da posição nos eixos horizontais e verticais. Neste trabalho, consideramos um crescimento da incerteza de forma semelhante ao trabalho de [Ono, Williams e Blackmore \(2013\)](#). Dessa forma, esse crescimento ao longo do eixo horizontal ocorre seguindo a Equação 3.36. Não foi considerado crescimento da incerteza ao longo do eixo vertical, como demonstra a Equação 3.37. Os valores dos parâmetros σ^h e σ^v utilizados estão definidos na seção de resultados 5.3.1.

$$\sigma_t^h = \left(\sigma^h \cdot \left(1 + \frac{\eta \cdot t}{T} \right) \right)^2 \quad (3.36)$$

$$\sigma_t^v = \sigma^v \quad (3.37)$$

Colisão com regiões não navegáveis

As restrições (3.39) garantem que a aeronave em voo fique fora de pelo menos um dos hiperplanos das regiões não navegáveis. $|\Phi_j^n|$ indica a quantidade de arestas (hiperplanos) da j -ésima região não navegável. O valor de $a'_{ji} \cdot \bar{x}_t \geq b_{ji}$ nas restrições (3.40) estabelece que o VANT no instante t esteja fora da aresta i do obstáculo j . Ao somar o valor de $c_{jut}(.)$, estamos garantindo uma margem de segurança (risco) entre a aeronave e o obstáculo. A Expressão (3.38) descreve os parâmetros a'_{ji} e b_{ji} que formam os hiperplanos i dos obstáculos não navegáveis j . Os valores representados por a_{ji}^1 , a_{ji}^2 e a_{ji}^3 indicam números que fazem com que a'_{ji} seja normal a aresta i do polígono j . A função $c_{jut}(.)$ na Restrição (3.40) efetua a alocação do risco, na variável δ_{jt}^n , para cada hiperplano i da região não navegável j e em cada instante t .

$$a'_{ji} = [a_{ji}^1 \ a_{ji}^2 \ a_{ji}^3 \ 0 \ 0 \ 0], \quad b_{ji} \quad \forall j \in \Phi^n, i \in \Phi_j^n \quad (3.38)$$

$$\sum_{i=1}^{|\Phi_j^n|} Z_{jut}^{\Phi^n} \geq H_t \quad \forall t, j \in \Phi^n \quad (3.39)$$

$$Z_{jut}^{\Phi^n} \Rightarrow a'_{ji} \cdot \bar{x}_t \geq b_{ji} + c_{jut}(\delta_{jt}^n) \quad \forall t, j \in \Phi^n, i \in \Phi_j^n \quad (3.40)$$

Colisão com regiões penalizadoras

De forma semelhante à colisão com regiões não navegáveis, a região penalizadora também possui limitação de pouso. Todavia, a aeronave pode sobrevoar tal região. As restrições (3.42) definem que, caso a aeronave tenha pousado (S_t ativo), a mesma deve estar fora dos hiperplanos das regiões penalizadoras. As restrições (3.43) alocam o risco para o pouso próximo à regiões penalizadoras. Os parâmetros a'_{ji} e b_{ji} nesse caso delimitam as regiões penalizadoras conforme mostra a Expressão 3.41.

$$a'_{ji} = [a_{ji}^1 \ a_{ji}^2 \ a_{ji}^3 \ 0 \ 0 \ 0], \quad b_{ji} \quad \forall j \in \Phi^p, i \in \Phi_j^p \quad (3.41)$$

$$\sum_{i=1}^{|\Phi_j^p|} Z_{jut}^{\Phi^p} \geq S_t \quad \forall t, j \in \Phi^p \quad (3.42)$$

$$Z_{jut}^{\Phi^p} \Rightarrow a'_{ji} \cdot \bar{x}_t \geq b_{ji} + c_{jut}(\delta_{jt}^p) \quad \forall t, j \in \Phi^p, i \in \Phi_j^p \quad (3.43)$$

Alocação dos riscos

A Restrição (3.44) garante que o risco total na rota planejada não ultrapasse o limite estabelecido Δ . Dessa forma, durante toda a rota de pouso a aeronave se manterá a uma certa distância das regiões não navegáveis.

$$\sum_{t=0}^T \sum_{j=1}^{|\Phi^n|} \delta_{jt}^n \leq \Delta \quad (3.44)$$

Pousar em região bonificadora ou remanescente

A Restrição (3.45) garante que o pouso ocorra no máximo em uma região bonificadora. As restrições (3.46) definem que, se o VANT pousar em uma região bonificadora, então uma variável $Y_{jt}^{\Phi^b}$ será usada para alocar o risco durante o pouso nas restrições (3.47). O fato de $c_{jut}(.)$ ser negativo indica que a aeronave deve estar dentro da j -ésima região bonificadora (Φ_j^b), mantendo uma certa distância das arestas dessa região. Logo, deve estar mais ao centro dessa região. As restrições (3.48) determinam a probabilidade de sucesso no pouso em regiões bonificadoras pela variável $P_j^{\Phi^b}$. Por fim, as restrições (3.49) fazem com que a variável $Y_{jt}^{\Phi^b}$ não seja ativada quando a aeronave estiver em voo. Esse conjunto de restrições garante que o pouso ocorra apenas nas regiões bonificadoras ou em regiões remanescentes.

$$\sum_{j=1}^{|\Phi^b|} L_j^{\Phi^b} \leq 1 \quad (3.45)$$

$$\sum_{t=0}^T Y_{jt}^{\Phi^b} = L_j^{\Phi^b} \quad \forall j \in \Phi^b \quad (3.46)$$

$$Y_{jt}^{\Phi^b} \Rightarrow a'_{ji} \cdot \bar{x}_t \leq b_{ji} - c_{jut}(\delta_{jut}^b) \quad \forall t, j \in \Phi^b, i \in \Phi_j^b \quad (3.47)$$

$$L_j^{\Phi^b} \Rightarrow 1 - \sum_{t=0}^T \sum_{i=1}^{|\Phi_j^b|} \delta_{jut}^b = P_j^{\Phi^b} \quad \forall j \in \Phi^b \quad (3.48)$$

$$H_t \leq 1 - \sum_{j=1}^{|\Phi^b|} Y_{jt}^{\Phi^b} \quad \forall t \quad (3.49)$$

Tratamentos das Situações Críticas:

O modelo matemático para o problema descrito até o presente momento é um modelo geral, onde não estão sendo considerados falhas específicas na aeronave. A seguir são apresentadas os tratamentos das situações críticas.

Problema no motor (Ψ^m)

No caso da aeronave apresentar um problema no motor, foi definido que o VANT desceria com velocidade praticamente constante e teria uma velocidade horizontal também próxima a um

valor constante. Logo, as restrições (3.23) e (3.24) devem ser substituídas pelas restrições (3.50) e (3.51) no modelo para melhor controlar a velocidade de descida. Essas restrições garantem que o VANT desça com velocidade vertical quase constante, entre $v_m^v - \mu$ e $v_m^v + \mu$ metros por segundo, onde μ representa um valor relativamente pequeno.

$$H_t \Rightarrow \bar{x}_v^z \leq v_m^v + \mu \quad \forall t \quad (3.50)$$

$$H_t \Rightarrow \bar{x}_v^z \geq v_m^v - \mu \quad \forall t \quad (3.51)$$

Para garantir que o VANT desenvolva uma velocidade horizontal praticamente constante. A Restrição (3.22), que trata o limite superior da velocidade, deve ser substituída pela Restrição (3.52). A Restrição (3.26), que trata o limite inferior de velocidade, deve ser substituída pela Restrição (3.53) de forma análoga ao caso anterior. Através desses reajustes dos parâmetros da velocidade horizontal máxima ($v_m^h + \mu$) e mínima ($v_m^h - \mu$), o veículo aéreo desenvolverá uma velocidade próxima de v_m^h .

$$H_t \Rightarrow a'_k \cdot \bar{x}_t \leq v_m^h + \mu \quad \forall t, k \in G_{max}^v \quad (3.52)$$

$$W_{kt} \Rightarrow a'_k \cdot \bar{x}_t \geq v_m^h - \mu \quad \forall t, k \in G_{min}^v \quad (3.53)$$

Problema na bateria (Ψ^b)

O problema na bateria, conforme descrito na Seção 3.2.2, indica uma falha em potencial como perda de potência ou superaquecimento. Todos os controles funcionam, mas se deve pousar a aeronave o mais rapidamente possível. Quando ocorrer essa falha, a função de avaliação utilizada não será a função objetivo (3.11), mas a função objetivo (3.12). A nova função possui a mesma bonificação que a função anterior em caso de queda em região bonificadora. Uma penalização exponencial ($2^{\frac{t-\tau_{max}}{10}}$) no tempo de voo é aplicada aos controles e ao tempo de voo. Dessa forma, apresenta melhor valor de objetivo as soluções que gastem menos *waypoints*.

Problema na superfície aerodinâmica tipo I (Ψ^{s1})

Nessa falha de superfície, a aeronave é capaz de girar somente para esquerda. A fim de descrever tal limitação, deve-se adicionar as restrições (3.56) e (3.57). No entanto, precisamos definir antes o parâmetro giro máximo permitido (N) seguindo a Equação (3.54). Assumimos que $\Gamma = 32$ e $\theta = 45^\circ = 0,785$ radianos, sendo assim, temos o valor de $N = 4$. O valor de N indicará o número de faces do polígono que poderemos saltar. O vetor para transformação de rotação em 90° (vetor perpendicular a W_{kt}) é dado pela Equação (3.55).

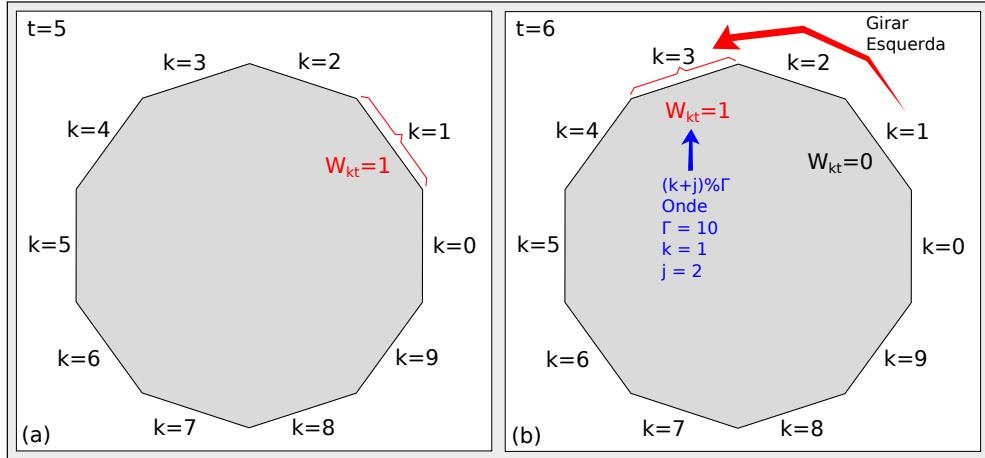
$$N = \frac{\Gamma}{2\pi} \cdot \theta \quad (3.54)$$

$$a'_k = \left[0 \ 0 \ 0 \ -\sin\left(\frac{2\pi k}{\Gamma}\right) \cos\left(\frac{2\pi k}{\Gamma}\right) \ 0 \right] \quad \forall k \in G_{min}^v \quad (3.55)$$

$$W_{kt} \wedge H_{t+1} \leq \sum_{j=0}^N W_{(k+j)\% \Gamma, t+1} \quad \forall t, k \in G_{min}^v \quad (3.56)$$

$$W_{k't} \Rightarrow \sum_{k \in G_{min}^v} a'_k \cdot (\bar{x}_{t+1} - \bar{x}_t) \geq 0 \quad \forall t, k' \in G_{min}^v \quad (3.57)$$

Figura 15 – Exemplo da modelagem da curva efetuada a esquerda.



Fonte: Elaborada pelo autor.

A Restrição (3.56) garante que o VANT curve somente para a esquerda. Caso W_{kt} esteja ativa, no próximo instante de voo, o valor dessa variável ($W_{(k+j)\% \Gamma, t+1}$) em algum dos j -ésimos períodos estará ativo também. $W_{(k+j)\% \Gamma, t+1}$ indica uma posição à esquerda da variável W_{kt} . Podemos entender o valor da variável W_{kt} ativa como sendo a direção k mais próxima ao deslocamento do VANT.

A Figura 15 ilustra o comportamento da variável W_{kt} em $t = 5$ e $t = 6$, usando um polígono de $\Gamma = 10$ faces. Os índices k representam as faces do polígono numeradas no sentido horário. Na Figura 15 (a), a variável W_{kt} ativada possui $k = 1$. A Figura 15 (b) mostra que o valor de W_{kt} ativado está sobre $k = 3$, que se encontra à esquerda de $k = 1$, fazendo assim o giro somente para esse lado.

A Restrição (3.57) complementa a primeira e também visa garantir que o VANT vire somente para a esquerda. Dado um W_{kt} ativo, o produto escalar entre o vetor perpendicular ao W_{kt} , representado por a'_k , e o deslocamento, representado por $\bar{x}_{t+1} - \bar{x}_t$, deve ser positivo. Isso garante que a aeronave gire para a esquerda.

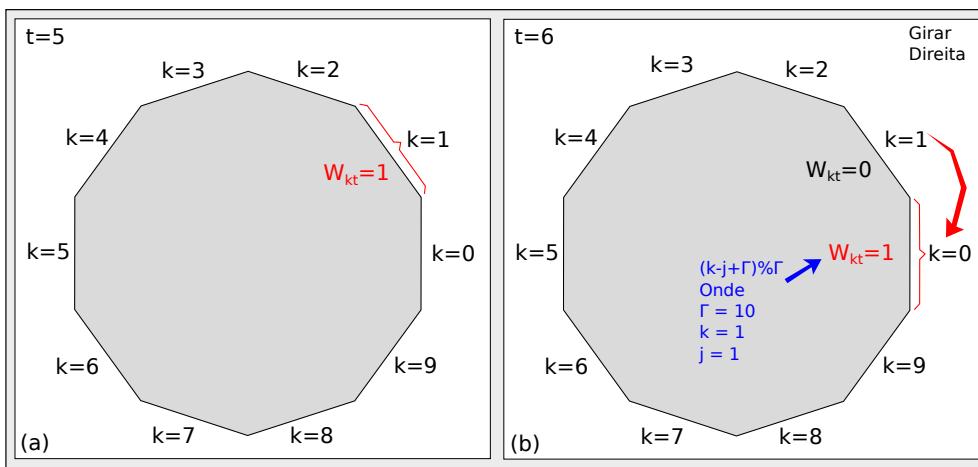
Problema na superfície aerodinâmica tipo 2 (Ψ^{s2})

Nesse problema, a aeronave é capaz de girar somente para direita. Logo, deve-se adicionar as seguintes restrições (3.58) e (3.59) ao modelo, de forma análoga ao problema Ψ^s ¹.

$$W_{kt} \wedge H_{t+1} \leq \sum_{j=0}^N W_{(k-j+\Gamma)\% \Gamma, t+1} \quad \forall t, k \in G_{min}^v \quad (3.58)$$

$$W_{k't} \Rightarrow \sum_{k \in G_{min}^v} a'_k \cdot (\bar{x}_{t+1} - \bar{x}_t) \leq 0 \quad \forall t, k' \in G_{min}^v \quad (3.59)$$

Figura 16 – Exemplo da modelagem da curva efetuada a direita.



Fonte: Elaborada pelo autor.

A Restrição (3.58) garante que o VANT curve somente para a direita, de forma semelhante à Restrição 3.56. A Figura 16 ilustra o comportamento da variável W_{kt} neste caso. Na Figura 16 (a), a variável W_{kt} ativada é a que possui $k = 1$. A Figura 16 (b) mostra que o valor de W_{kt} ativado está sobre $k = 0$, que se encontra a direita de $k = 1$, fazendo assim o giro somente para esse lado.

A Restrição (3.59) complementa a primeira, de maneira semelhante a Restrição 3.57, garantindo que o VANT virará somente para a direita. Dado um W_{kt} ativo, o produto escalar entre o vetor perpendicular ao W_{kt} , representado por a'_k , e o deslocamento, representado por $\bar{x}_{t+1} - \bar{x}_t$, deve ser negativo, ou seja, girar para a direita.

3.5 Conclusão

Este capítulo descreveu o problema abordado em maiores detalhes, onde um exemplo de planejamento de missão com situação crítica foi descrito e foram definidos os conjuntos de regiões mapeados e os tipos de situações críticas a serem tratados.

Baseado nesse problema, duas formulações matemáticas foram propostas: uma restrita e outra completa. A primeira formulação modela apenas o pouso sobre duas dimensões e considera que a altitude cai linearmente com o tempo. Essa formulação será utilizada como base para o

desenvolvimento dos métodos HG, AG e AGMP, considerando a incerteza da posição constante ao longo do tempo. A segunda formulação efetua o pouso para o planejamento da rota em três dimensões. Essa formulação será utilizada no método PLIM e trata o crescimento da incerteza da posição ao longo do tempo. Os dois modelos tratam a alocação do risco em sua formulação.



METODOLOGIA

“A imaginação é mais importante que a ciência, porque a ciência é limitada, ao passo que a imaginação abrange o mundo inteiro.”

Albert Einstein

4.1 Introdução

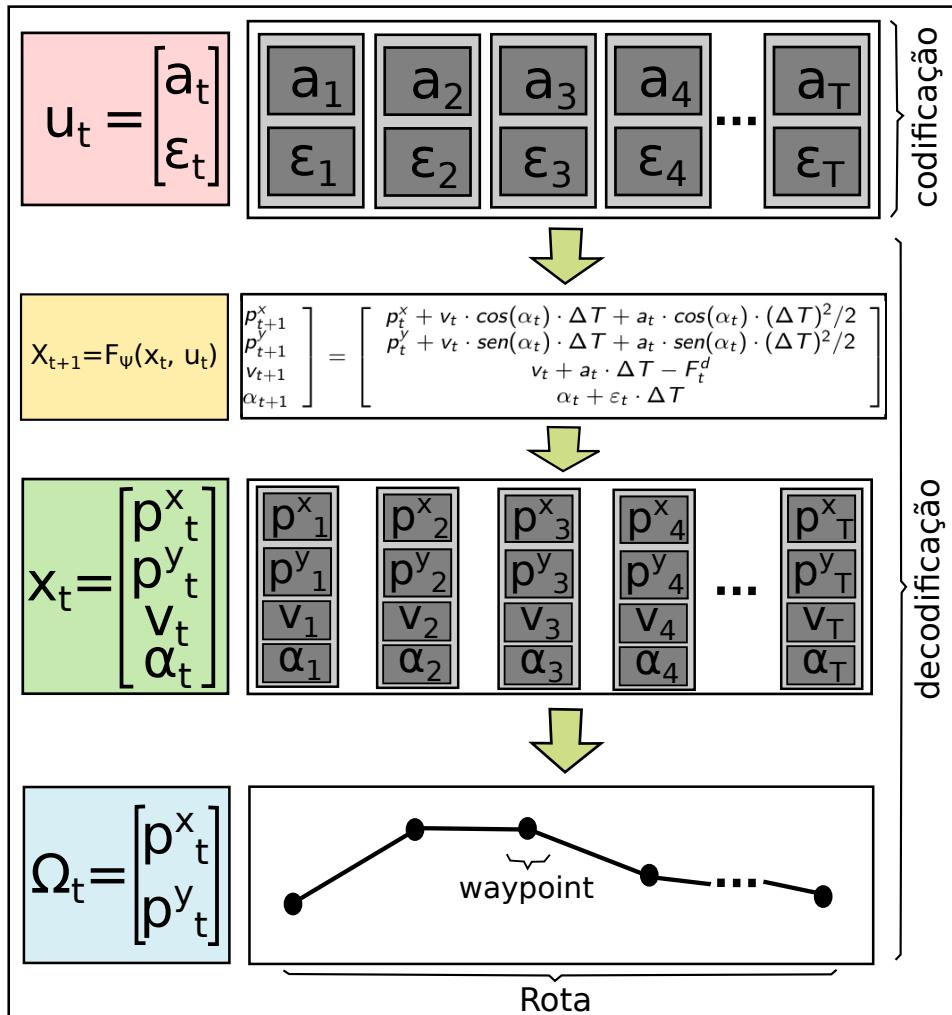
O presente capítulo da dissertação apresenta a metodologia seguida no desenvolvimento deste trabalho. Inicialmente é mostrada a codificação do problema bem como sua função objetivo. Quatro algoritmos planejadores de rotas propostos são descritos: Heurística Gulosa (HG), Algoritmo Genético (AG), Algoritmo Genético Multi-Populacional (AGMP) e Programação Linear Inteira Mista (PLIM). Por fim, é mostrada a arquitetura do sistema planejador, as estratégias de pouso, algumas considerações sobre o problema e o *framework* utilizado.

4.2 Codificação do Problema

A codificação do problema é uma etapa importante no desenvolvimento do método, pois uma boa representação da solução afeta diretamente a qualidade da rota gerada. Os métodos devem encontrar os controles da aeronave de forma a levá-la para um local de pouso seguro, minimizando danos. Para isso, uma codificação real foi definida, onde os controles são elementos de aceleração e variação angular suportados pelo VANT.

A solução do problema é codificada com valores reais para os controles $u_t = [a_t \ \varepsilon_t]'$ aplicados à aeronave, onde a_t e ε_t representam a aceleração e a variação angular (direção) no instante t . A codificação dos indivíduos é representada por vetores do tamanho do horizonte de planejamento T , ou seja, caso $T = 60$, o vetor teria tamanho 60 com 120 variáveis a serem determinadas pelo algoritmo planejador de rota. A Figura 17 exemplifica tal codificação.

Figura 17 – Codificação e decodificação dos controles na rota da aeronave utilizada na modelagem restrita.



Fonte: Elaborada pelo autor.

Tal codificação é traduzida como uma rota a partir da decodificação dos controles em um conjuntos de *waypoints* do VANT. A função de transição $F_\Psi(x_t, u_t)$ é usada para decodificar os controles. Essa função descreve o movimento de um VANT de asa fixa no espaço bidimensional.

Dado que x_t é uma variável aleatória ($x_t \sim \mathcal{N}(\bar{x}_t, \Sigma_t)$), os autores em [Blackmore, Ono e Williams \(2011\)](#) mostram que o próximo estado esperado (\bar{x}_{t+1}) depende apenas do estado esperado atual (\bar{x}_t) e dos controles nominais aplicadas (\bar{u}_t), assim a função de transição pode ser utilizada para calcular todos os próximos estados esperado ($\bar{x}_{t+1} = F_\Psi(\bar{x}_t, \bar{u}_t)$). No entanto, a incerteza Σ_t em todo estado esperado \bar{x}_t cresce a cada instante de tempo.

Os autores em [Ono, Williams e Blackmore \(2013\)](#) apresentam uma abordagem de controle *close-loop* para definir entradas de controle a partir de uma entrada de controle nominal \bar{u}_t . Nela é aplicada uma correção que diminui o crescimento da incerteza Σ_t . Na modelagem restrita, assumimos que a incerteza Σ_t a qualquer tempo t é constante ($\Sigma_t = Q$). Sabe-se que a incerteza no estado x_t a qualquer momento t é dada por $x_t \sim \mathcal{N}(\bar{x}_t, \Sigma_t)$, sendo possível calcular a

função de probabilidade $Pr(x_t \in Z_{\Phi}^i)$ usando uma tabela para a distribuição Gaussiana. A matriz de covariância utilizada neste trabalho é dada pela Equação (4.1), onde $\sigma = 10$ metros.

$$\Sigma_t = Q = \begin{bmatrix} \sigma^2 & 0 & 0 & 0 \\ 0 & \sigma^2 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \quad (4.1)$$

O processo de decodificação descrito retornará o conjunto Ω com todos os *waypoints* necessários para pousar o VANT como descrito na Restrição (3.2) da Seção 3.3. A Figura 17 mostra um exemplo da solução obtida, após a decodificação dos controles pela função de transição citada. É possível pousar a aeronave sem gastar todos os passos de tempo disponível, assim se pode pousar em um tempo K tal que $0 < K \leq T$.

4.2.1 Função Objetivo

A função objetivo utilizada pelos métodos propostos HG, AG e AGMP é baseada na Expressão (3.1) da modelagem matemática restrita do problema. Porém, foram adicionados mais alguns elementos que ajudam a trabalhar com as falhas associadas a aeronave. A Equação (4.2) descreve tal função objetivo.

Função objetivo:

$$fitness = f_{Pouso_{\Phi^b}} + f_{Pousos_{\Phi^p}} + f_{PousosEVoo_{\Phi^n}} + f_{Curvas} + f_{DistVANT_{\Phi^b}} + f_{Violout} + f_{\Psi^b} \quad (4.2)$$

Expressões usadas:

A Equação (4.3) define recompensa em caso de pouso em regiões bonificadoras.

$$f_{Pousos_{\Phi^b}} = -C_{\Phi^b} \cdot \sum_{i=1}^{|\Phi^b|} (Pr(x_K \in Z_{\Phi^b}^i)) \quad (4.3)$$

A Equação (4.4) define punição em caso de pouso em regiões penalizadoras.

$$f_{Pousos_{\Phi^p}} = C_{\Phi^p} \cdot \sum_{i=1}^{|\Phi^p|} (Pr(x_K \in Z_{\Phi^p}^i)) \quad (4.4)$$

A Equação (4.5) penaliza o pouso ou voo da aeronave sobre regiões não navegáveis.

$$f_{PousosEVoo_{\Phi^n}} = C_{\Phi^n} \cdot \max \left(0, 1 - \Delta - Pr \left(\bigwedge_{t=0}^K \bigwedge_{i=1}^{|\Phi^n|} x_t \notin Z_{\Phi^n}^i \right) \right) \quad (4.5)$$

A Equação (4.6) prioriza rotas que evitem fazer curvas desnecessárias.

$$f_{Curvas} = \frac{1}{|\epsilon_{max}|} \cdot \sum_{t=0}^K |\epsilon_t| \quad (4.6)$$

A Equação (4.7) dá mais chance a rotas com menores distâncias das regiões bonificadoras.

$$f_{DistVANT_{\Phi^b}} = menorDist(\bar{x}_K, \Phi^b) \quad (4.7)$$

Se a aeronave tem velocidade final maior do que o seu valor mínimo, não ocorre de fato um pouso. Dessa maneira, a Equação (4.8) evita rotas em que o VANT não consegue pousar, mesmo que atinja uma região bonificadora.

$$f_{Violout} = \begin{cases} C_{\Phi^b} & , \text{caso } v_K - v_{min} > 0 \\ 0 & , \text{caso contrário} \end{cases} \quad (4.8)$$

Se houver um problema na bateria, a Equação (4.9) é adicionada à função de *fitness*. O objetivo é reduzir o tempo de voo, assim a aeronave buscará regiões para pouso mais próximas de sua localização no momento onde ocorreu a pane.

$$f_{\Psi^b} = \begin{cases} C_{\Phi^b} \cdot 2^{\frac{(K-T)}{10}} & , \text{caso } \Psi^* = \Psi^b \\ 0 & , \text{caso contrário} \end{cases} \quad (4.9)$$

4.3 Heurística Gulosa

A Heurística Gulosa (HG) é uma estratégia de pouso desenvolvida com o objetivo de ser um método simples o suficiente para chegar a uma solução dentro de um curto espaço de tempo. Essa abordagem pode ser útil para o pouso diante de situações críticas, quando uma tomada de decisão rápida é necessária e uma solução viável se torna mais interessante do que a obtenção de soluções próximas ao ótimo. O Algoritmo 1 descreve a HG mostrando suas entradas, saídas e funcionamento.

O HG possui como entrada o mapa contendo informações do cenário, enquanto a saída é a melhor rota para pouso encontrada. O funcionamento básico dessa heurística é gerar uma solução candidata para cada uma das regiões bonificadoras ($mapa.|\Phi^b|$). Essa solução candidata (gerada pelo método `inicializar(...)` na linha 4) é determinada rotacionando-se a aeronave até que a mesma esteja direcionada para uma região bonificadora ($mapa.Z_{\Phi^b}^i$). Em seguida, calcula-se uma trajetória em linha reta até que ela pouse nessa região. A rotação ocorre para o lado com menor ângulo entre o VANT e uma linha reta para uma região bonificadora. Isto é ilustrado na Figura 18 através dos ângulos $\lambda_1 < \lambda_1'$ para a região b_1 , onde a rotação ocorre em direção ao ângulo λ_1 .

Algoritmo 1: Heurística Gulosa.

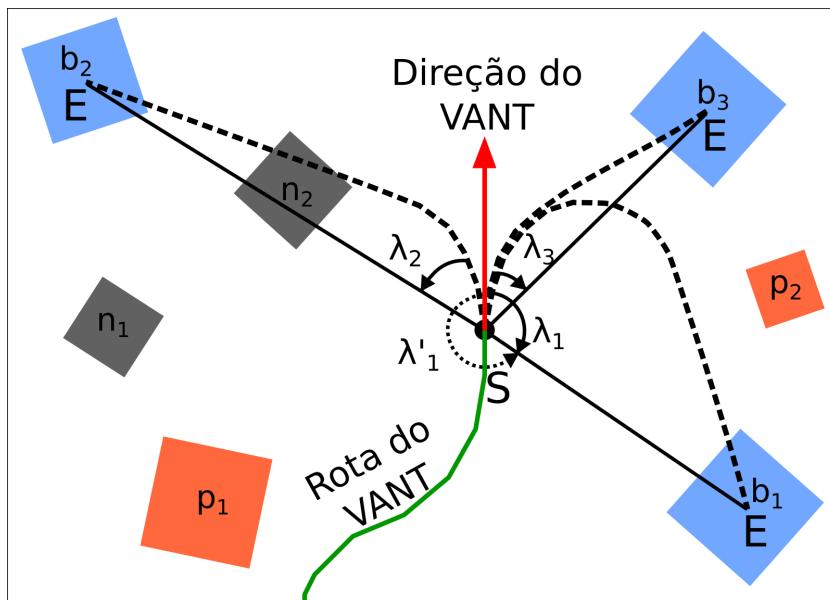
Entrada: Mapa: *mapa*
Saída: RotaPousos: *melhorRota*

```

1 início
2   RotaPousos rota[] ← instância RotaPousos() [mapa.|Φb|];
3   para i = 1 até mapa.|Φb| faz
4     inicializar(rota[i], mapa.ZiΦb);
5     avaliar(rota[i]);
6   RotaPousos melhorRota ← obtemMelhorRota(rota);
7   retorna melhorRota;

```

Figura 18 – Rotas candidatas usando a estratégia Heurística Gulosa.



Fonte: Elaborada pelo autor.

As soluções candidatas são avaliadas de acordo com o seu desempenho na Equação (4.2). A melhor trajetória definida entre todas as regiões bonificadoras é retornada pela HG. Na Figura 18, a melhor solução é a que pousa na região b_3 que é a região mais próxima ao VANT. A solução que pousa em b_1 tem desempenho inferior, visto que possui uma distância maior; já a solução que pousa em b_2 viola a restrição de não navegabilidade, por isso tem baixa qualidade.

A estratégia gulosa pode ser utilizada como um operador de inicialização para criar indivíduos no AGMP. Nesse caso, uma região bonificadora é selecionada aleatoriamente pelo HG e uma trajetória é construída. Os controles definidos para essa trajetória serão codificados como um indivíduo no AGMP.

É importante lembrar que esse algoritmo na etapa de inicialização não leva em conta a condição de não navegabilidade. Caso a rota gerada para uma determinada região viole tal condição, ela terá um valor alto de penalização de forma a torná-la inviável. A HG implementada

possui garantia de convergência, no entanto, não possui garantia de otimalidade da solução. Esse método possui comportamento determinístico e retorna em geral soluções dentro de um curto espaço de tempo na ordem de 0,07 segundo.

4.4 Algoritmo Genético

Os métodos de computação evolutiva são metaheurísticas baseadas em população de soluções. As técnicas para gerar e testar soluções, explorando e intensificando a busca no espaço de soluções do problema, são inspiradas na teoria da evolução de Darwin. Novas soluções são criadas e adicionadas à população pela aplicação de operadores de seleção, recombinação e mutação. Tais métodos têm sido aplicados em vários problemas de otimização nas mais diversas áreas [Jong \(2006\)](#), [Jin \(2011\)](#), onde os principais métodos são os Algoritmos Genéticos (AGs), Programação Genética (PG), Estratégia Evolutiva (EE) e Programação Evolutiva (PE) ([EIBEN; SMITH, 2003](#)). No contexto desta dissertação, pretende-se desenvolver um método para o planejamento de rotas em situação crítica baseado em AGs.

AGs foram inicialmente propostos por [Holland \(1975\)](#) e seguem o princípio da seleção natural. O conceito de seleção natural foi proposto por Charles Darwin em 1859 em seu livro “A Origem das Espécies”. A ideia principal consiste em que espécies com características mais adaptadas ao meio onde vivem tendem a sobreviver, enquanto espécies pouco adaptadas tendem a ser extintas.

Um AG inicialmente define uma população de indivíduos, onde cada indivíduo é uma representação (codificação) de uma solução do problema em questão. Essa população tem seus indivíduos inicializados de forma aleatória. Um laço (estrutura de repetição) é executado, onde a população de indivíduos é evoluída durante um certo número de gerações ou outro critério de parada. O processo de evolução começa pela seleção de indivíduos da população para realização de cruzamento ou *crossover*, onde novos indivíduos são gerados. Em seguida, os novos indivíduos tem certa probabilidade de sofrer mutações em sua codificação. Finalmente, um segundo processo de seleção define quais indivíduos, entre os existentes e aqueles gerados, serão selecionados para a próxima geração.

Um Algoritmo Genético (AG) foi desenvolvido como alternativa à HG proposta inicialmente. Esse método, ao contrário da HG, é não determinístico e, assim como a HG, não garante a otimalidade da solução. O [Algoritmo 2](#) descreve o seu funcionamento.

Inicialmente cria-se a população de rotas que é avaliada conforme Expressão (4.2). Em seguida, o processo evolutivo ocorre enquanto não houver convergência. Para isso, um total de $taxaCross \times tamPop$ indivíduos são gerados através dos operadores *crossover* e mutação a cada repetição. A seleção dos indivíduos para cruzamento utiliza o operador torneio. A convergência ocorre quando nenhum novo indivíduo é inserido na população. Caso tenha convergido, a população será reiniciada sendo mantido apenas o melhor indivíduo (melhor solução). Esse

Algoritmo 2: Algoritmo Genético.

Entrada: Critério: *critérioDeParada*; Inteiro: *tamPop*, *taxaCross*; Mapa: *mapa*

Saída: RotaPouso: *melhorRota*

1 **início**

```

2   criarPopulação(rotas);
3   inicializar(rotas);
4   avaliar(rotas, mapa);
5   enquanto nãoAtingir(critérioDeParada) faça
6     enquanto nãoConvergir() faça
7       para i = 1 até taxaCross × tamPop faça
8         selecionar(pais);
9         filho ← crossover(pais);
10        mutação(filho);
11        avaliar(filho, mapa);
12        adicionar(filho);
13     reiniciar(rotas);
14   RotaPouso melhorRota ← obtemMelhorRota(rotas);
15   retorna melhorRota;

```

processo é repetido até um certo número de avaliações do *fitness* ser atingido (critério de parada).

A seguir são descritos os operados de inicialização, *crossover* e mutação utilizados tanto no AG quanto no AGMP.

Operadores de Inicialização:

Os operadores de inicialização criados para as estratégias evolutivas são:

- **Inicialização Aleatória:** a inicialização aleatória gera valores com distribuição uniforme representado por $U[inf; sup]$, onde *inf* e *sup* representam, respectivamente, o valor inferior e superior da distribuição. Assim, os valores da aceleração e variação angular são dados por $a_t \in U[a_{min}; a_{max}]$ e $\varepsilon_t \in U[\varepsilon_{min}; \varepsilon_{max}]$.
- **Inicialização Curva Curta:** o operador curva curta gera valores para $a_t \in U[a_{min}; a_{max}]$, mas gera valores reduzidos para a variação angular com $\varepsilon_t \in U[\frac{1}{4} \cdot \varepsilon_{min}; \frac{1}{4} \cdot \varepsilon_{max}]$. Esse operador dá maior chance de se aplicar controles mais suaves para as mudanças de direção na aeronave. O valor $\frac{1}{4}$ foi determinado de forma arbitrária.
- **Inicialização Aceleração Curta:** o operador aceleração curta faz o contrário, ou seja, gera $\varepsilon_t \in U[\varepsilon_{min}; \varepsilon_{max}]$ e $a_t \in U[\frac{1}{4} \cdot a_{min}; \frac{1}{4} \cdot a_{max}]$. Esse operador dá maior chance de se aplicar controles de acelerações mais suaves a aeronave, fazendo a aeronave pousar mais rapidamente.
- **Inicialização Gulosa:** a inicialização gulosa aplica a HG, explicada na Seção 4.3, para gerar um conjunto de controles que garantem atingir regiões bonificadoras.

Operadores de Crossover:

Os operadores de *crossover* implementados para lidar com a codificação real, descritos em Eshelman e Schaffer (1992), Michalewicz e Schoenauer (1996), Eiben e Smith (2003), Herrera, Lozano e Sánchez (2005) e utilizados pelo AG e AGMP, são listados a seguir:

- **Crossover Média:** os controles do indivíduo filho são formados pela média aritmética dos controles dos pais.
- **Crossover Aritmético:** os controles do indivíduo filho são formados pela média ponderada dos controles dos pais.
- **Crossover Geométrico:** os controles do indivíduo filho são formados pela média geométrica dos controles dos pais.
- **Crossover OX:** os controles do indivíduo filho são compostos pelos controles de ambos os pais, tomados aleatoriamente.
- **Crossover BLX- α :** os controles do indivíduo filho são submetidos a expressão: $u_i^{filho} = u_i^{pai1} + \beta(u_i^{pai1} - u_i^{pai2})$, onde $i = 1, 2, \dots, T$ e $\beta = U(-\alpha, 1 + \alpha)$.

Operadores de Mutação:

Os operadores de mutação, descritos em Michalewicz e Schoenauer (1996), Eiben e Smith (2003), Herrera, Lozano e Sánchez (2005), são listados a seguir:

- **Mutação Uniforme:** este operador sorteia aleatoriamente algum controle e aleatoriamente estabelece novos valores para aceleração e direção.
- **Mutação Limite:** o operador Limite aleatoriamente sorteia um par de controles. Em seguida, um valor 0 ou 1 é aleatoriamente escolhido, fazendo com que as variáveis dos controles seja ajustadas no seu limite máximo ou mínimo, respectivamente.
- **Mutação Creep:** este operador adiciona ao controle um pequeno valor aleatório, causando uma pequena perturbação no controle.

Os diversos tipos de operadores de inicialização, recombinação e mutação são selecionados de forma aleatória, cada vez que o respectivo operador é executado no AG e no AGMP.

4.5 Algoritmo Genético Multi-Populacional

Uma outra estratégia evolutiva utilizada neste trabalho é o AGMP. Esse método é similar ao AG descrito na seção anterior, entretanto, ele trabalha com multi-populações, migrações entre

Algoritmo 3: Algoritmo Genético Multi-Populacional.

Entrada: Critério: *critérioDeParada*; Inteiro: *numPopulações*, *numIndivíduos*, *taxaCross*; Mapa: *mapa*

Saída: RotaPouso: *melhorRota*

```

1 início
2   repita
3     para i = 1 até numPopulações faça
4       para j = 1 até numIndivíduos faça
5         inicializar(pop(i).ind(j));
6         avaliar(pop(i).ind(j), mapa);
7         estruturar(pop(i));
8       repita
9         para j = 1 até taxaCross × numIndivíduos faça
10        selecionar(pais);
11        filho ← crossover(pais);
12        mutação(filho);
13        avaliar(filho, mapa);
14        adicionar(filho, pop(i));
15        estruturar(pop(i));
16        até convergir(pop(i));
17     para i = 1 até numPopulações faça
18       migrar(pop(i));
19     até atingir(critérioDeParada);
20     RotaPouso melhorRota ← obtemMelhorRota(pop);
21   retorna melhorRota;

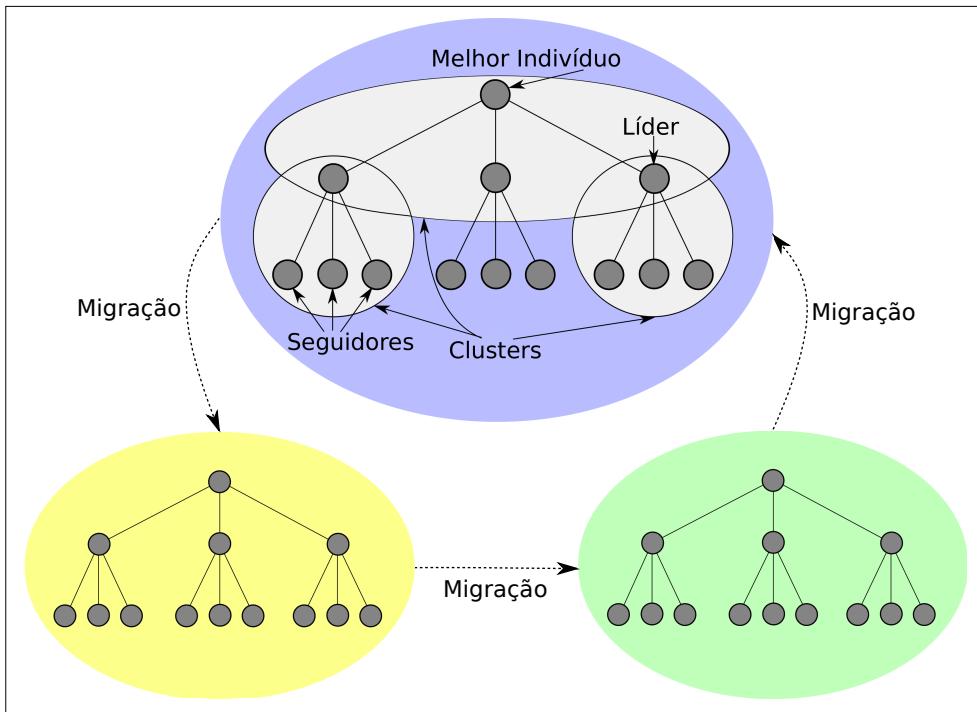
```

as populações e indivíduos organizados em forma de árvore. O Algoritmo 3 descreve o AGMP proposto que é baseado no trabalho de Toledo *et al.* (2009).

A população é inicializada e todos os indivíduos são avaliados (conforme Expressão (4.2)). Em seguida, esses indivíduos são estruturados em árvore como ilustrado pela Figura 19. A hierarquia é representada pela posição dos indivíduos (nós) nos *clusters* da árvore. Há um líder em cada *cluster* que é o indivíduo com melhor valor de *fitness*. O melhor indivíduo da população é representado pelo nó raiz e os piores indivíduos são os nós folha.

O processo de evolução seleciona aleatoriamente dois indivíduos para reprodução que são sempre um nó líder e um dos seus seguidores. O novo indivíduo gerado substitui o pior pai, caso ele possua um melhor valor de *fitness*. Após a nova inserção de indivíduos, a população é hierarquicamente reestruturada, onde os melhores indivíduos se tornam líderes em seus *clusters*. A população converge se nenhum indivíduo for inserido em *taxaCross × numIndividuos* tentativas. Quando todas as populações convergem, um operador de migração é executado e elas são reinicializadas, exceto pelo melhor indivíduo de cada população e os indivíduos que migraram. O operador migração envia uma cópia do melhor indivíduo da população *i* para a população

Figura 19 – Estrutura hierárquica em árvore usada no AGMP.



Fonte: Elaborada pelo autor.

$i + 1$. O critério de parada é dado pelo número de avaliações do *fitness*. Ao alcançar o critério de parada, o algoritmo retorna a melhor solução (rota) encontrada.

Tanto o AG quanto o AGMP implementados não possuem garantia de convergência, sendo assim, a solução é retornada após o método atingir um número de avaliações. Por se tratar de uma metaheurística, tais métodos não garantem a otimalidade da solução e sua execução é não determinística. De forma geral, tanto o AG quanto o AGMP retornam soluções relativamente rápidas na ordem de 1 segundo.

4.6 Programação Linear Inteira Mista

Esta seção apresenta o método usado para resolver o problema a partir do modelo baseado em Programação Linear Inteira Mista (PLIM). O *solver* de otimização usado para resolver esse modelo foi IBM ILOG CPLEX. Internamente esse *solver* executa um algoritmo *Branch and Cut* (B&C) para resolver o modelo. Esse método caracteriza-se por retornar soluções ótimas para o modelo e ser determinístico, no entanto, suas soluções demandam alto tempo de processamento. Primeiramente, é apresentada a forma como o modelo de PLIM foi implementado. Finalmente, uma estratégia para resolver o problema usando rotas pré-processadas é proposta.

É importante ressaltar que o método baseado em PLIM não será embarcado em aeronave real e nem será testado em simulações, devido ao seu alto custo computacional. A definição e utilização de tal método visa fornecer uma comparação para os outros métodos propostos. Por

exemplo, a resolução exata do problema permitirá fornecer um limitante superior para a taxa de sucesso no pouso da aeronave.

O Algoritmo 4 descreve a implementação desse modelo. A função objetivo é definida no campo *Função Objetivo*, através da função *minimizar(·)* que será minimizada pelo *solver*. O argumento utilizado nas funções indica o número da função objetivo. Dependendo do tipo de situação crítica tratada, a função objetivo poderá ser alterada como será descrito a seguir. No campo *Sujeito a* são adicionadas as restrições do modelo pelo comando *adicionar(·)*. No campo *Situação Crítica*, estão as mudanças que devem ser acrescentadas ao modelo de forma a simular cada uma das respectivas falhas. Por exemplo, na falha do motor (Ψ^m), deve-se adicionar ao modelo as restrições (3.50) a (3.53). Por outro lado, a falha da bateria (Ψ^b) demanda a remoção da função objetivo (3.11), através do comando *remover(·)*, e a inclusão da função objetivo (3.12) pelo comando *minimizar(·)*. Por fim, o *solver* resolve o modelo PLIM e o método retorna a melhor rota de pouso emergencial.

Os experimentos serão realizados a partir de dois modelos. O primeiro modelo, chamado de PLIM1, considera o Algoritmo 4 usando a função objetivo (3.11). O segundo modelo, chamado de PLIM2, assume a função objetivo (3.12). Essa comparação permitirá estudar o comportamento das soluções ao se enfatizar a minimização do tempo gasto para realizar o pouso via função objetivo (3.12).

Os resultados computacionais descritos na próxima seção revelam que soluções exatas obtidas a partir do modelo de PLIM2 gastam em média entre 32 a 444 segundos, ou seja, alto custo computacional. Dependendo do tipo de aplicação, esse tempo de processamento para replanejamento de rota pode inviabilizar a realização do pouso emergencial. Por isso, será descrita uma abordagem que permite utilizar a solução exata, mas através do processamento prévio de um subconjunto de rotas alternativas para pouso.

A Figura 20 exemplifica um cenário no qual o VANT parte do ponto *Início* e executa sua missão que termina no ponto *Fim* (rota na cor verde). No momento do planejamento da rota principal, rotas emergenciais também são previamente calculadas. Para isso, a rota principal é subdividida em vários pontos-chave. Para cada ponto-chave, uma rota de pouso emergencial é calculada para cada possível situação crítica (rotas na cor vermelho). Essas rotas ficarão armazenadas na aeronave e, caso uma determinada falha ocorra, a aeronave passa a seguir os *waypoints* da rota de pouso emergencial mais próxima desse local. A Figura 20 ilustra como seriam os possíveis casos de rota de pouso para as situações críticas Ψ^m , Ψ^b , Ψ^{s1} e Ψ^{s2} . É importante reparar que parte das rotas poderão levar o VANT até uma região bonificadora e outras rotas, no entanto, apenas conseguirão pousar na região remanescente.

Algoritmo 4: Algoritmo do Modelo PLIM.

Input: Modelo: *modelo*; TipoModelo: *tipoModelo*; SituaçãoCrítica: *tipoFalha*; Mapa: *mapa*

Output: RotaPousos: *melhorRota*

```

1 begin
2   Função Objetivo:
3   {
4     caso (tipoModelo = TipoModelo.PLIM1)
5       modelo.minimizar(3.11)
6     caso (tipoModelo = TipoModelo.PLIM2)
7       modelo.minimizar(3.12)
8   }
9   Sujeito a:
10  {
11    modelo.adicionar(3.13 - 3.20)
12    modelo.adicionar(3.22 - 3.24)
13    modelo.adicionar(3.26 - 3.27)
14    modelo.adicionar(3.29 - 3.34)
15    modelo.adicionar(3.39 - 3.40)
16    modelo.adicionar(3.42 - 3.49)
17  }
18  Situação Crítica:
19  {
20    caso (tipoFalha =  $\Psi^m$ )
21      modelo.adicionar(3.50 - 3.53)
22    caso (tipoFalha =  $\Psi^b$ )
23      modelo.remover(3.11)
24      modelo.minimizar(3.12)
25    caso (tipoFalha =  $\Psi^{s1}$ )
26      modelo.adicionar(3.56 - 3.57)
27    caso (tipoFalha =  $\Psi^{s2}$ )
28      modelo.adicionar(3.58 - 3.59)
29  }
30  modelo.resolve()
31  melhorRota ← modelo.obtemMelhorRota()
32  return melhorRota

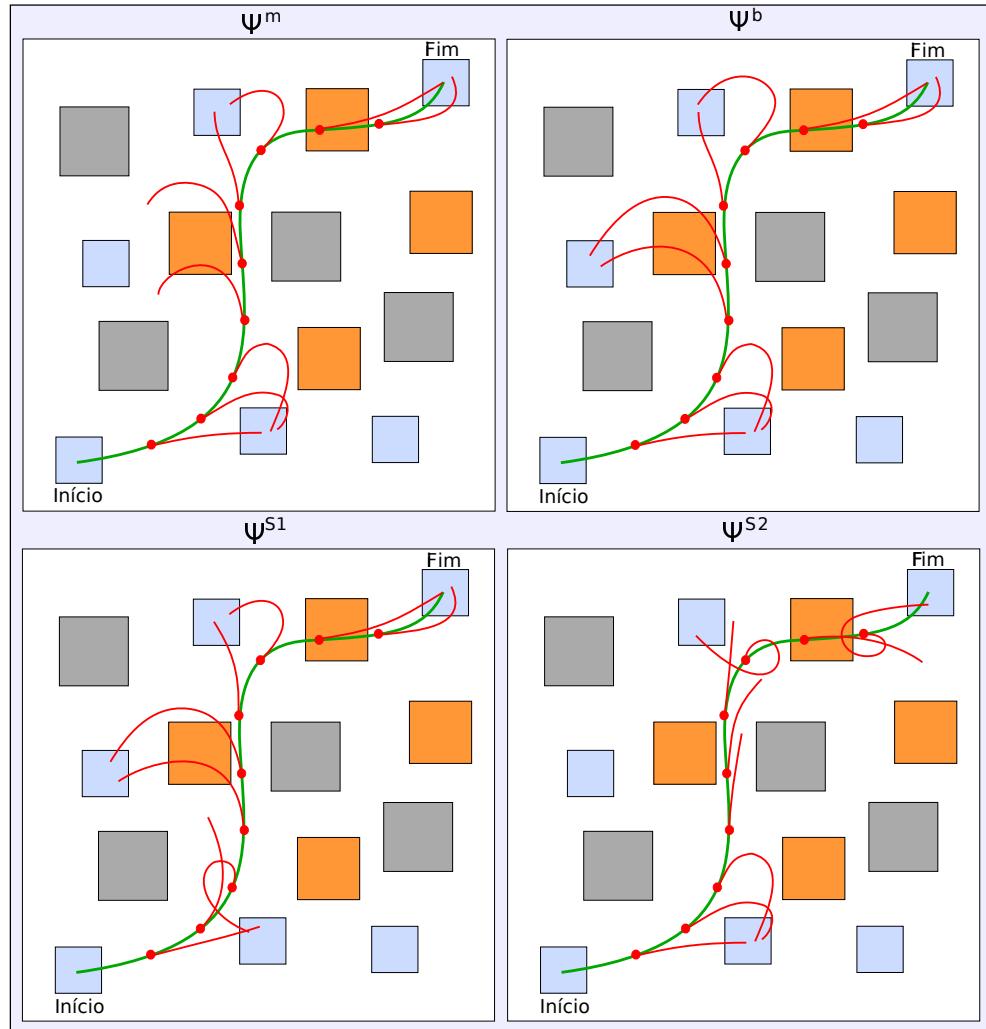
```

4.7 Arquitetura Proposta

Esta seção mostra a arquitetura proposta para o algoritmo planejador de rota de pouso com todas as suas entradas e saídas. Neste trabalho quatro métodos planejadores de rotas foram desenvolvidos que são: HG, AG, AGMP e PLIM. A arquitetura busca encontrar rotas seguras para pouso e utiliza como entrada algumas informações fornecidas pelo sistema supervisor da aeronave IFA. Isso resulta em uma saída contendo um conjunto de *waypoints*, local do pouso, probabilidade de pouso e valor do objetivo. A Figura 21 apresenta essa arquitetura.

O sistema IFA acessa uma série de informações como sensores da aeronave, situações críticas, mapa, modelo de dinâmica do VANT, entre outras informações relevantes. O algoritmo planejador é acionado no momento em que o IFA detecta a falha e usa os parâmetros de entrada

Figura 20 – Esquema das rotas emergenciais para a missão definida para as falhas Ψ^m , Ψ^b , Ψ^{s1} e Ψ^{s2} , respectivamente.

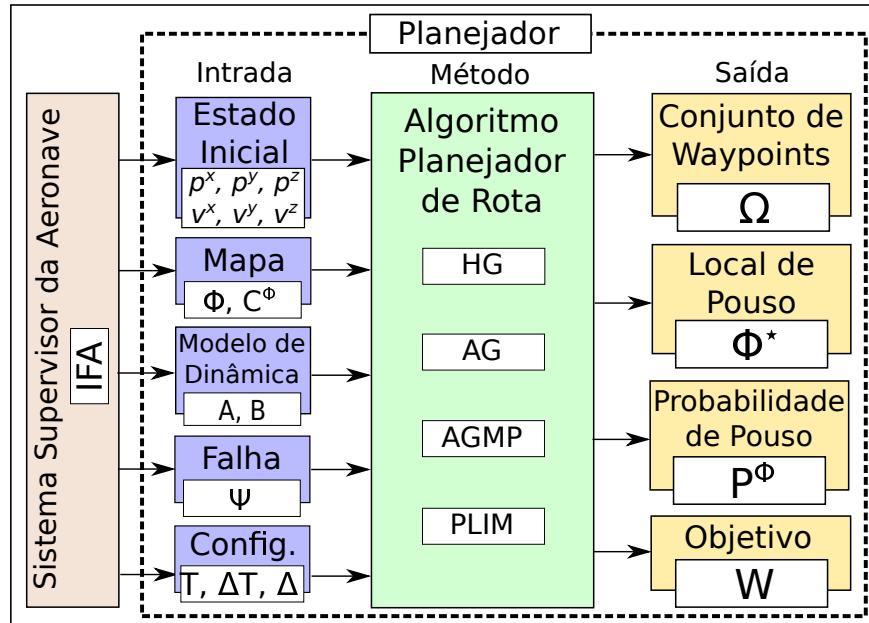


Fonte: Elaborada pelo autor.

descritos a seguir:

1. **Estado Inicial:** o estado inicial do VANT contém dados como posição cartesiana (p^x , p^y , p^z), velocidade (v^x , v^y , v^z), que serão obtidas por sensores GPS, e velocímetro, quando integrados a um sistema real;
2. **Mapa:** o mapa (Z_Φ e C_Φ) foi obtido através do IFA que tem acesso a todas as informações do cenário mapeado;
3. **Modelo de Dinâmica:** o modelo de dinâmica descreve a forma como o VANT se desloca no espaço. É dado através das matrizes A e B do modelo, ou ainda, através da função de transição de estados F_Ψ ;
4. **Falha:** a falha Ψ é obtida pelo sistema IFA através do monitoramento da aeronave;

Figura 21 – Arquitetura do planejador de rota de pouso proposto.



Fonte: Elaborada pelo autor.

5. **Config.:** configurações do planejador obtidas pelo sistema IFA, por exemplo, tempo máximo de voo (T), discretização do tempo (ΔT) e a probabilidade da aeronave violar regiões não navegáveis (Δ).

O algoritmo planejador usará esse conjunto de informações de entrada e obterá como saída as seguintes informações:

1. **Conjunto de Waypoints (Ω):** o sistema planejador retornará a melhor rota encontrada, representada por Ω , para a situação crítica ocorrida;
2. **Local de Pouso (Φ^*):** o planejador estimará o local de pouso mais provável, que é representado por Φ^* ;
3. **Probabilidade de Pouso (P^Φ):** o planejador estimará a probabilidade de pouso sobre a região Φ^* ;
4. **Objetivo (\mathcal{W}):** o algoritmo planejador calculará um custo associado a queda da aeronave (objetivo ou *fitness*) para a situação crítica ocorrida, representado por \mathcal{W} .

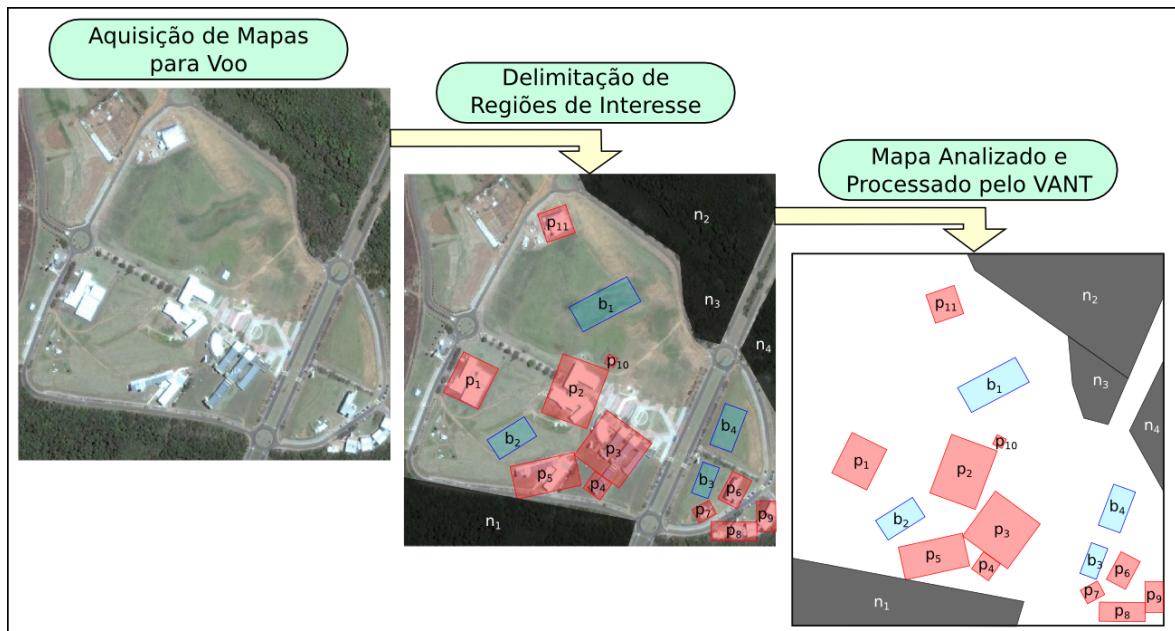
4.8 Geração Manual de Mapas

A estratégia usada para construção manual de mapas é descrita nesta seção. Inicialmente, a modelagem das regiões bonificadoras (Φ^b), regiões penalizadoras (Φ^p) e regiões não navegáveis

(Φ^n) foi definida como regiões convexas. A construção de regiões não convexas é feita através da superposição de duas ou mais regiões convexas.

A Figura 22 mostra as etapas seguidas na construção dos mapas. A primeira etapa é a aquisição dos mapas e, nesse caso, utilizou-se um mapa obtido pelo *Google Earth* (Campus II da USP de São Carlos) que representa a região onde o VANT deve sobrevoar. A próxima etapa a ser executada é a definição de regiões de interesse nesse mapa. Sendo assim, foram delimitadas regiões penalizadoras, representando áreas construídas; regiões bonificadoras, representadas por regiões planas e sem construções; regiões não navegáveis, delimitadas sobre as florestas densas; região restante, considerada como sendo as demais áreas remanescentes do mapa. Na última etapa, tem-se a eliminação de detalhes da imagem do cenário de voo. Dessa forma, apenas as regiões mapeadas (polígonos) serão utilizados pela aeronave. Isso economiza memória e processamento, evitando um processamento de imagem como foi realizado nos trabalhos de Saripalli, Montgomery e Sukhatme (2002), Li (2013), Zhao e Pei (2012), Kim *et al.* (2013), Hermansson *et al.* (2010), Masuko *et al.* (2010).

Figura 22 – Estratégia para construção manual de mapas contendo regiões de interesse.



Fonte: Elaborada pelo autor.

Acima foi apresentado um exemplo, com base na Figura 22, de como poderia ser gerado o mapa para a missão do VANT. Entretanto, o processo de construção desse mapa é bastante lento por ser totalmente manual. A seguir será descrita uma forma automatizada de geração de mapas que também será utilizada neste trabalho.

4.9 Geração Automática de Mapas

O conjunto de mapas (cenários) a serem utilizados nas simulações e a forma como são gerados é uma etapa importante neste trabalho. Esses cenários serão usados para efetuar a validação do planejador de rotas de pouso. O gerador automático de mapas implementado será discutido nesta seção.

Inicialmente, buscando atingir uma maior compreensão de como os cenários foram criados, algumas categorias de mapas possíveis são definidas. Nesta etapa inicial, não foi considerada a topologia do terreno com relevos 3D, ou seja, considera-se apenas cenários com topologias planas 2D.

4.9.1 Classificação dos Mapas

O gerador de mapas é capaz de criar cenários com diferentes níveis de dificuldades e áreas de cobertura. A seguir, são apresentadas duas classificações criadas para os mapas. Um dos principais fatores que caracteriza a dificuldade do mapa é a proporção entre os diferentes tipos de regiões. Três níveis de dificuldade são definidos.

Nível de Dificuldade dos Mapas:

1. *Mapa Fácil*: Esta categoria de mapa é considerada a mais simples de ser resolvida. Neste tipo de cenário, pretende-se colocar um número maior de regiões do tipo bonificadora (Φ^b) (facilitando assim o pouso nessas regiões), uma quantidade mediana do tipo penalizadora (Φ^P) e poucas não navegáveis (Φ^n).
2. *Mapa Normal*: Nesta categoria de mapa, a dificuldade é considerada mediana. Mapas dessa categoria foram projetados buscando um equilíbrio na proporção das regiões do tipo Φ^n , Φ^P e Φ^b .
3. *Mapa Difícil*: Esta categoria é a que possui a maior complexidade de solução. Há um nível maior de regiões do tipo Φ^n (dificultando as soluções), um nível mediano do tipo Φ^P e poucas Φ^b .

Neste trabalho, quando se afirma que um mapa possui um nível maior de regiões de determinado tipo, significa que ela aparece em maior quantidade naquele mapa.

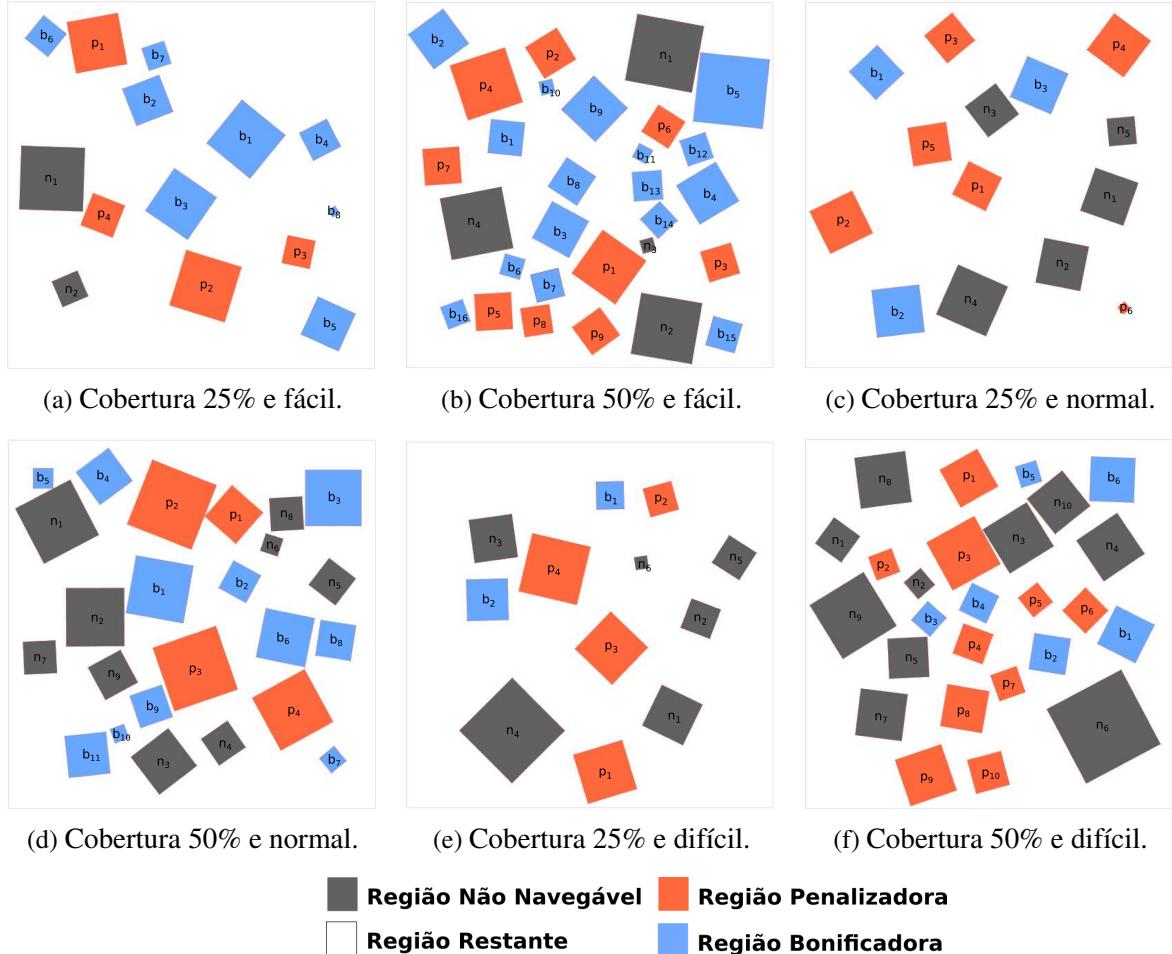
Uma outra classificação considerada nos cenários está relacionada às áreas de cobertura (densidade das regiões). A área de cobertura representa uma parcela da área total do mapa.

Área de Cobertura dos Mapas:

1. *Cobertura 25%*: Esta categoria de mapa é considerada mais esparsa, ou seja, possui um nível menor de área coberta em relação ao total. Isso retrata um cenário onde há pouca informação a respeito das regiões de interesse.

2. *Cobertura 50%:* Este tipo de mapa é considerada mais denso, já que possui um nível maior de informações sobre a área de cobertura. Em geral, representa um cenário com muitas regiões de interesse mapeadas.

Figura 23 – Exemplos de mapas gerados automaticamente.



Fonte: Elaborada pelo autor.

A Figura 23 exemplifica um total de seis tipos de mapas gerados de acordo com as combinações das classificações descritas anteriormente.

4.9.2 Algoritmo de Geração de Mapas

O algoritmo para geração automática de mapas implementado será descrito nesta seção. O Algoritmo 5 mostra, de forma geral, que a geração dos mapas aleatórios é feita sem interseção de regiões diferentes. Esse algoritmo é baseado no trabalho de Blackmore, Ono e Williams (2011). Por isso, foi definido que os polígonos seriam inicialmente apenas quadrados por questões de maior simplicidade, mas o gerador poderia ser facilmente modificado para polígonos em geral.

Algoritmo 5: Algoritmo Gerador Automático de Mapas.

Entrada: Numérico: $taxaOcupaçãoÁrea$, $dimXMap$, $dimYMap$

Saída: Mapa: $mapa$

1 **início**

```

2   |   listaPolígono ← ∅;
3   |   ocupação ← 0;
4   |   enquanto ( $ocupação < taxaOcupaçãoÁrea$ ) faça
5   |     repita
6   |       posX ← posXRegiãoAleatório();
7   |       posY ← posYRegiãoAleatório();
8   |       comprimento ← comprimentoRegiãoAleatório();
9   |       ângulo ← ânguloRegiãoAleatório();
10  |       polígono ← criarPolígono(posX, posY, comprimento, ângulo);
11  |       até (nãoTerIntercessão(listaPolígono, polígono));
12  |       listaPolígono.adiciona(polígono);
13  |       ocupação ← ocupação + comprimento2 / ( $dimXMap \times dimYMap$ );
14  |
15  |   mapa ← criarMapa(listaPolígono);
  |   retorna mapa;
```

Inicialmente o algoritmo recebe como entrada a taxa de ocupação da área total e as dimensões x e y do mapa. A saída do mesmo é um mapa aleatório gerado. Uma lista de polígonos que representará o cenário é criada. Em seguida, a variável $ocupação$ guardará a taxa de ocupação atual do mapa, iniciada como zero quando o ambiente está vazio. Logo após, inicia-se uma estrutura de repetição até que a taxa de ocupação desejada seja atingida. Dentro dessa estrutura, executa-se outra repetição onde são gerados novos polígonos que não tenham interseção com aqueles da lista anterior. A seguir, adiciona-se o polígono na lista de polígonos. Atualiza-se a taxa de ocupação do cenário e, por fim, cria-se um mapa através da lista de polígonos.

4.9.3 Configurações Usadas na Geração dos Mapas

As especificações das configurações dos mapas utilizados em cada uma das categorias propostas é dada na Tabela 4. Essas configurações foram utilizadas para gerar um conjunto de mapas aleatórios que serão utilizados nos experimentos para validar o algoritmo planejador. Observando essa tabela, percebe-se a relação existente entre os diferentes tipos de regiões. Por exemplo, os cenários do tipo Mapa Fácil possuem 20% de cobertura por regiões do tipo Φ^n , 30% do tipo Φ^p e 50% do tipo Φ^b . Tal porcentagem de cobertura ocorre sobre a porcentagem cadastrada do mapa. Um processo análogo é feito para as outras categorias chamadas Mapa Normal e Mapa Difícil. Um total de 100 cenários para cada configuração foi gerado automaticamente, assim ao todo foi gerado 600 mapas.

Neste trabalho foi considerado que cada mapa teria como dimensões $1000m \times 1000m$. Os experimentos assumem que a situação crítica do VANT ocorreu no centro do mapa, assim a

Tabela 4 – Configurações do gerador de mapas para cada tipo de categoria.

Área de Cobertura	Região	Dificuldade		
		Mapa Fácil	Mapa Normal	Mapa Difícil
25%	Φ^n	20%	30%	50%
	Φ^p	30%	30%	30%
	Φ^b	50%	40%	20%
50%	Φ^n	20%	30%	50%
	Φ^p	30%	30%	30%
	Φ^b	50%	40%	20%

Fonte: Dados da pesquisa.

aeronave terá um grau maior de liberdade e poderá ir para cada uma das direções possíveis. A Tabela 5 apresenta a notação para descrição das instâncias utilizadas nos experimentos. Todos os mapas estão disponíveis para download no site¹.

Tabela 5 – Notação utilizada em cada instância com sua descrição.

Instância	Notação	Descrição
I1	$\mathcal{M}_F^{25\%}$	Mapa fácil com 25% de cobertura
I2	$\mathcal{M}_F^{50\%}$	Mapa fácil com 50% de cobertura
I3	$\mathcal{M}_N^{25\%}$	Mapa normal com 25% de cobertura
I4	$\mathcal{M}_N^{50\%}$	Mapa normal com 50% de cobertura
I5	$\mathcal{M}_D^{25\%}$	Mapa difícil com 25% de cobertura
I6	$\mathcal{M}_D^{50\%}$	Mapa difícil com 50% de cobertura

Fonte: Dados da pesquisa.

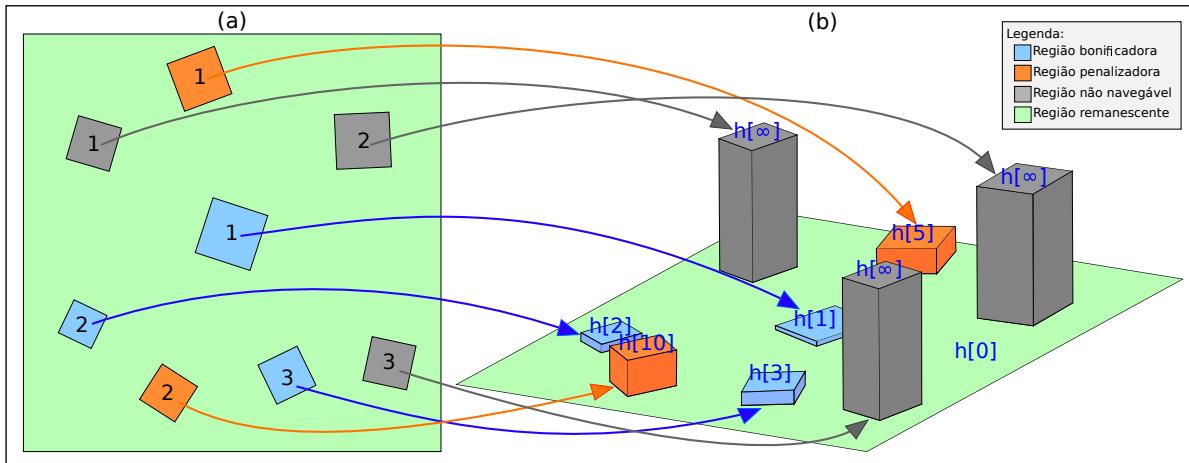
4.9.4 Mapeamento de Cenários 2D para 3D

Os mapas tridimensionais utilizados neste trabalho foram adaptados dos 600 mapas bidimensionais gerados. O método para transformação dos mapas bidimensionais em tridimensionais é ilustrado na Figura 24.

No mapeamento ilustrado, as regiões não navegáveis possuem altitude infinita indicada pela notação $h[\infty]$. As regiões penalizadoras possuem altitudes que variam no intervalo [5; 25] com incremento (inc^p) de 5. Por exemplo, a primeira região penalizadora $id = 1$ possui altitude 5m indicada por $h[5]$, já a segunda região penalizadora $id = 2$ tem altitude 10m representado por $h[10]$. De forma análoga, as regiões bonificadoras possuem altitude que variam no intervalo de [1; 5] com incremento (inc^b) de 1. Exemplificando, a primeira região bonificadora $id = 1$ tem altitude de 1m ($h[1]$), a segunda possui altitude 2m ($h[2]$) e a terceira possui altitude 3m ($h[3]$). As regiões remanescente possuem uma altitude de 0m indicado por $h[0]$. Ao se atingir o intervalo

¹ <http://lcrserver.icmc.usp.br/projects/uav/wiki/Wiki>

Figura 24 – Mapeamento dos cenários 2D para 3D. (a) mapa de origem 2D. (b) mapa de destino mapeado em 3D.



Fonte: Elaborada pelo autor.

superior das regiões a operação módulo (%) será aplicada conforme a Equação 4.10, onde ub indica o *upper bound* da região e inc representa o fator de incremento. A Tabela 6 destaca os valores dos parâmetros inc e ub usados nas regiões bonificadoras e penalizadoras. Será sempre considerado neste trabalho que no momento da situação crítica o VANT estaria a uma altitude padrão de 100m.

$$h = (((id - 1) \times inc)\%ub) + inc \quad (4.10)$$

Tabela 6 – Parâmetros e valores das alturas das regiões no mapa 3D.

Região	Incremento	Valor	Upper Bound	Valor	Altura
Φ^b	inc^b	1	ub^b	5	$h[1, 2, 3, 4, 5]$
Φ^p	inc^p	5	ub^p	25	$h[5, 10, 15, 20, 25]$

Fonte: Dados da pesquisa.

4.10 Estratégias de Pouso

A seguir é feita uma descrição de como foi desenvolvida as estratégias de pouso 2D e pouso 3D.

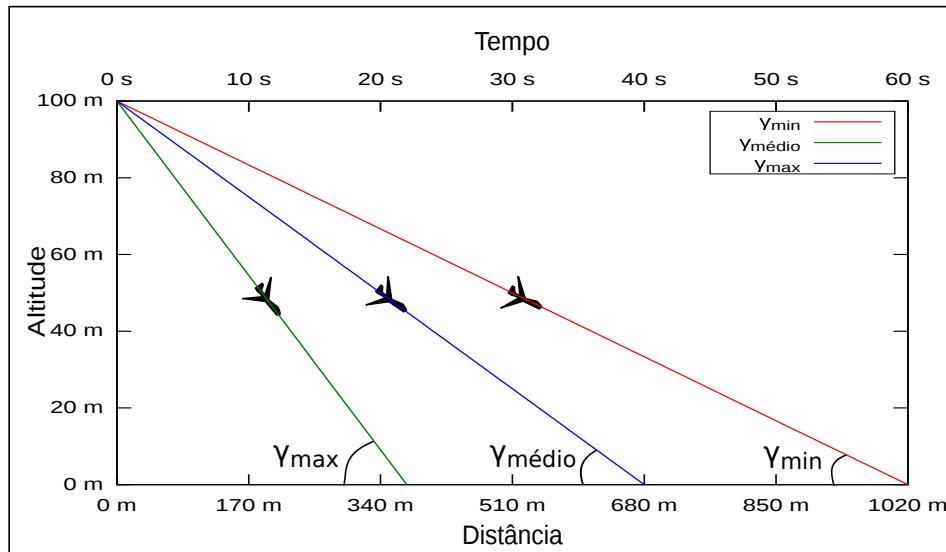
4.10.1 Estratégia para Pouso em mapas 2D

Na estratégia de pouso bidimensional, o algoritmo planejador de rota delibera apenas sobre a posição horizontal, ou seja, coordenadas (x, y) da aeronave. Todavia, uma estratégia sobre o pouso do VANT deve levar em conta não apenas as latitudes e longitudes, mas também as altitudes. A abordagem utilizada foi considerar que a altitude do VANT cai constantemente com

o tempo, assim, torna-se possível saber quando ele chegará ao solo. O algoritmo planejador do pouso não precisa levar em conta a dimensão altitude, apenas latitude e longitude. A altitude pode ser calculada posteriormente, através de um processo simples a partir do número de *waypoints* e da altura inicial da aeronave. Essas informações permitem estimar os valores dos pontos no eixo *z* em cada *waypoints*. Os métodos HG, AG e AGMP usam essa estratégia.

A Figura 25 mostra como foi assumida a estratégia de pouso para duas dimensões, onde o ângulo de descida da aeronave foi considerado constante e a altura cai linearmente. O valor do ângulo de descida depende do tempo de voo até o pouso. Por isso, foi considerado que a altitude em que o VANT estava no momento da situação crítica era de 100 metros e o tempo máximo até o pouso é de 60 segundos. Estimou-se a distância máxima alcançada pela aeronave após a falha como sendo 1020 metros, obtida a partir da velocidade da mesma.

Figura 25 – Estratégia de pouso 2D para diferentes tempos gastos no pouso.



Fonte: Elaborada pelo autor.

Os valores dos ângulos são dados pela seguinte equação: $\gamma = \arctan(\frac{h}{d})$, onde h representa a altura do VANT no momento da falha e d a distância percorrida até o pouso. A distância percorrida é dada por: $d = \frac{(v_0 + v_K)}{2} \cdot K$, onde v_0 , v_K e K representam respectivamente a velocidade inicial, velocidade final e tempo gasto até o pouso. Considerando tais fatores foi calculado o ângulo de descida mínimo (γ_{min}), máximo (γ_{max}) e médio (γ_{medio}).

Os valores obtidos para os ângulos foram:

$$\gamma_{\text{min}} = \arctan\left(\frac{h}{d}\right) = \arctan\left(\frac{100 \cdot 2}{(24 + 11) \cdot 60}\right) = 5,6^\circ \quad (4.11)$$

$$\gamma_{\text{medio}} = \arctan\left(\frac{h}{d}\right) = \arctan\left(\frac{100 \cdot 2}{(24 + 11) \cdot 40}\right) = 8,4^\circ \quad (4.12)$$

$$\gamma_{max} = \arctan\left(\frac{h}{d}\right) = \arctan\left(\frac{100 \cdot 2}{(24+11) \cdot 22}\right) = 15,0^\circ \quad (4.13)$$

Na Equação (4.11), desejamos encontrar o valor do menor ângulo de pouso com $h = 100m$, $v_0 = 24m/s$, $v_K = 11m/s$ e $K = 60s$. Calculando o arco tangente, obteremos o valor do ângulo que é de $5,6^\circ$. Na Equação (4.12), o tempo médio de pouso para as várias falhas consideradas foi de 40 segundos (veja Tabela 11). Nesse caso o valor do ângulo encontrado foi de $8,4^\circ$. O valor do maior ângulo é dado pela Equação (4.13), onde foi considerado que a taxa de descida mais forte é para o problema no motor. Assim o tempo médio de descida é de $K = 22$ (veja Tabela 11), o valor do ângulo encontrado é de $15,0^\circ$.

Observando o valor do ângulo $\gamma_{min} = 5,6^\circ$ e $\gamma_{medio} = 8,4^\circ$, percebemos que eles estão dentro do intervalo de ângulo de pouso seguro encontrado na literatura entre $4,64^\circ$ e $9,21^\circ$ para aeronaves de grande porte (EASA, 2012). O valor do ângulo obtido para o caso de problema no motor é de $\gamma_{max} = 15,0^\circ$ que corresponde a um valor acima dos ângulos encontrados. Todavia, por se tratar de um VANT, esse valor pode ser admitido uma vez que aeronaves pequenas fazem curvas mais bruscas que uma aeronave de grande porte.

4.10.2 Pousos 3D

Uma outra estratégia de pouso desenvolvida é a tridimensional, onde o algoritmo planejador de rota define em conjunto as coordenadas (x, y, z) da aeronave. O modelo baseado em PLIM usa essa estratégia.

A Figura (26) exemplifica essa estratégia de pouso, onde o cenário possui um terreno plano contendo três regiões povoadas, um aeroporto e um campo de futebol abandonado. O modelo PLIM decide toda a rota de pouso e aqui não é considerado o decaimento linear da altitude do VANT. Essa estratégia faz a alocação da altitude da melhor forma, baseado no modelo de dinâmica e no risco envolvido nas colisões laterais e sobre o topo dos obstáculos.

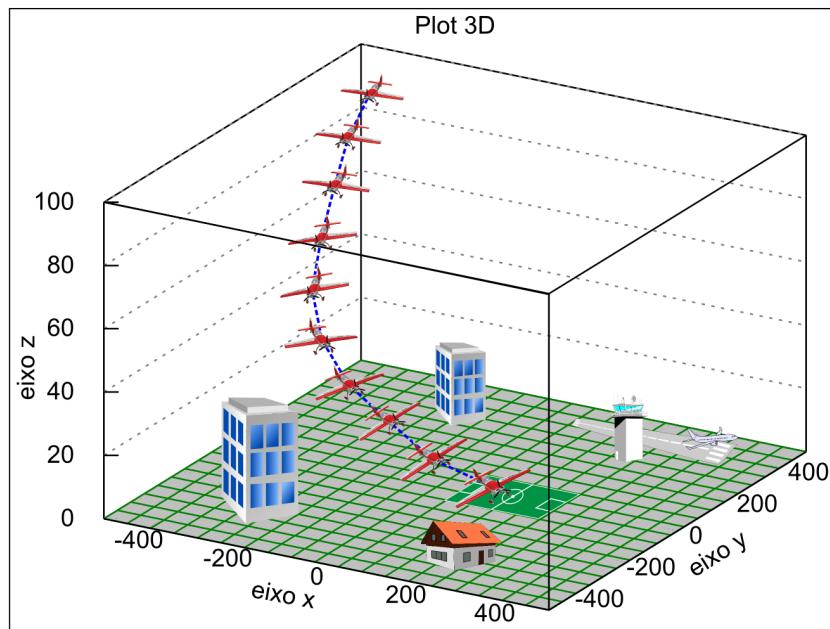
4.11 Considerações Gerais

Esta seção faz algumas considerações sobre o VANT, os experimentos realizados e os pesos atribuídos às regiões.

4.11.1 Considerações Sobre o VANT

O veículo aéreo possui um conjunto geral de sensores os quais o sistema IFA acessa e que podem ser utilizados por outros sistemas embarcados como o MOSA. Por sua vez, o sistema supervisor da aeronave (IFA) possui um conjunto específico de sensores para auxílio na descoberta de situações críticas. Abaixo, encontra-se a descrição desses sensores.

Figura 26 – Estratégia de pouso 3D em terreno plano



Fonte: Elaborada pelo autor.

Conjunto Geral de Sensores da Aeronave:

- **GPS:** sensor capaz de obter a localização;
- **Velocímetro:** sensor capaz de retornar a velocidade;
- **Altímetro:** sensor capaz de descobrir a altitude do local;
- **Giroscópio:** sensor capaz de obter a orientação.

Conjunto Específico de Sensores do IFA:

- **Termômetro:** sensor que obtém a temperatura da bateria e, caso ultrapasse um determinado limiar, o IFA aciona o algoritmo de pouso emergencial;
- **Nível da Bateria:** sensor que verifica a carga atual da bateria e, se estiver abaixo de um *threshold*, o IFA aciona o algoritmo de pouso emergencial;
- **Tacômetro:** sensor que indica a velocidade de rotação do motor e, caso haja pouca ou nenhuma rotação, o sistema IFA aciona o mecanismo de pouso;

- **Verificador de Problemas na Asa:** algoritmo capaz de verificar problemas na asa, dado o histórico do conjunto de sensores da aeronave. Esse verificador é capaz de fazer a identificação de falhas nos controles látero-direcionais. Em caso de falhas associadas, o IFA dispara um método para pousá-la ou, em último recurso, aciona um paraquedas.

Esse conjunto de sensores é importante para que a aeronave tenha o aparato suficiente para descobrir as falhas e as informações necessárias do estado atual do VANT. Sem os sensores do IFA, o VANT não saberia da ocorrência da falha e sem seus sensores gerais ele não teria informações suficientes para decidir onde pousar.

4.11.2 Considerações Sobre os Experimentos Realizados

Esta seção apresenta as ideias contidas por trás dos três tipos de experimentos realizados nesta dissertação. As condições das simulações e as diferenças entre elas são aqui elucidadas. A Figura 27 retrata os experimentos efetuados que aqui são chamados *offline*, *online* e real. Os dois primeiros experimentos foram totalmente simulados e o terceiro utiliza um VANT real.

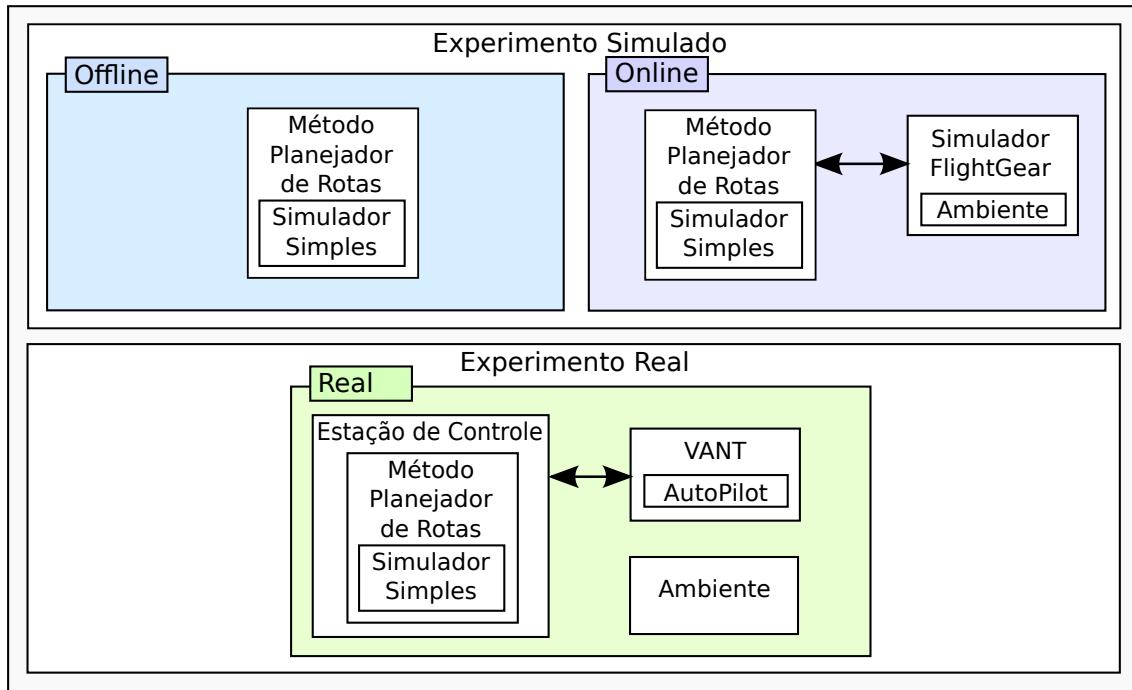
No experimento *offline*, o método planejador de rotas considera um simulador simples que retrata apenas as informações básicas relacionadas ao ambiente (mapa). Assim, os dados de entrada do algoritmo são sempre parâmetros fixos definidos para a posição, velocidade e direção. Os dados relacionados aos tipos, quantidades e posicionamento dos obstáculos é que podem variar de uma simulação para a outra.

No experimento *online*, tem-se uma aeronave executando uma missão no simulador FlightGear. Durante o voo, ocorre uma situação crítica e, nesse momento, as informações dos sensores a bordo da aeronave são passados ao algoritmo planejador de rota. Logo, como a aeronave estava simulando a execução de uma missão, os parâmetros de posição, velocidade e direção são obtidos diretamente da aeronave. Em seguida, uma rota é calculada pelo algoritmo planejador, a missão em curso é abortada e o pouso emergencial é executado.

Um Piloto Automático (AP) foi utilizado para fazer o veículo aéreo seguir a rota. Esse AP usa um controlador Proporcional Integral Derivativo (PID) e está integrado ao ambiente de testes AutoFG. Esse ambiente foi desenvolvido pelo grupo de pesquisa localizado no LCR/ICMC/USP ([HOSSOMI, 2015](#)). A simulação *online* é mais realista por considerar uma missão em curso, podendo contar com diferentes condições de vento e direção.

O último tipo de experimento ocorre em um voo real com um VANT. Nesse caso, a rota é previamente calculada na estação de controle, usando a simulação *offline*. Em seguida, a rota é carregada no VANT que a executa no mundo real. Nessa execução, um AP (ardopilot) controla a aeronave para seguir os *waypoints* gerados pelo método de pouso.

Figura 27 – Tipos de experimentos realizados.



Fonte: Elaborada pelo autor.

4.11.3 Considerações Sobre os Pesos das Regiões

As equações (3.1), (4.3), (4.4) e (4.5) possuem como parâmetros os pesos associados a queda e voo do VANT sobre algumas regiões. Esta seção tem por objetivo discutir os valores utilizados para esses parâmetros. Os pesos das regiões foram baseados em custo financeiro da aeronave e das regiões de pouso ou navegabilidade. A Tabela 7 mostra os valores dos pesos das regiões usadas em todas as simulações efetuadas.

Tabela 7 – Configurações dos pesos de cada região do mapa.

Parâmetro	Valor
C_{Φ^b}	\$2.000,00
C_{Φ^p}	\$8.000,00
C_{Φ^n}	\$100.000,00
C_{Φ^r}	\$0,00

Fonte: Dados da pesquisa.

Observando a Tabela 7, nota-se que o custo de se pousar o VANT em uma região bonificadora (C_{Φ^b}) é de \$2.000,00 unidades financeira. Esse valor pode ser interpretado como o custo de uma aeronave que foi salva (valor negativo nas equações que utilizam C_{Φ^b}) ao pouso nessa região. O prejuízo associado ao pouso do VANT em uma região penalizadora (C_{Φ^p}) é de \$8.000,00. Tal valor representa uma despesa associada ao dano estimado à aeronave e ao ambiente. O custo correspondente ao pouso e voo sobre a região não navegável (C_{Φ^n}) indica

uma penalização de \$100.000,00. Com relação a esse valor é difícil fazer uma associação com prejuízos financeiros, assim, a alta penalização faz com que os métodos escolham não sobrevoar nem pousar em regiões não navegáveis. Por fim, o custo da região remanescente (C_{Φ^r}) é zero indicando assim que a aeronave ao cair nessa região não terá nenhum prejuízo associado. Deve-se ressaltar que o fato de não se penalizar tal região é uma forma de não colocá-la no modelo, visto que, essa região é não convexa.

4.12 Considerações Sobre a Constante de Arrasto

A modelagem da dinâmica de voo depende de uma série de fatores, onde um deles é a resistência do ar. Algumas deduções matemáticas são realizadas nesta seção para determinar um valor adequado para a constante de arrasto que se ajuste ao VANT Tiriba usado neste trabalho. Seja a equação do arrasto (4.14):

$$F_t^d = \frac{1}{2} \cdot C_d \cdot \rho \cdot \mathcal{A} \cdot v_t^2 \quad (4.14)$$

com os parâmetros: coeficiente de arrasto C_d , massa específica do fluido ρ e a área de referência \mathcal{A} . Seja K_d uma constante de arrasto associada a cada aeronave dada pela equação (4.15).

$$K_d = C_d \cdot \rho \cdot \mathcal{A} / 2 \quad (4.15)$$

Assim, a equação de arrasto pode ser reescrita pela equação (4.16).

$$F_t^d = K_d \cdot v_t^2 \quad (4.16)$$

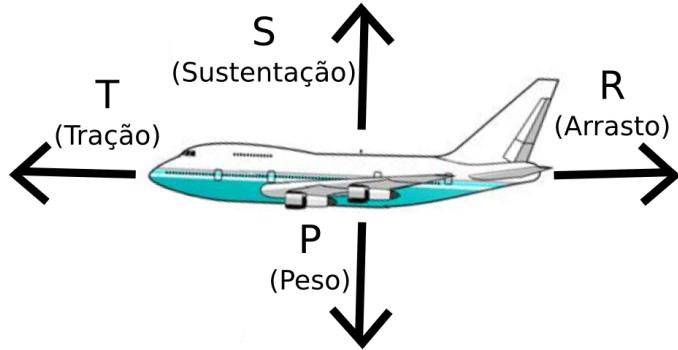
A unidade de medida de K_d é kg/m . O problema em determinar o valor de K_d ocorre por não ser conhecido o valor de \mathcal{A} e C_d associado ao VANT.

A Figura 28 mostra uma aeronave e as forças envolvidas na mesma. A primeira força aerodinâmica é a T que indica a tração produzida pelos motores. A segunda força S é a força de sustentação da aeronave. A terceira é o arrasto R , contrária ao movimento da tração T . A última força é dada pelo peso P da aeronave e aponta sempre para baixo. Para uma aeronave em voo de cruzeiro, as forças $T = R$ e $S = P$, ou seja, a aeronave está em equilíbrio.

Usando a expressão $T = R$, pode-se encontrar a constante K_d que faz com que a aeronave não consiga ultrapassar a velocidade terminal, representada por v_{term} . Para isso, basta fazer $T_{max} = R_{max}$ com $T_{max} = m \cdot a_{max}$, onde os parâmetros T_{max} , m e a_{max} representam, respectivamente, a tração máxima, a massa e a aceleração máxima que a aeronave consegue manter. Tem-se também que $R_{max} = K_d \cdot v_{term}^2$. Desenvolvendo as expressões e substituindo pelos dados do VANT Tiriba temos:

$$T_{max} = R_{max} \quad (4.17)$$

Figura 28 – Análise das forças envolvidas na aeronave.



Fonte: Adaptada de [Studart e Dahmen \(2006\)](#).

$$m \cdot a_{max} = K_d \cdot v_{term}^2 \quad (4.18)$$

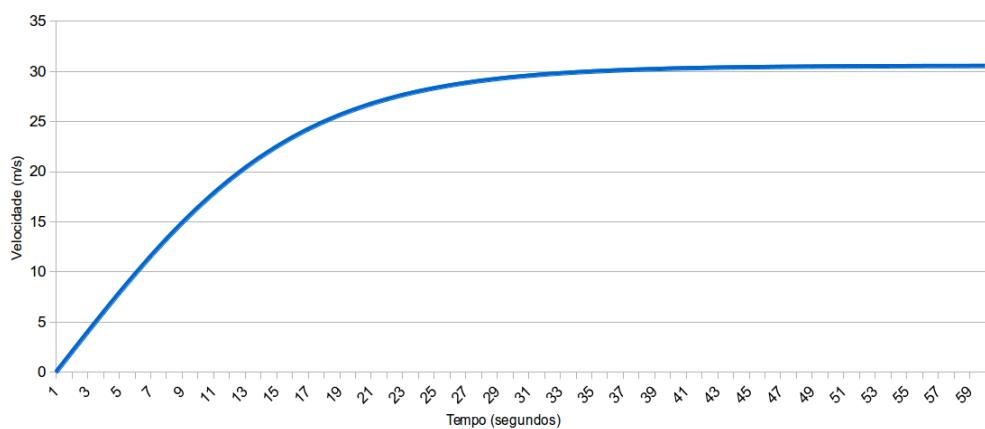
$$K_d = \frac{m \cdot a_{max}}{v_{term}^2} \quad (4.19)$$

$$K_d = \frac{3,7 \cdot 2,0}{30,5^2} = 0,007925 \quad (4.20)$$

A Expressão (4.20) mostra o valor de K_d para o Tiriba, onde os dados de massa, aceleração máxima e velocidade terminal para essa aeronave são dados na Tabela 1.

A Figura 29 apresenta um gráfico ilustrando a modelagem feita para o Tiriba. A mudança de velocidade do Tiriba é observada, partindo do repouso e com aceleração máxima. Após o instante 31, a velocidade da aeronave satura, ou seja, atingiu a velocidade terminal. O valor da constante $K_d = 0,007925$ será utilizado em todos os experimentos desta dissertação.

Figura 29 – Mudança de velocidade do VANT partindo do repouso e com aceleração constante.



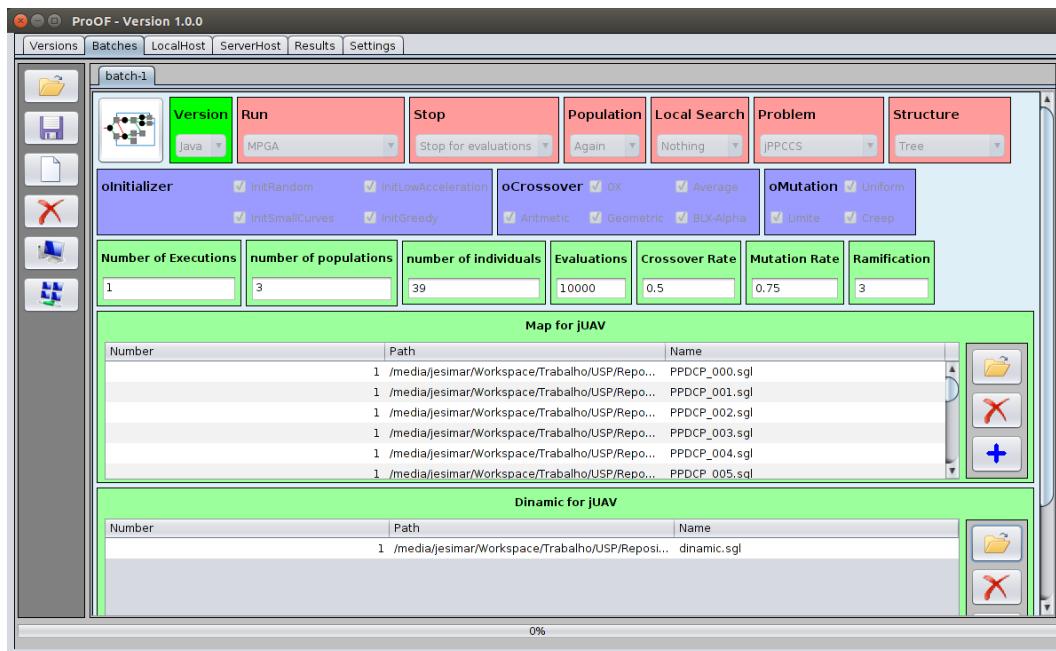
Fonte: Elaborada pelo autor.

4.13 Framework ProOF

Esta seção visa descrever o *framework* de otimização utilizado no desenvolvimento dos métodos de planejamento de rota. O ProOF (*Professional Optimization Framework*) é um ambiente desenvolvido para auxiliar a geração de códigos para métodos aplicados a problemas de otimização mono e multi-objetivo. Esse *framework* permite ao usuário tanto a inclusão de métodos quanto problemas. O ProOF também oferece suporte para o desenvolvimento de métodos exatos, heurísticos, metaheurísticos e híbridos (ARANTES, 2014).

A Figura 30 apresenta a interface gráfica do ambiente de otimização ProOF. A interface, após carregar o problema e os métodos programados, permite ao usuário definir a configuração dos parâmetros utilizados nos experimentos.

Figura 30 – Framework ProOF utilizado no desenvolvimento dos métodos.



Fonte: Elaborada pelo autor.

Todos os métodos de planejamento de rota utilizados nesta dissertação foram programados dentro desse *framework* em linguagem Java. Uma das grandes vantagens desse ambiente é o fato de possuir uma série de métodos de otimização e problemas já definidos. Outra vantagem é a facilidade de fazer testes com diferentes parâmetros de forma automatizada, o que acelera a geração de resultados.

4.14 Conclusão

O capítulo apresentou a codificação do problema e a definição da função objetivo. Os métodos propostos para o planejamento da rota HG, AG, AGMP e PLIM são descritos com seus

respectivos pseudocódigos. A arquitetura do sistema desenvolvida com suas entradas e saídas foi apresentada.

Esse capítulo mostrou também a forma como são criados os mapas e as estratégias de pouso 2D e 3D. Algumas considerações são feitas a respeito dos sensores do VANT, os experimentos realizados, a definição dos pesos das regiões e a dedução da constante de arrasto. Por fim, é mencionado o *framework* ProOF utilizado para facilitar o desenvolvimento dos métodos.



RESULTADOS

“Você nunca sabe que resultados virão da sua ação. Mas se você não fizer nada, não existirão resultados.”

Mohandas Karamchand Gandhi

5.1 Introdução

Este capítulo apresenta os resultados obtidos pela presente dissertação de mestrado. A próxima seção descreve os experimentos efetuados para a Heurística Gulosa (HG), Algoritmo Genético (AG) e Algoritmo Genético Multi-Populacional (AGMP) usando a modelagem restrita. Em seguida, os experimentos para o modelo de Programação Linear Inteira Mista (PLIM) usando a modelagem completa são apresentados. Por fim, os resultados envolvendo simulador de voo e um voo real são mostrados.

5.2 Experimentos da Modelagem Restrita

A seguir são apresentados os resultados obtidos pelos métodos propostos usando a modelagem restrita. Dessa forma, os métodos HG, AG e AGMP foram avaliados usando a estratégia de pouso 2D.

5.2.1 Configurações Utilizadas

Uma série de experimentos foi feita para analisar e decidir os melhores parâmetros, as configurações e o comportamento dos métodos. As configurações gerais do mapa, VANT, modelo e pesos das regiões utilizadas estão definidas na Tabela 8. Os valores dos pesos foram definidas conforme a Tabela 7 discutida anteriormente. Alguns valores dos parâmetros relativos ao VANT foram definidos conforme a Tabela 1 baseados em dados do Tiriba.

Observando a Tabela 8 temos que a posição da aeronave no momento da falha é o centro do mapa na coordenada (0,0); a velocidade inicial é de $v_0 = 24m/s$; o ângulo do VANT

Tabela 8 – Configurações do mapa, VANT, modelo e pesos das regiões usados.

Categoría	Descrição do Parâmetro	Símbolo	Valor	Unidade
Mapa	Dimensão no eixo X	-	1000	metros (m)
	Dimensão no eixo Y	-	1000	metros (m)
VANT	Posição inicial	(p_0^x, p_0^y)	(0,0)	metros (m)
	Velocidade inicial	v_0	24	m/s
	Ângulo inicial	α_0	90	graus ($^{\circ}$)
	Velocidade mínima	v_{min}	11,1	m/s
	Velocidade máxima	v_{max}	30,5	m/s
	Velocidade angular mínima	ε_{min}	-3	$^{\circ}/s$
	Velocidade angular máxima	ε_{max}	3	$^{\circ}/s$
	Aceleração mínima	a_{min}	0,0	m/s^2
	Aceleração máxima	a_{max}	2,0	m/s^2
Modelo	Horizonte de planejamento	T	60	segundos (s)
	Discretização do tempo	ΔT	1	segundos (s)
	Probabilidade de violar Φ^n	Δ	0,001	-
Pesos	Custo bonificadora	C_{Φ^b}	2.000	\$
	Custo penalizadora	C_{Φ^p}	8.000	\$
	Custo não navegável	C_{Φ^n}	100.000	\$
	Custo remanescente	C_{Φ^r}	0	\$

Fonte: Dados da pesquisa.

é de $\alpha_0 = 90^{\circ}$, ou seja, voando em direção ao norte; a mudança de direção está limitada a $-3^{\circ}/s \leq \varepsilon \leq 3^{\circ}/s$; a aceleração está limitada ao intervalo $0,0m/s^2 \leq a \leq 1,1m/s^2$. O valor do peso C_{Φ^p} indica uma penalização de \$8.000,00 quando associada a queda da aeronave na região do tipo penalizadora. Vale relembrar que o peso C_{Φ^b} de \$2.000,00 não indica penalização nas equações, mas sim um bônus. Os experimentos apresentados a seguir foram executados em um computador com as especificações da Tabela 9.

Tabela 9 – Configurações do computador utilizado nos experimentos.

Atributo	Configuração
Sistema Operacional	Linux - Ubuntu 13.10
Processador	Intel(R) Core(TM) i5-3337U
Conjunto de Instruções	64 bits
Número de Núcleos	4
Frequência	1.80 GHz
Memória RAM	4 GB

Fonte: Dados da pesquisa.

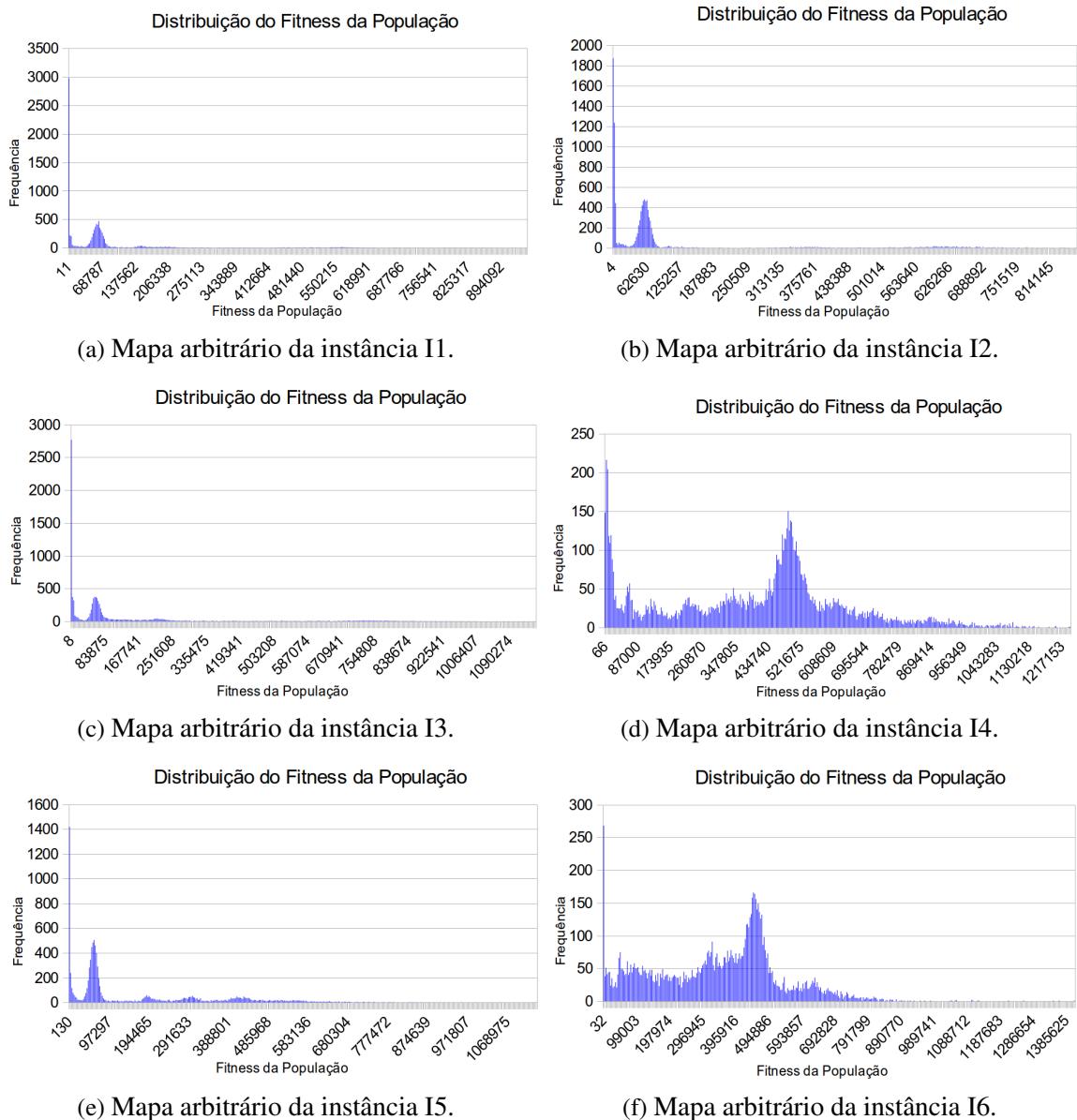
5.2.2 Análise da Distribuição do Fitness dos Indivíduos

A análise da distribuição do *fitness* dos indivíduos é importante, pois nos informa como estão agrupados os indivíduos da população em função do *fitness*. Fazer essa análise em diferentes

gerações também revela como os indivíduos evoluem através das gerações. Dois experimentos foram efetuados para capturar essas informações considerando a primeira e a última geração.

O algoritmo utilizado nesses experimentos foi o AGMP e a configuração utilizada foi: todos os operadores de inicialização, cruzamento e mutação; tamanho da população de 10.000 indivíduos; número de avaliações de 100.000; estrutura em árvore com fator de ramificação 3; taxa de *crossover* de 1,5; taxa de mutação de 0,75.

Figura 31 – Resultados da distribuição do *fitness* dos indivíduos, após a primeira geração. Os mapas foram escolhidos arbitrariamente.



Fonte: Elaborada pelo autor.

A Figura 31 apresenta a distribuição do *fitness* após a primeira geração do algoritmo. Percebe-se uma grande variação nas distribuições geradas para cada um dos mapas de categorias distintas, isto ocorre pelo fato dos mapas terem características bem diferentes e o método ser

estocástico. A diversidade de *fitness* dos indivíduos ocorre pela natureza do AGMP implementado, que inicialmente procura gerar soluções com características aleatórias.

Pode-se perceber alguns picos (Figura 31) nas distribuições no início do gráfico (*fitness* próximo de 4 a 130) que indica queda em uma região bonificadora. Essas soluções em regiões bonificadoras foram obtidos provavelmente pelo inicializador guloso. O segundo pico percebido, nos gráficos (a), (b), (c) e (e), indica soluções que sobrevoaram provavelmente algumas regiões não navegáveis, mas não caíram em tal região. Isso é indicado pelo alto valor de *fitness*, mas abaixo do custo da penalização $C_{\Phi^n} = \$100.000,00$. O restante das regiões dos gráficos, com custo acima de \$100.000,00, são soluções que provavelmente sobrevoaram e pousaram em região não navegável (Φ^n).

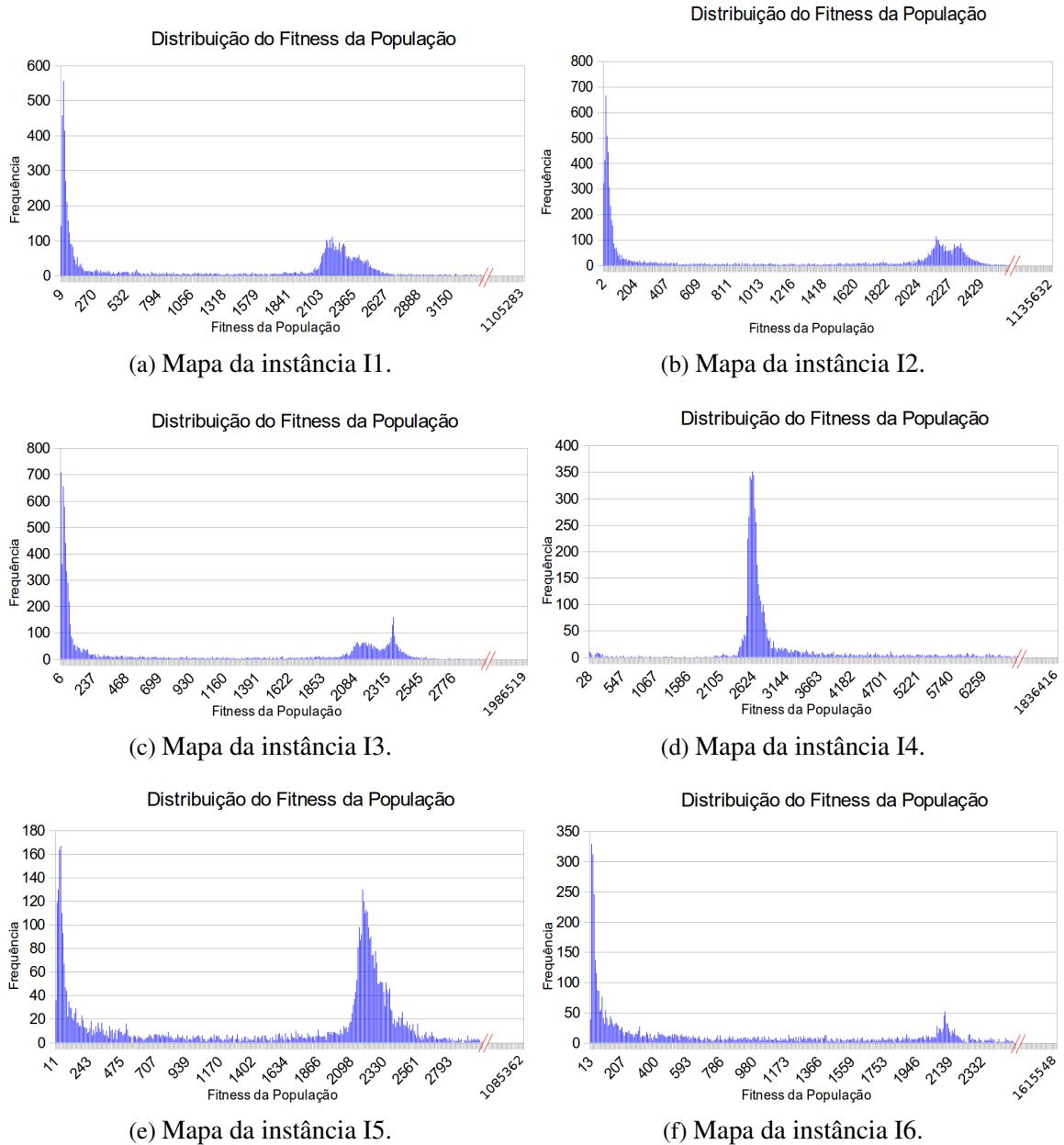
A Figura 32 apresenta as distribuições, após a última geração. Há uma relevante melhoria na distribuição dos indivíduos, onde se nota um agrupamento maior de indivíduos no início do gráfico (*fitness* próximo a 2 a 200). Isso indica maior queda em região bonificadora. Um outro agrupamento de indivíduos ocorre próximo ao *fitness* 2000 relacionado a queda da aeronave em uma região remanescente. As soluções restantes (indivíduos da população) estão distribuídos de forma bastante esparsa pelo gráfico. Isso ocorre já que, mesmo estando na última geração, indivíduos com *fitness* podem ocorrer na população dada a característica do método, onde operadores de mutação e *crossover* podem gerar indivíduos "ruins".

5.2.3 Análise de Sensibilidade do AG e AGMP

Esta seção apresenta uma análise de sensibilidade dos parâmetros do AG e do AGMP. A Tabela 10 contém o conjunto de parâmetros analisados pelo algoritmo. Por exemplo, o número de populações testadas varia de {1; 2; 3; 4; 5}, já os operadores de mutação foram {Uniforme; Limite; Creep; Todos}. Neste caso, *Todos* indica que os operadores anteriores citados estavam ativados, onde o método aplica um deles ao indivíduo via seleção aleatória. Essa tabela apresenta também alguns valores já utilizados na literatura (coluna P^L). A coluna P^* mostra os melhores parâmetros encontrados na análise de sensibilidade efetuada. A ordem dos testes também é importante, visto que, para cada variação feita, fixou-se os demais parâmetros analisados. O valor no conjunto sublinhado foi fixado para efetuar os demais experimentos. Assim, quando se encontrou os melhores resultados para (P^*), este valor passou a ser utilizado e então se prosseguiu avaliando os demais parâmetros, seguindo a ordem estabelecida na tabela.

As condições em que foram realizados o ajuste dos parâmetros são as seguintes: selecionou-se arbitrariamente um mapa de cada conjunto de instância ($I1, I2, \dots, I6$), totalizando seis mapas, sendo um para cada categoria. Um total de 50 execuções foi realizado para cada mapa e o algoritmo para após executar 10.000 avaliações. O AGMP replaneja a rota de pouso sem considerar uma falha associada, assim se procurou descobrir os melhores parâmetros independente de situação crítica.

Figura 32 – Resultados da distribuição do *fitness* dos indivíduos, após a última geração. Os mapas utilizados foram os mesmos do experimento anterior.



Fonte: Elaborada pelo autor.

A métrica utilizada, em caso de empate estatístico dos parâmetros, é usar o valor padrão da literatura e prosseguir os experimentos. Esse critério foi especificado devido ao grande número de empates entre os parâmetros. Esse empate ocorre dada a considerável complexidade do problema tratado, indicando que o método se comporta de forma parecida independente dos valores aqui testados para os parâmetros. Isso também se explica pela complexa curva do *fitness* (veja Seção 5.2.2).

Analizando a Tabela 10 na coluna P^* e a linha do N° de Ramificação, nota-se a escolha de 3^L . Esse tipo de resultado indica que houve empate entre os diferentes parâmetros, entretanto,

Tabela 10 – Análise de sensibilidade de parâmetros do AGMP e AG.

Método	Ordem	Parâmetro Analisado	Variações dos Parâmetros	P^L	P^*
AGMP	1	Nº de Ramificação	{2; 3; 4}	3	3^L
	2	Nº de Populações	{1; 2; 3; 4; 5}	3	3^L
	3	Nº de Indivíduos	{6; 13; 20; 27; 34}	13	13^L
	4	Taxa de Crossover	{0,5; 1,0; 1,5; 2,0; 2,5}	1,5	0,5
	5	Taxa de Mutação	{0,0; 0,25; 0,5; 0,75; 1,0}	0,75	$0,75^L$
	6	Operad. de Crossover	{Aritmético; Média; Geométrico; OX; BLX- α ; Todos}	-	Todos
	7	Operad. de Mutação	{Uniforme; Limite; Creep; Todos}	-	Todos
	8	Operad. de Inicialização	{Aleatório; CurvaCurta; Aceler.Curta; Gulosa; Todos}	-	Todos
AG	9	Nº de Indivíduos	{13; 39}	-	39
	10	Tamanho do Torneio	{2; 3; 4; 5}	-	3

Fonte: Dados da pesquisa.

usou-se como desempate o valor 3 usado também por outros trabalhos da literatura.

Os resultados da análise de sensibilidade foram executados primeiramente para o AGMP e depois para o AG. Assim, os melhores parâmetros encontrados no multi-populacional foram também usados no AG. Contudo, o único parâmetro que ocorria no AG e não havia equivalente no AGMP foi o tamanho do torneio. Devido a estrutura de população diferenciada do AG em relação ao AGMP, dois experimentos medindo número de indivíduos ocorreram em paralelo. O primeiro a ser testado verifica o tamanho do torneio para uma população de 13 indivíduos. O segundo experimento toma o número de indivíduos igual a 39 e verifica o tamanho do torneio. É importante ressaltar que nesse experimento também houve empate e foi selecionado o número de indivíduos igual a 39 e tamanho do torneio de 3. Todos os resultados da análise de sensibilidade de parâmetros estão disponíveis no site¹.

5.2.4 Exemplo de Soluções para Situações Críticas

Esta seção apresenta alguns exemplos de soluções obtidas para diferentes situações críticas em um mesmo mapa. Tais exemplos, ajudam a avaliar o comportamento do VANT para cada uma das situações enfrentadas. Ao todo foram realizadas cinco simulações diferentes, onde em quatro delas ocorreram problemas associados à aeronave e em uma simulação foi efetuado o pouso sem situação crítica.

O propósito é ilustrar as rotas seguidas pela aeronave, assim o AGMP foi utilizado. A Tabela 11 apresenta os resultados obtidos para um mapa do conjunto de instâncias. Por exemplo, na segunda linha dessa tabela há informações sobre o pouso, dado o problema no motor (Ψ^m), onde o *fitness* associado a solução foi de 21,00. A quantidade de tempo gasto (K) até o pouso foi de 22 segundos e o local de pouso do VANT foi uma região bonificadora (Φ^b).

A Figura 33 mostra as trajetórias seguidas nos poucos para cada uma das situações críticas. Inicialmente o veículo aéreo parte da posição Início e ao chegar ao centro do mapa

¹ http://lcrserver.icmc.usp.br/projects/uav_critical_situation/wiki

Tabela 11 – Resultado obtido para diferentes situações críticas enfrentadas pelo VANT.

Ψ	Fitness	K	Local Pouso
Ψ^m	21,00	22	Φ^b
Ψ^b	344,10	34	Φ^b
Ψ^{s1}	37,80	60	Φ^b
Ψ^{s2}	28,10	48	Φ^b
Ψ^\emptyset	13,40	34	Φ^b
Média	88,88	40	Φ^b

Fonte: Dados da pesquisa.

(rota na cor verde) ocorre uma falha que lhe obriga a efetuar o pouso emergencial. Na Figura 33 (a), percebe-se que a aeronave pousou rapidamente devido ao problema no motor e alcançou uma região bonificadora (rota na cor vermelho). Já na Figura 33 (b), o veículo aéreo com problema na bateria percorreu uma distância maior até chegar na região Φ^b . Nesse caso, seu *fitness* foi penalizado em relação ao problema no motor por percorrer uma maior distância. Na Figura 33 (c), o VANT consegue encontrar uma trajetória até uma região bonificadora, mesmo com a restrição de girar somente para a esquerda. E por fim, a Figura 33 (d) indica que a aeronave é capaz de pousar em uma região Φ^b , apesar da restrição de somente conseguir girar para a direita. Observa-se que, devido a essa restrição de manobra, o VANT teve que efetuar um laço para pousar.

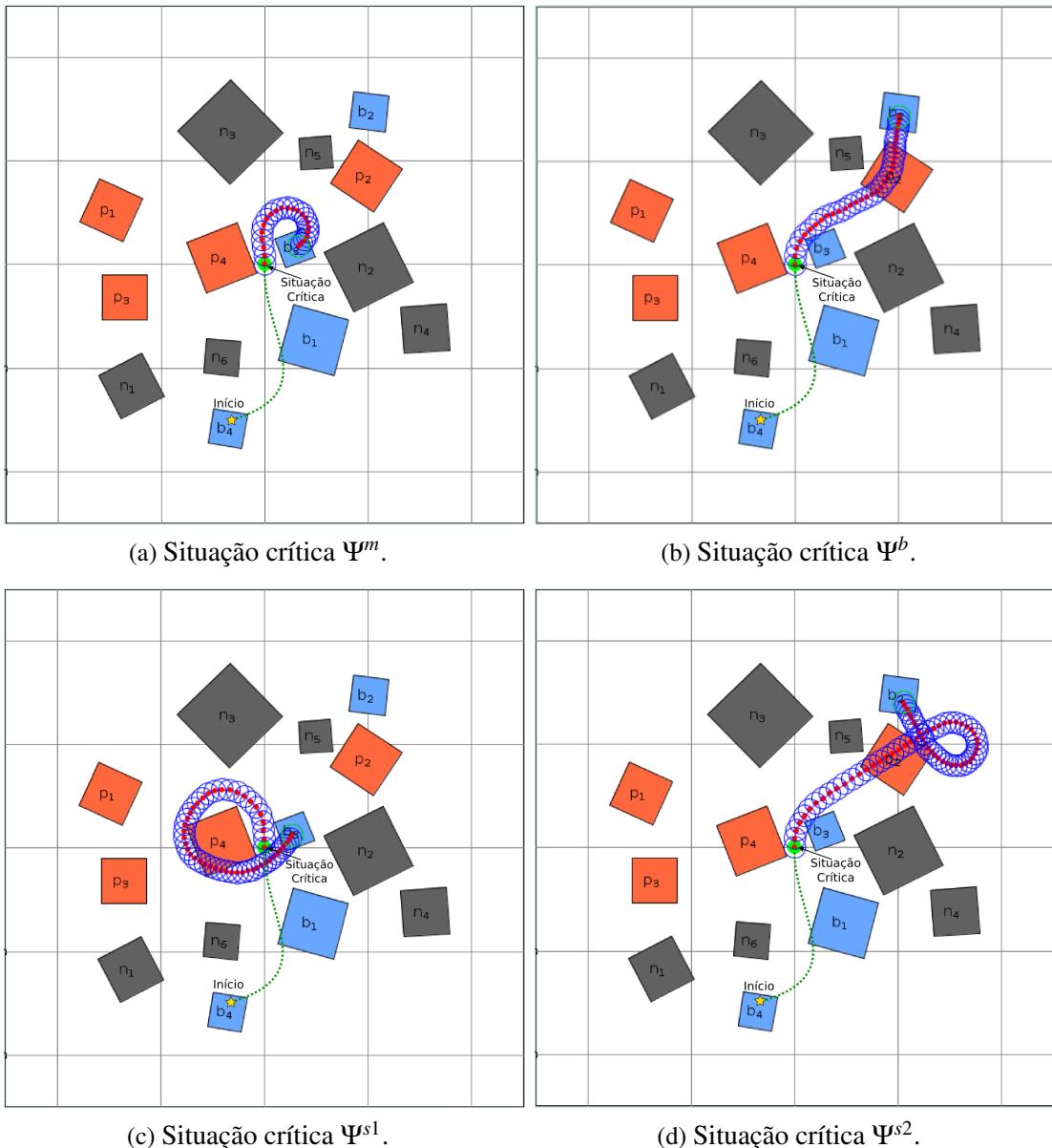
As seções seguintes apresentarão os resultados obtidos para cada um dos métodos implementados. Todavia, algumas considerações precisam ser feitas. Quando se fala que uma solução (rota) é factível significa que ela não violou três restrições importantes: Pouso (Φ^b), Velocidade (v_{min}) e Risco (Δ). As Restrições de Pouso estabelecem que o VANT deve encontrar uma região bonificadora e conseguir pousar nela; as restrições de Velocidade verificam se a aeronave não conseguiu pousar no último instante, pois estava acima da velocidade mínima; as restrições de Risco avaliam se o risco foi violado, ou seja, a aeronave invadiu o espaço aéreo não navegável (Φ^n).

5.2.5 Heurística Gulosa

Esta seção, apresenta os resultados obtidos pela Heurística Gulosa (HG) que podem ser vistos na Tabela 12. As condições iniciais em que os experimentos foram submetidos são: todos os 600 mapas criados ($I1, I2, \dots, I6$) foram avaliados uma vez pela HG; todas as situações críticas modeladas foram avaliadas, conforme apresentado na tabela.

A Tabela 12 mostra o tempo médio (em segundos) para cada tipo de instância. As colunas indicadas por *Pousou Φ^b* e *Pousou Φ^r* apresentam a quantidade de soluções que pousaram em regiões bonificadoras e remanescente com segurança. A coluna *Infactível* indica a quantidade de soluções que violou uma ou mais das seguintes restrições: condição de não navegabilidade;

Figura 33 – Exemplos de rotas para cada uma das situações críticas.



Fonte: Elaborada pelo autor.

pouso em região penalizadora; condição de velocidade mínima para o pouso. Como para cada tipo de instância há 100 mapas, o somatório das colunas *Pousou* Φ^b , *Pousou* Φ^r e Infactível totaliza 100. Por exemplo, para a situação crítica Ψ^\otimes a instância *I1* obteve o melhor resultado com 99% soluções que levam a rotas seguras de voo, pousando em regiões bonificadoras. Nessa mesma situação, a instância *I5* obteve o pior resultado com 67% das soluções nessas condições.

É importante observar que o tempo gasto na execução do algoritmo é inferior a 0,120 segundo, em geral, sendo aproximadamente 0,071 segundo. Ressalta-se que a taxa de sucesso é de 77,83%, considerando todas as situações de pouso descritas. A HG não é satisfatória em 13,27% dos mapas testados. O baixo custo computacional justifica sua utilização antes de outras

Tabela 12 – Resultado obtido para a HG, para todos as situações críticas e todas as instâncias.

Ψ	Instâncias	Tempo	Pousou Φ^b	Pousou Φ^r	Infactível
Ψ^\emptyset	I1	0,041	99	0	1
	I2	0,091	97	0	3
	I3	0,037	93	3	4
	I4	0,085	98	0	2
	I5	0,025	67	5	28
	I6	0,065	83	0	17
	Média	0,057	89,50	1,33	9,17
Ψ^m	I1	0,067	79	21	0
	I2	0,111	92	6	2
	I3	0,059	58	39	3
	I4	0,104	86	12	2
	I5	0,042	30	52	18
	I6	0,088	62	28	10
	Média	0,079	67,83	26,33	5,83
Ψ^b	I1	0,041	99	0	1
	I2	0,092	97	0	3
	I3	0,037	93	3	4
	I4	0,087	98	0	2
	I5	0,024	67	5	28
	I6	0,065	83	0	17
	Média	0,058	89,50	1,33	9,17
Ψ^{s1}	I1	0,067	81	8	11
	I2	0,120	88	0	12
	I3	0,061	68	16	16
	I4	0,113	82	1	17
	I5	0,042	41	23	36
	I6	0,095	56	0	44
	Média	0,083	69,33	8,00	22,67
Ψ^{s2}	I1	0,066	90	4	6
	I2	0,117	90	0	10
	I3	0,060	70	20	10
	I4	0,109	87	1	12
	I5	0,038	40	17	43
	I6	0,091	61	3	36
	Média	0,080	73,00	7,50	19,50
	Média Final	0,071	77,83	8,90	13,27

Fonte: Dados da pesquisa.

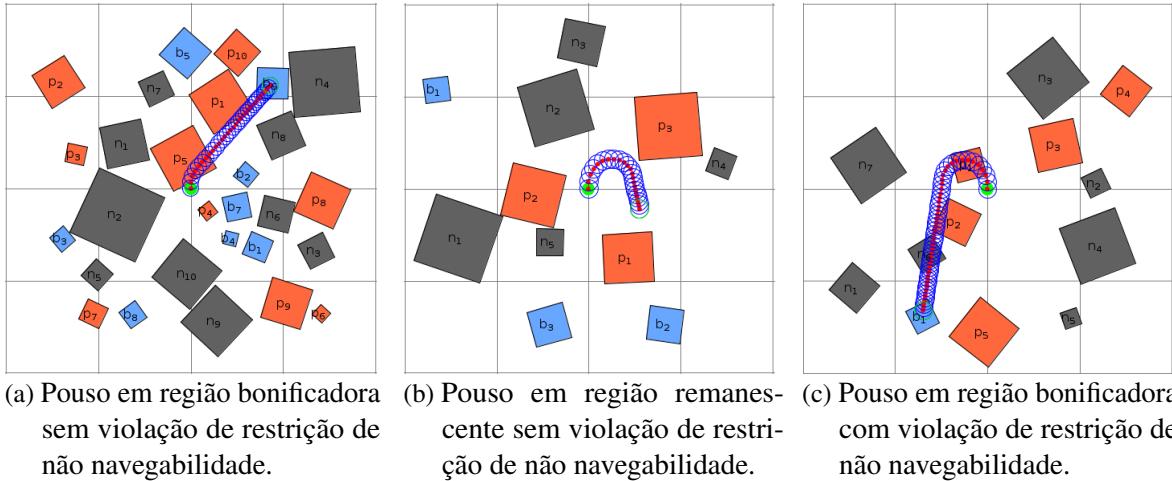
técnicas mais complexas e que eventualmente gastem um tempo maior para retornar uma rota.

A Figura 34 mostra três exemplos de rotas geradas pela HG. Na Figura 34 (a), nota-se que o veículo aéreo conseguiu chegar até a região bonificadora. Já na Figura 34 (b), a aeronave não conseguiu chegar a Φ^{b_2} devido a sua longa distância. As regiões Φ^{b_1} e Φ^{b_3} não foram escolhidas, pois violariam a restrição de não navegabilidade, caso o VANT tentasse alcançá-las. O resultado da Figura 34 (c) revela que houve um voo sobre a região não navegável Φ^{n_6} indicando uma solução ruim, mesmo que tenha pousado em região bonificadora (Φ^{b_1}). Esses resultados para a HG foram reportados em Arantes *et al.* (2015).

5.2.6 Algoritmo Genético

Os resultados obtidos pelo Algoritmo Genético (AG) estão agrupados na Tabela 13. O critério de parada do algoritmo ocorre após 10.000 avaliações. Duas versões diferentes do AG

Figura 34 – Exemplos de rotas obtidas pela HG.

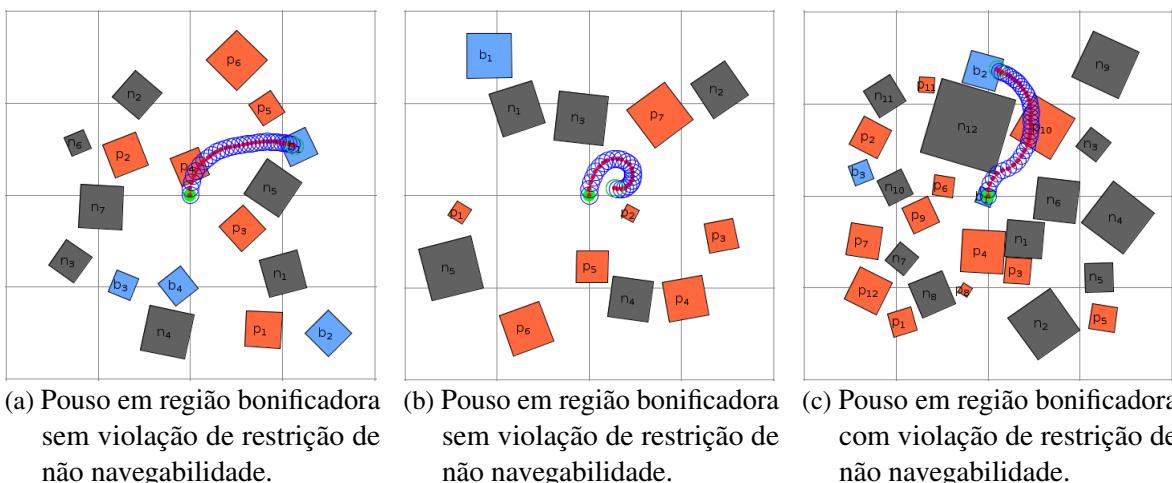


Fonte: Elaborada pelo autor.

foram criadas sem e com a inicialização gulosa, sendo chamadas de AG1 e AG2, respectivamente. Essas versões foram criadas para mostrar a robustez da inicialização gulosa.

Analizando os dados da Tabela 13, percebe-se que o AG1 teve um desempenho semelhante à HG resolvendo 77,96% dos mapas. O AG com o operador de inicialização guloso (AG2) teve um desempenho melhor ao resolver 89,86% das instâncias. Assim, o AG2 define rotas para cerca de 12,0% mais mapas que o AG1. No quesito tempo gasto, as duas versões acabaram gastando tempos semelhantes na ordem de 0,955 e 0,912 segundo. Comparando o tempo gasto do AG2 com a HG, percebe-se que HG é 12 vezes mais rápido que o método evolutivo.

Figura 35 – Exemplos de rotas obtidas pelo AG.



Fonte: Elaborada pelo autor.

Alguns exemplos de rotas gerados pelo AG são ilustrados na Figura 35. Na Figura 35 (a), percebe-se que a aeronave conseguiu chegar até uma região bonificadora e pousá-la com

Tabela 13 – Resultado obtido para o AG1 e AG2, para todos as situações críticas e todas as instâncias.

Ψ	Instâncias	AG1				AG2			
		Tempo	Pousou Φ^b	Pousou Φ^r	Infactível	Tempo	Pousou Φ^b	Pousou Φ^r	Infactível
Ψ^\emptyset	I1	0,718	100	0	0	0,650	100	0	0
	I2	0,791	98	1	1	0,734	99	0	1
	I3	0,804	96	1	3	0,682	99	0	1
	I4	0,932	97	0	3	0,801	99	0	1
	I5	0,901	81	18	1	0,777	97	3	0
	I6	1,239	66	15	19	0,996	94	4	2
	Média	0,897	89,67	5,83	4,50	0,773	98,00	1,17	0,83
Ψ^m	I1	0,789	81	19	0	0,739	90	10	0
	I2	0,751	88	11	1	0,783	96	3	1
	I3	0,831	59	39	2	0,788	71	28	1
	I4	0,822	81	18	1	0,808	95	5	0
	I5	0,916	34	65	1	0,905	40	60	0
	I6	0,982	59	31	10	0,960	81	16	3
	Média	0,848	67,00	30,50	2,50	0,830	78,83	20,33	0,83
Ψ^b	I1	1,059	96	3	1	0,921	100	0	0
	I2	1,097	99	0	1	1,002	99	0	1
	I3	1,068	91	7	2	0,928	98	1	1
	I4	1,141	96	1	3	1,072	99	0	1
	I5	1,107	68	31	1	1,024	95	5	0
	I6	1,301	62	17	21	1,191	96	2	2
	Média	1,129	85,33	9,83	4,83	1,023	97,83	1,33	0,83
Ψ^{s1}	I1	0,773	91	6	3	0,846	92	6	2
	I2	0,888	88	0	12	0,888	95	0	5
	I3	0,894	77	17	6	0,926	86	9	5
	I4	1,028	81	2	17	1,001	90	0	10
	I5	1,045	49	43	8	1,021	73	22	5
	I6	1,338	39	25	36	1,224	78	5	17
	Média	0,994	70,83	15,50	13,67	0,984	85,67	7,00	7,33
Ψ^{s2}	I1	0,709	93	5	2	0,786	99	0	1
	I2	0,759	94	1	5	0,885	94	2	4
	I3	0,811	78	16	6	0,813	92	5	3
	I4	0,968	83	8	9	1,014	94	0	6
	I5	0,912	65	31	4	0,979	78	20	2
	I6	1,286	49	17	34	1,203	77	3	20
	Média	0,908	77,00	13,00	10,00	0,947	89,00	5,00	6,00
Média Final		0,955	77,96	14,93	7,10	0,912	89,86	6,96	3,16

Fonte: Dados da pesquisa.

segurança. Já na Figura 35 (b), não se conseguiu alcançar a região Φ^{b_1} , pousando assim na região remanescente. Na Figura 35 (c), houve uma pequena sobreposição do VANT sobre a região não navegável $\Phi^{n_{12}}$ e pouso em Φ^{b_2} .

5.2.7 Algoritmo Genético Multi-Populacional

Os resultados obtidos com o Algoritmo Genético Multi-Populacional (AGMP) estão na Tabela 14. O critério de parada do algoritmo também é definido como 10.000 avaliações. Esse algoritmo evolutivo teve duas versões sem e com inicialização gulosa chamadas de AGMP1 e de AGMP2, respectivamente.

A taxa de mapas solucionados foi de 89,53% com a versão AGMP2 contra 80,13% do AGMP1. O tempo médio gasto no AGMP2 foi de 0,874 segundo que foi ligeiramente menor

Tabela 14 – Resultado obtido para o AGMP1 e AGMP2, para todos as situações críticas e todas as instâncias.

Ψ	Instâncias	AGMP1				AGMP2			
		Tempo	Pousou Φ^b	Pousou Φ^r	Infactível	Tempo	Pousou Φ^b	Pousou Φ^r	Infactível
Ψ^\emptyset	I1	0,787	100	0	0	0,723	100	0	0
	I2	0,875	99	0	1	0,810	99	0	1
	I3	0,865	95	3	2	0,846	96	4	0
	I4	1,015	98	0	2	1,046	96	1	3
	I5	1,014	85	15	0	0,762	99	0	1
	I6	1,258	77	13	10	0,890	100	0	0
Média		0,969	92,33	5,17	2,50	0,846	98,33	0,83	0,83
Ψ^m	I1	0,741	81	19	0	0,642	90	10	0
	I2	0,716	92	7	1	0,665	96	3	1
	I3	0,770	60	39	1	0,671	71	28	1
	I4	0,736	84	16	0	0,717	96	4	0
	I5	0,875	36	64	0	0,732	40	60	0
	I6	0,866	60	33	7	0,819	82	15	3
Média		0,784	68,83	29,67	1,50	0,708	79,17	20,00	0,83
Ψ^b	I1	1,044	100	0	0	0,870	100	0	0
	I2	1,107	99	0	1	0,921	99	0	1
	I3	1,064	94	5	1	0,866	99	0	1
	I4	1,144	99	0	1	1,012	100	0	0
	I5	1,157	73	27	0	0,976	94	6	0
	I6	1,310	68	17	15	1,178	95	2	3
Média		1,138	88,83	8,17	3,00	0,970	97,83	1,33	0,83
Ψ^{s1}	I1	0,938	90	8	2	0,830	91	7	2
	I2	1,059	89	0	11	0,922	93	0	7
	I3	1,057	76	18	6	0,917	86	8	6
	I4	1,160	84	3	13	1,013	89	0	11
	I5	1,195	49	46	5	1,039	67	28	5
	I6	1,443	46	23	31	1,221	78	4	18
Média		1,142	72,33	16,33	11,33	0,990	84,00	7,83	8,17
Ψ^{s2}	I1	0,859	94	4	2	0,691	99	0	1
	I2	0,931	95	1	4	0,792	95	1	4
	I3	0,956	79	16	5	0,755	92	5	3
	I4	1,113	83	8	9	0,901	94	0	6
	I5	1,080	62	35	3	0,880	74	24	2
	I6	1,373	57	13	30	1,109	76	4	20
Média		1,052	78,33	12,83	8,83	0,855	88,33	5,67	6,00
Média Final		1,017	80,13	14,43	5,43	0,874	89,53	7,13	3,33

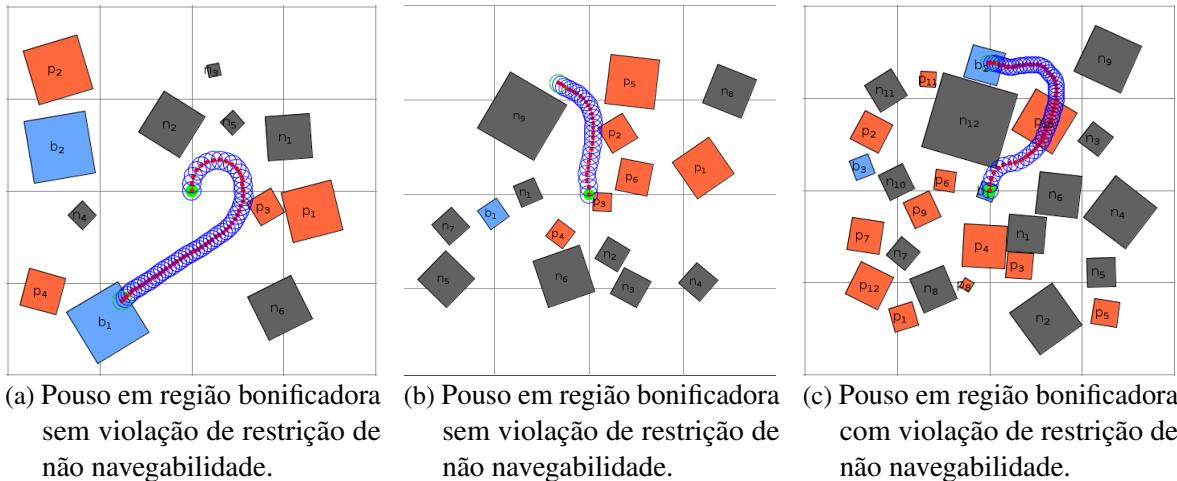
Fonte: Dados da pesquisa.

do que o AGMP1 (1,017 segundo). Ainda nessa tabela, percebe-se que o problema no motor (Ψ^m) possui o pior resultado em termos de pouso sobre Φ^b , enquanto os melhores resultados são obtidos ao tratar o pouso com problema na bateria (Ψ^b).

Uma outra análise verificou a existência de mapas infactíveis. Observou-se que apenas oito mapas de um total de 600 não foram resolvidos por nenhum dos três algoritmos (HG, AG e AGMP). Isso não prova que esses oito mapas não tenham solução, porém se pode afirmar que a maior parte (592 mapas) é factível. Assim, ao se combinar os três métodos anteriores, cerca de 98,66% dos mapas podem ser resolvidos.

Alguns exemplos de rotas alcançados pela estratégia multi-populacional são apresentadas na Figura 36. Na Figura 36 (a), nota-se que a aeronave conseguiu chegar até uma região bonificadora e pousar a aeronave. No entanto, na Figura 36 (b), a aeronave não conseguiu chegar até Φ^{b_1} , pousando na região remanescente. A Figura 36 (c) mostra que houve violação da não

Figura 36 – Exemplos de rotas obtidas pelo AGMP.



Fonte: Elaborada pelo autor.

naveabilidade sobre a região $\Phi^{n_{12}}$ semelhantemente ao AG. Os resultados apresentados para o AGMP também foram reportados em [Arantes et al. \(2015\)](#).

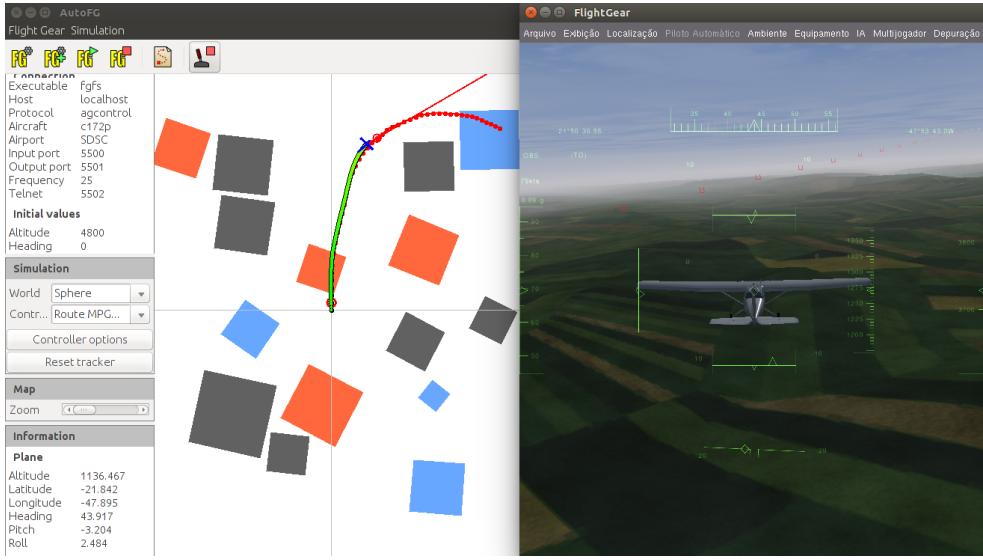
5.2.8 Experimentos Online

Os testes simulados utilizaram o simulador FlightGear (FG) por ser facilmente configurável e multiplataforma. Esse simulador também apresenta realismo, variedade de aeronaves, aeroportos, comunicação via *sockets* e facilidade de aprendizado.

Para a validação das rotas emergenciais, um Piloto Automático (AP) capaz de seguir a rota no FlightGear foi desenvolvido de forma a viabilizar voos autônomos. Logo, a simulação não foi realizada por um piloto humano. Esse controlador (AP) recebe os dados da rota e retornam os controles da aeronave. A Figura 37 mostra a estrutura que integra o FG, o AP e o AGMP. No lado esquerdo dessa figura, tem-se uma linha verde indicando o caminho original executado pela aeronave, e o trajeto vermelho é o caminho obtido pelo replanejamento do pouso emergencial. Ainda nessa figura, no lado direito, aparece o simulador de voo executando a missão de pouso emergencial.

Algumas simplificações foram assumidas nos testes com simulador. Uma vez que há poucos VANTS disponíveis no FG e aqueles existentes são pouco realísticos, a aeronave de grande porte Cessna 172 foi utilizada para atuar como VANT. Como os mapas gerados nos testes anteriores estavam projetados para aeronave de pequeno porte, os testes em uma aeronave maior como o Cessna 172 demandaram adaptações naqueles mapas. Isso foi realizado através de uma ampliação das dimensões originais do mapa e da rota em 10 vezes. O sistema de AP desenvolvido segue a rota em duas dimensões e decresce linearmente a altitude do VANT até chegar no último *waypoint*.

Figura 37 – Simulação usando o FlightGear para seguir a rota gerada pelo AGMP.



Fonte: Elaborada pelo autor.

A Figura 38 mostra quatro exemplos de rotas geradas pelo AGMP e como foi o percurso executado pela aeronave. É importante notar que a rota seguida pela aeronave é bastante fiel a rota gerada pelo AG. Alguns vídeos com essas simulações estão disponíveis no site² e no site³. As rotas divergem um pouco da rota original pelo fato do ambiente ser mais complexo e de algumas limitações do controlador desenvolvido. Apesar dessas limitações, as equações de dinâmica desenvolvidas são bastante fieis à dinâmica suportada pela aeronave, já que a aeronave consegue seguir a rota gerada.

Ainda na Figura 38, as simulações são feitas para dois mapas distintos: um do grupo I3 e outro do grupo I6. Durante o voo, foi assumido um problema na bateria (Ψ^b). As Figuras 38 (a) e (c) mostram os caminhos quando há ventos de 10 knots durante a simulação de voo, enquanto as Figuras 38 (b) e (d) apresentam ventos de 50 knots. Os ventos trazem distúrbios que não permitem seguir exatamente a rota do AGMP em ambos os mapas. Como esperado, a diferença entre o caminho do AGMP e o caminho executado pelo AP é maior sob ventos fortes. No entanto, a aeronave fica longe das regiões não navegáveis durante todo o caminho do AGMP para ambos os mapas.

5.2.9 Estudo das Soluções do HG e AGMP

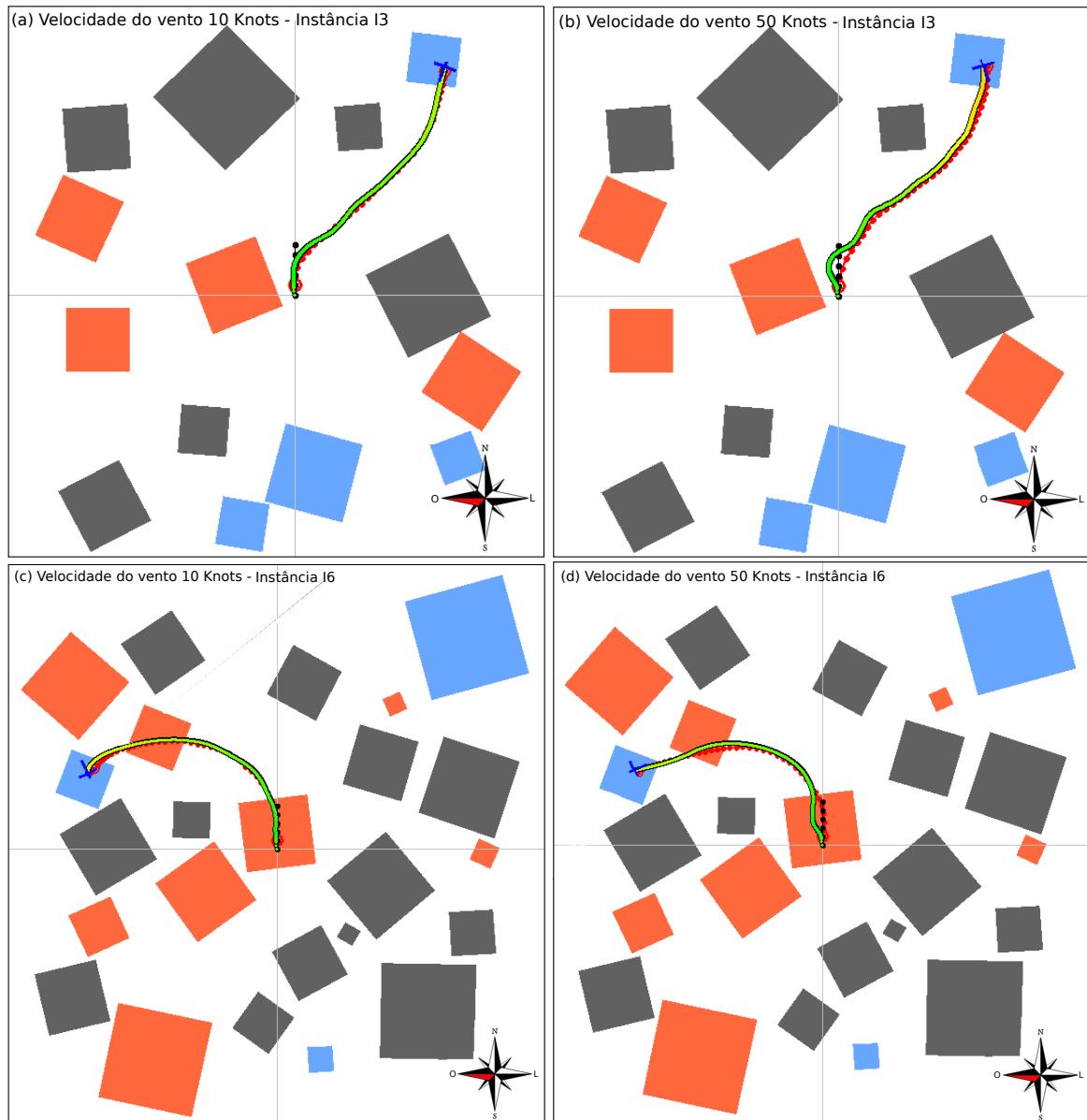
A seguir é mostrado um estudo do comportamento das soluções geradas pela HG e AGMP. Para isso, foi fixado um mapa da categoria $M_F^{50\%}$ (instância I2) e todos os comportamentos analisados foram para a aeronave sem problema (Ψ^\emptyset).

A Figura 39 apresenta o conjunto de todas as rotas obtidas pela HG. O mapa considerado

² https://youtu.be/P7t_CSSmbMs

³ <https://youtu.be/zJfiV-aWDHU>

Figura 38 – Simulação no FlightGear usando o AGMP. Em verde tem as rotas percorridas pela Cessna e em vermelho as geradas pelo AGMP. (a) e (c) ventos de 10 knots. (b) e (d) ventos de 50 knots. Direção do vento: oeste.

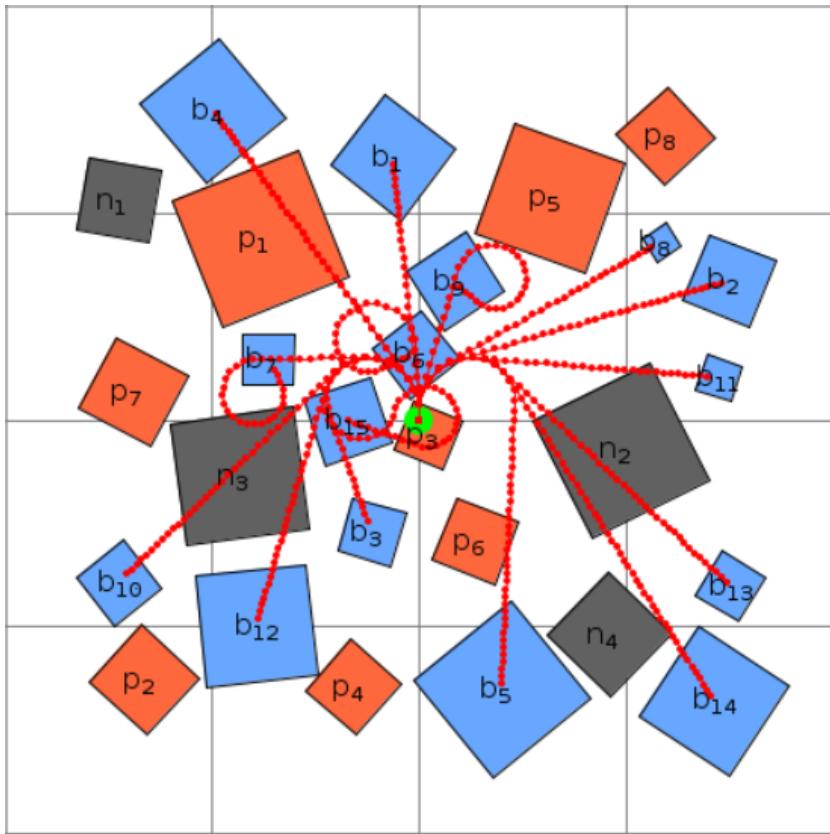


Fonte: Elaborada pelo autor.

possui um total de 15 regiões bonificadoras o que explica a HG gerar 15 rotas de pouso. Essas rotas podem ser visualizadas na cor vermelha. O método HG percorre um espaço de busca proporcional a quantidade de regiões bonificadoras, onde apenas a melhor rota é retornada. Também há algumas soluções que violam a restrição de não navegabilidade sobre Φ^n , no entanto, são penalizadas pela função objetivo.

Como o comportamento do AGMP é semelhante ao AG, não foram avaliadas as soluções obtidas pelo AG. A Figura 40 mostra como estão distribuídas as rotas do AGMP a medida que se aproxima o critério de parada (10.000 avaliações de *fitness*). As rotas estão representadas na cor verde.

Figura 39 – Todas as rotas obtidas pela HG. A melhor rota é retornada pela HG.



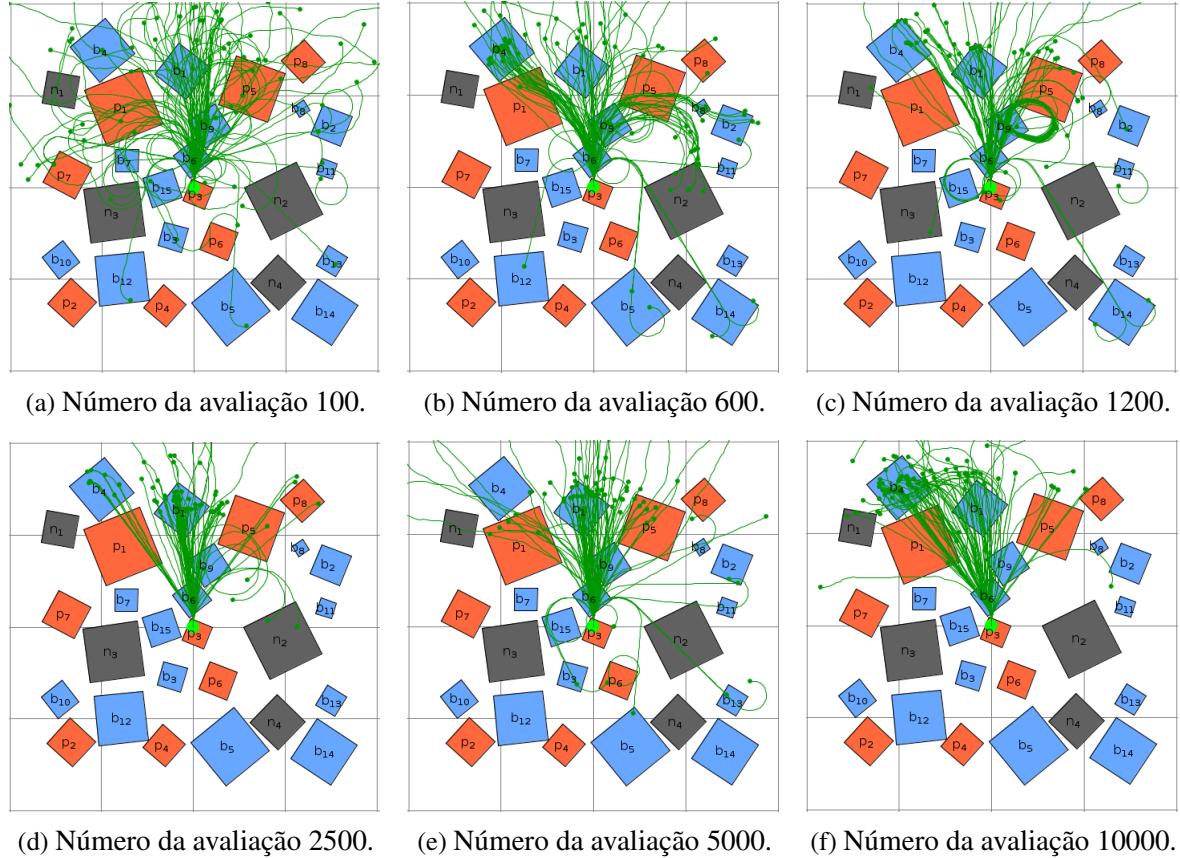
Fonte: Elaborada pelo autor.

Nessa figura, tem-se um total de 100 rotas desenhadas em todas as imagens. Na Figura 40 (a), temos os 100 primeiros indivíduos criados. Nas Figuras 40 (b), (c), (d), (e) e (f), temos as rotas dos últimos 100 indivíduos até um total de avaliações de 600, 1.200, 2.500, 5.000 e 10.000, respectivamente. Os indivíduos estão melhor distribuídos no espaço nas primeiras imagens e tendem a se concentrar ao longo das regiões bonificadoras. Isso ocorre a medida que mais avaliações são executadas, ou seja, a medida que as gerações passam e as rotas (representadas por indivíduos) evoluem. Por exemplo, na Figura 40 (a), várias rotas navegam sobre Φ^n , mas apenas duas rotas navegam sobre essa região na Figura 40 (f).

Uma última análise de dados é feita com relação a posição final ao longo de todas as 10.000 soluções geradas pelo AGMP. Devido ao grande número de indivíduos, toda a rota não foi impressa para cada um deles. Apenas o local de pouso (último *waypoint* da rota) é apresentado na Figura 41.

Pode-se perceber a diversidade de soluções obtidas pela estratégia evolutiva ao longo das 10.000 soluções geradas. Repare que há poucos em todas as categorias de regiões. As únicas regiões nas quais não ocorreram pouso foram Φ^{p_4} e Φ^{n_4} . Os locais de pouso dessa simulação foram coletados e estão transcritos na Tabela 15. Nota-se que houve um total de 54,9% de pouso em regiões bonificadoras, 40,1% sobre remanescente, 4% em penalizadora e apenas 1% de

Figura 40 – Conjunto de rotas obtidas pelo AGMP. Estão plotados as 100 últimas rotas geradas até o número da avaliação indicado.



Fonte: Elaborada pelo autor.

pouso sobre a região não navegável. Esse resultado mostra que a função objetivo guia bem o método para evitar pousar em regiões não navegáveis e penalizadoras.

Tabela 15 – Dados do local de pouso das 10.000 rotas gerados pelo AGMP.

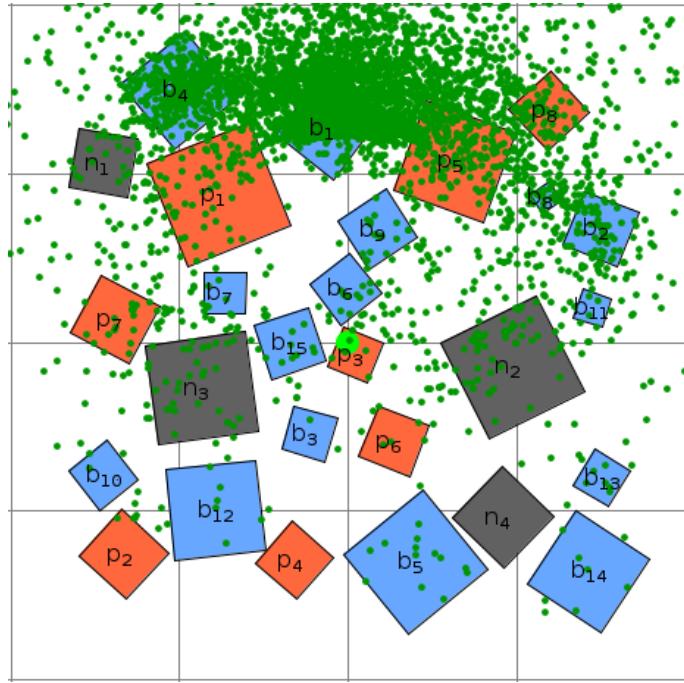
Local de Pouso	Quantidade	Porcentagem
Pouso Φ^b	5485	54,9%
Pouso Φ^r	4016	40,1%
Pouso Φ^p	401	4,0%
Pouso Φ^n	98	1,0%

Fonte: Dados da pesquisa.

5.3 Experimentos com Modelagem Completa

A seguir são apresentados os resultados obtidos a partir da resolução exata via modelo de PLIM. Nestes experimentos foi considerada a modelagem completa, ou seja, a estratégia de pouso 3D.

Figura 41 – Conjunto de todos os locais de pouso obtido pelo AGMP em 10.000 avaliações efetuadas.



Fonte: Elaborada pelo autor.

5.3.1 Configurações Utilizadas

A Tabela 16 contém as principais configurações definidas para os parâmetros do mapa, VANT, modelo matemático e *solver* CPLEX.

Um exemplo de configuração do mapa é a dimensão $1000m \times 1000m$. Na configuração do VANT, vemos que a posição inicial dele é o centro do mapa localizado a uma altitude padrão de 100 metros e a velocidade inicial do VANT é $24m/s$ com orientação para o norte. Como exemplo de configuração do modelo, temos o valor máximo do risco de violar zonas não navegáveis (Δ) e o tamanho do horizonte de planejamento (T).

O valor de Δ foi o mesmo utilizado nos experimentos usando HG, AG e AGMP, já o valor de T foi menor, pois o modelo aumenta sua complexidade em função do tamanho do horizonte. Nessa tabela temos também dois parâmetros do *solver*: limite de tempo de execução e o limite de *gap* que define a proximidade da solução encontrada da solução ótima. O Gap é definido na Equação (5.1), onde *ub* indica o *upper bound* e *lb* indica o *lower bound* da solução encontrada pelo *solver* CPLEX. Observe que o critério de parada, limite do Gap, é igual a 0,0001 indicando que as soluções retornadas pelo *solver* devem estar bastante próximas ao valor ótimo.

$$Gap(\%) = 100 \cdot \frac{CPLEX^{ub} - CPLEX^{lb}}{CPLEX^{ub}} \quad (5.1)$$

Os experimentos foram realizados em um *cluster* contendo 6 computadores com as configurações mostradas na Tabela 17, possuindo um HD compartilhado entre todos os computadores.

Tabela 16 – Configurações do mapa, VANT, modelo, *solver* e pesos usados.

Categoría	Descrição do Parâmetro	Símbolo	Valor	Unidade
Mapa	Dimensão no eixo X	-	1000	metros (<i>m</i>)
	Dimensão no eixo Y	-	1000	metros (<i>m</i>)
	Faces do polígono dos mapas	∇	4	-
VANT	Posição inicial	(p_0^x, p_0^y, p_0^z)	(0, 0, 100)	metros (<i>m</i>)
	Velocidade inicial	(v_0^x, v_0^y, v_0^z)	(0, -24, 0)	<i>m/s</i>
	Velocidade horizontal mínima	v_{min}^h	11,1	<i>m/s</i>
	Velocidade horizontal máxima	v_{max}^h	30,5	<i>m/s</i>
	Velocidade vertical mínima	v_{min}^v	-10	<i>m/s</i>
	Velocidade vertical máxima	v_{max}^v	10	<i>m/s</i>
	Velocidade horizontal no prob. motor	v_m^h	24	<i>m/s</i>
	Velocidade vertical no prob. motor	v_m^v	-6	<i>m/s</i>
	Aceleração horizontal mínima	a_{min}^h	0	<i>m/s²</i>
	Aceleração horizontal máxima	a_{max}^h	5	<i>m/s²</i>
	Aceleração vertical mínima	a_{min}^v	-5	<i>m/s²</i>
	Aceleração vertical máxima	a_{max}^v	5	<i>m/s²</i>
Modelo	Horizonte de planejamento	T	30	segundos (<i>s</i>)
	Tempo mínimo para voo	τ^{min}	10	segundos (<i>s</i>)
	Tempo máximo para voo	τ^{max}	30	segundos (<i>s</i>)
	Probabilidade de violar Φ^n	Δ	0,001	-
	Incerteza do VANT na horizontal	σ^h	3	<i>m</i>
	Incerteza do VANT na vertical	σ^v	1	<i>m</i>
	Fator de crescimento da incerteza	η	1	-
	Faces do polígono da linearização	Γ	32	-
	Velocidade de folga na falha no motor	μ	$\frac{1}{2}$	<i>m/s</i>
Solver	Limite de Tempo	-	3600	segundos (<i>s</i>)
	Limite de Gap	-	0,0001	-
Pesos	Custo bonificadora	C_{Φ^b}	2000	\$

Fonte: Dados da pesquisa.

5.3.2 Análise do Modelo

A Figura 42 compara a quantidade média de variáveis binárias, contínuas e restrições em cada categoria de instância. Analisando as instâncias I2, I4 e I6, que possuem densidade 50%, em relação as instâncias I1, I3 e I5, que possuem densidade 25%, nota-se que houve em torno do dobro de variáveis binárias e restrições. O número de variáveis contínuas é cerca de 2,5 vezes maior. Isso ocorre, pois ao dobrar a densidade (regiões), dobra-se as variáveis do modelos que são indexadas pelas regiões.

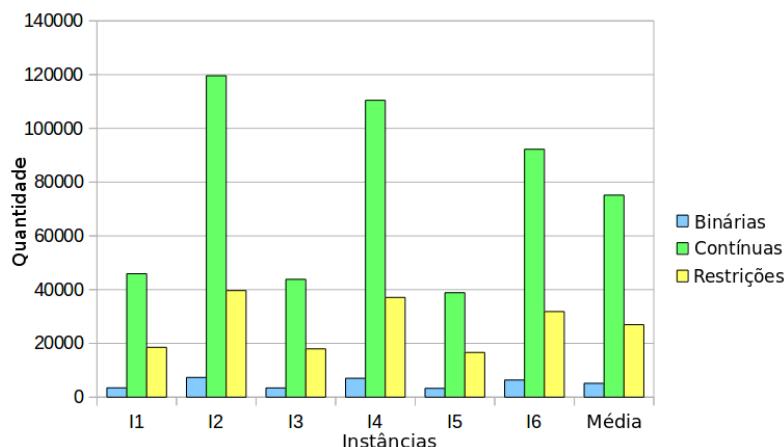
O número médio de variáveis binárias gira em torno de 5.000, enquanto a quantidade média de variáveis contínuas atinge 75.000. A quantidade de restrições em média é de 26.900.

Tabela 17 – Configurações do computador utilizado nos experimentos.

Atributo	Configuração
Sistema Operacional	Linux - Debian 8.2
Processador	Intel(R) Core(TM) i7
Número de Núcleos	8
Frequência	1.6 GHz
Memória RAM	32 GB

Fonte: Dados da pesquisa.

Figura 42 – Análise do número de variáveis contínuas e binárias e número de restrições.



Fonte: Elaborada pelo autor.

5.3.3 Experimentos Offline

Esta seção descreve os testes realizados de forma *offline*, ou seja, sem validação no simulador de voo. Os resultados foram obtidos através do modelo PLIM1 e estão sintetizados na Tabela 18, onde aparecem separados por tipo de situação crítica e por conjunto de instância.

Os dados coletados dizem respeito ao tipo de solução obtida pelo *solver*: ótima, factível, infactível e desconhecida. A solução ótima indica a melhor solução possível encontrada dentro do limite de gap. Factível indica que a solução é possível, entretanto não foi provada a optimalidade da mesma. Infactível simboliza que não existe solução para a instância. Desconhecida indica que, durante o tempo de execução, não foi encontrada solução. Também são apresentados os valores médio da função objetivo (\mathcal{W}) e do tempo de simulação, além da quantidade de pouso em regiões do tipo Φ^b e Φ^r .

A Tabela 18 indica que o maior número de rotas ótimas foi encontrado para a situação crítica problema na bateria (Ψ^b) com 99,67% dos 600 mapas. Em apenas um dos mapas foi retornada uma solução factível. Também em apenas um dos mapas não houve solução encontrada (infactível), ou seja, o *solver* não conseguiu definir uma rota de pouso em região bonificadora ou em região remanescente. O desempenho do *solver*, quando o modelo está ajustado para tratar Ψ^b , é bastante superior aquele obtido para as demais falhas. O *solver* retorna rotas em média 24

Tabela 18 – Resultados obtidos para o modelo PLIM1 para todas as situações críticas e todas as instâncias.

Ψ	Instâncias	Ótima	Factível	Infactível	Desconhecida	\mathcal{W}	Tempo	Pousou Φ^b	Pousou Φ^r
Ψ^m	I1	89	11	0	0	-758,12	450,06	44	56
	I2	66	31	3	0	-699,04	1228,88	41	56
	I3	89	9	2	0	-408,09	357,45	25	73
	I4	58	31	11	0	-573,40	1339,70	32	57
	I5	95	5	0	0	-119,30	213,65	10	90
	I6	49	26	23	2	-143,81	1348,50	11	64
Média		74,33	18,83	6,50	0,33	-458,51	789,22	27,17	66,00
Ψ^b	I1	100	0	0	0	-1309,41	9,32	100	0
	I2	100	0	0	0	-1240,47	48,99	95	5
	I3	100	0	0	0	-1038,44	12,02	89	11
	I4	100	0	0	0	-1115,36	56,13	90	10
	I5	100	0	0	0	-412,34	7,98	59	41
	I6	98	1	1	0	-396,07	60,61	51	48
Média		99,67	0,17	0,17	0,00	-918,68	32,51	80,67	19,17
Ψ^{s1}	I1	70	29	1	0	-1690,64	1237,92	91	8
	I2	67	32	1	0	-1571,51	1516,75	84	15
	I3	73	26	1	0	-1307,17	1149,11	71	28
	I4	55	44	1	0	-1347,47	1790,44	74	25
	I5	59	40	1	0	-708,93	1743,52	39	60
	I6	43	54	3	0	-660,36	2263,23	39	58
Média		61,17	37,50	1,33	0,00	-1215,97	1613,68	66,33	32,33
Ψ^{s2}	I1	61	38	1	0	-1637,22	1570,83	88	11
	I2	56	43	1	0	-1555,43	1773,33	83	16
	I3	63	36	1	0	-1341,98	1428,63	73	26
	I4	50	49	1	0	-1408,01	1933,46	77	22
	I5	60	39	1	0	-782,30	1553,70	43	56
	I6	40	55	3	2	-715,46	2195,83	40	55
Média		55,00	43,33	1,33	0,33	-1243,59	1739,15	67,33	31,00
Média Final		72,54	24,96	2,33	0,17	-959,19	1043,64	60,38	37,13

vezes mais rápido em Ψ^b do que ao tratar Ψ^m , 49 vezes mais rápido do que ao tratar o problema Ψ^{s1} e 53 vezes mais rápido do que Ψ^{s2} . Isso pode ser explicado pelo fato da função objetivo penalizar a quantidade de *waypoints* alocados no caso de problema na bateria. Assim, o pouso imediato torna-se uma prioridade nesse tipo de situação crítica.

A falha com pior desempenho no pouso foi o problema no motor, onde a aeronave foi levada até uma região bonificadora em apenas 27,17% dos mapas. Nessa tabela, vale lembrar que o somatório da quantidade de soluções ótimas, factíveis, infactíveis e desconhecida é sempre 100 para cada linha da Tabela 18. O somatório da quantidade de Pouso Ψ^b , Ψ^r , infactível também é sempre igual a 100.

Na Tabela 19, estão reportados os resultados obtidos pelo modelo PLIM2. Essa tabela está organizada da mesma forma que a Tabela 18. O comportamento nos dois modelos propostos (PLIM1 e PLIM2) não difere no caso de problema na bateria já que ambos usam a mesma função objetivo. Assim, a análise será sempre em relação as outras três falhas implementadas (Ψ^m , Ψ^{s1} e Ψ^{s2}).

O tempo computacional ficou em média 4 vezes mais rápido usando o PLIM2. Isso se explica pela menor alocação de *waypoints* no PLIM2. A menor ativação de *waypoints*, induzida pela função objetivo (3.12), leva a uma rota mais curta o que impacta num menor número de

restrições ativadas para descrever o voo da aeronave.

O fato de ter trocado a função objetivo (Equação (3.11)) por outra que possui mais penalizações (Equação (3.12)) justifica o valor em \mathcal{W} ter aumentado na Tabela 19. No entanto, é interessante observar que a quantidade das soluções que pousam na região bonificadora e remanescente praticamente não se alterou.

Tabela 19 – Resultados obtidos para o modelo PLIM2 para todas as situações críticas e todas as instâncias.

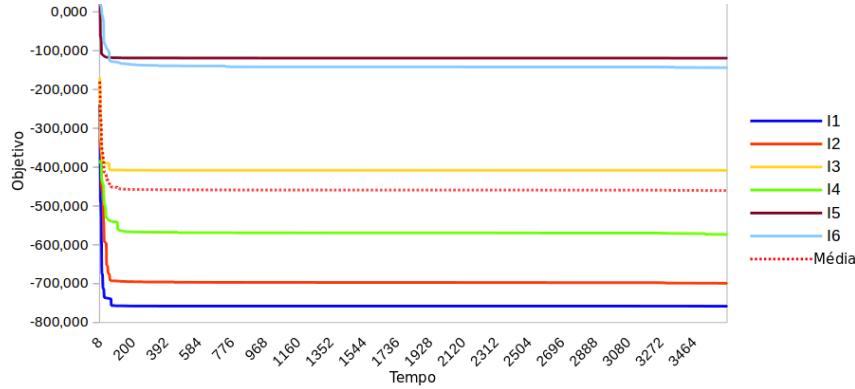
Ψ	Instância	Ótima	Factível	Infactível	Desconhecida	\mathcal{W}	Tempo	Pousou Φ^b	Pousou Φ^r
Ψ^m	I1	99	1	0	0	-69,55	63,06	44	56
	I2	93	4	3	0	-31,16	263,08	41	56
	I3	98	0	2	0	306,09	35,69	25	73
	I4	88	1	11	0	91,84	185,74	32	57
	I5	98	2	0	0	615,19	105,88	10	90
	I6	73	3	23	1	529,46	220,60	11	65
Média		91,50	1,83	6,50	0,17	231,35	141,57	27,17	66,17
Ψ^b	I1	100	0	0	0	-1309,41	9,32	100	0
	I2	100	0	0	0	-1240,47	48,99	95	5
	I3	100	0	0	0	-1038,44	12,02	89	11
	I4	100	0	0	0	-1115,36	56,13	90	10
	I5	100	0	0	0	-412,34	7,98	59	41
	I6	98	1	1	0	-396,07	60,61	51	48
Média		99,67	0,17	0,17	0,00	-918,68	32,51	80,67	19,17
Ψ^{s1}	I1	93	7	0	0	-1059,11	446,75	90	10
	I2	93	6	1	0	-1036,45	402,33	84	15
	I3	97	2	1	0	-661,40	201,98	71	28
	I4	86	12	1	1	-770,88	645,32	73	25
	I5	96	3	1	0	-87,68	229,79	39	60
	I6	88	9	3	0	-170,53	486,76	39	58
Média		92,17	6,50	1,17	0,17	-632,72	402,37	66,00	32,67
Ψ^{s2}	I1	90	9	1	0	-1015,80	426,70	86	13
	I2	88	11	1	0	-1013,14	523,16	83	16
	I3	92	7	0	1	-712,05	460,62	73	26
	I4	86	13	1	0	-872,56	591,68	78	21
	I5	91	7	1	1	-187,27	369,07	43	55
	I6	90	5	3	2	-249,02	291,18	41	54
Média		89,50	8,67	1,17	0,67	-679,06	444,81	67,33	30,83
Média Final		93,21	4,29	2,25	0,25	-499,78	255,31	60,29	37,21

O tempo médio total é 1043 segundos (17 minutos) utilizando o modelo PLIM1 e 255 segundos (4 minutos) com o PLIM2. Nos dois modelos, 60% das soluções conseguiram pousar em regiões bonificadoras e 37% pousaram em regiões remanescentes. O restante das soluções se mostra infactível ou desconhecida. O PLIM2, por rodar mais rapidamente, conseguiu provar a otimalidade das soluções em aproximadamente 93% das instâncias, já o PLIM1 provou apenas em 72% das instâncias.

A Figura 43 descreve como ocorreu a evolução do valor do objetivo no problema do motor nas seis categorias de instâncias executando o modelo PLIM1. Conforme a descrição da Tabela 5, as instâncias I1 e I2 são mais fáceis, uma vez que encontram logo de início uma rota com valor baixo para a função objetivo. As instâncias I3 e I4 apresentam dificuldade média e as instâncias I5 e I6 são consideradas mais difíceis, por terem um valor objetivo mais alto. A solução é melhorada rapidamente o que pode ser observado pela alta queda da função objetivo, seguida

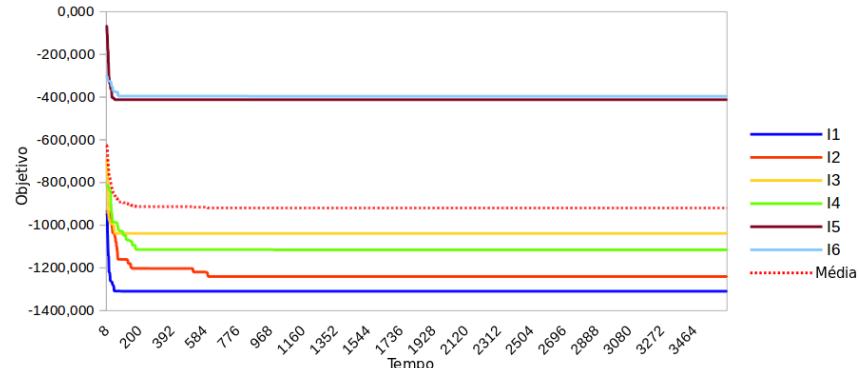
por pequenos ajustes. Comportamentos similares são observados nos gráficos das Figuras 44, 45, 46 para as demais situações críticas.

Figura 43 – Evolução do objetivo em função do tempo para o problema no motor no modelo PLIM1.



Fonte: Elaborada pelo autor.

Figura 44 – Evolução do objetivo em função do tempo para o problema na bateria no modelo PLIM1.

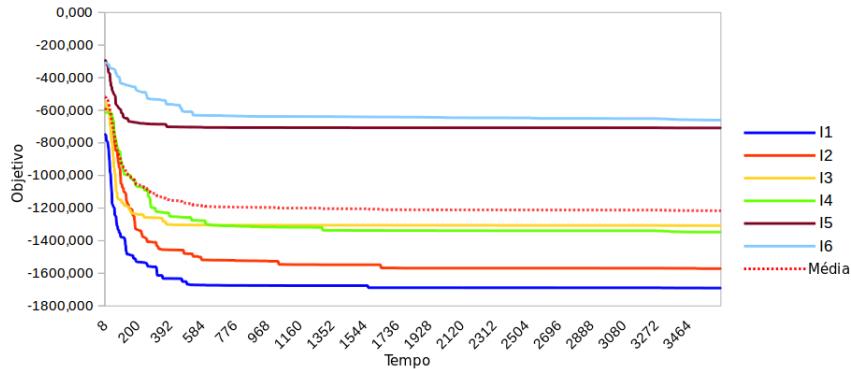


Fonte: Elaborada pelo autor.

A Figura 47 compara a média dos objetivos para diferentes situações críticas e o tempo para a obtenção das soluções. Há uma proximidade entre as falhas Ψ^{s1} e Ψ^{s2} , uma vez que as duas se distinguem apenas nas trocas de sinas e desigualdades nas restrições. Percebe-se uma queda mais acentuada nas curvas do problema no motor e na bateria com pequenos ajustes efetuados ao longo do tempo. Já nos problemas Ψ^{s1} e Ψ^{s2} uma queda mais suave no valor do objetivo ao longo do tempo ocorre, indicando que o método leva mais tempo para encontrar as soluções do problema.

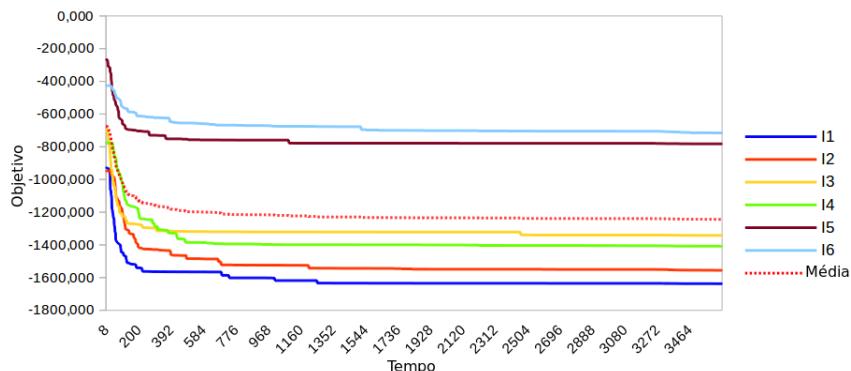
Resultados semelhantes aos apresentados são mostrados na Figura 48 para o modelo PLIM2. Na Figura 48 (a) temos o resultado para o Ψ^m , (b) Ψ^b , (c) Ψ^{s1} e (d) Ψ^{s2} . Nesse modelo, nota-se uma queda ligeiramente mais brusca no valor do objetivo, isso ocorre devido ao seu melhor desempenho em termos de tempo. Percebe-se também que houve uma piora no valor do objetivo, pois essa função possui mais termos penalizados.

Figura 45 – Evolução do objetivo em função do tempo para o problema na superfície aerodinâmica tipo 1 no modelo PLIM1.



Fonte: Elaborada pelo autor.

Figura 46 – Evolução do objetivo em função do tempo para o problema na superfície aerodinâmica tipo 2 no modelo PLIM1.

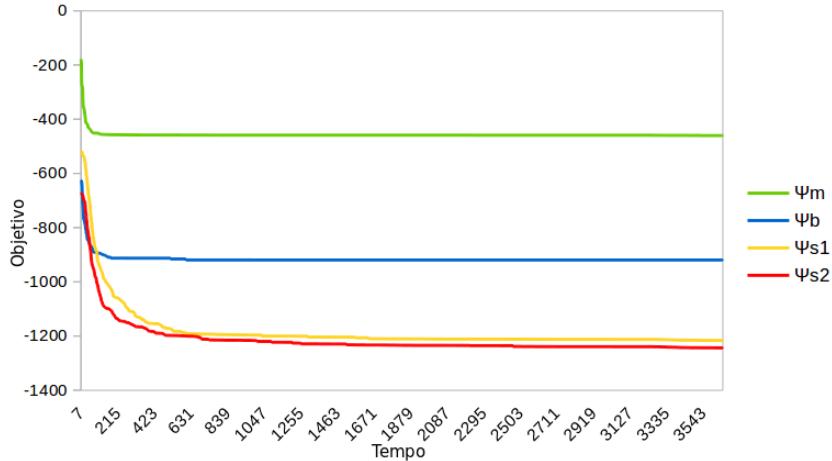


Fonte: Elaborada pelo autor.

No tipo de problema abordado nesta dissertação, estamos lidando com uma situação crítica e necessitamos de uma trajetória rápida para o pouso emergencial. Logo, uma solução factível obtida rapidamente pode ser mais relevante que uma solução ótima que demande maior tempo computacional. Uma possível forma de se obter uma solução rápida é parar a execução do método exato após encontrar a primeira solução factível.

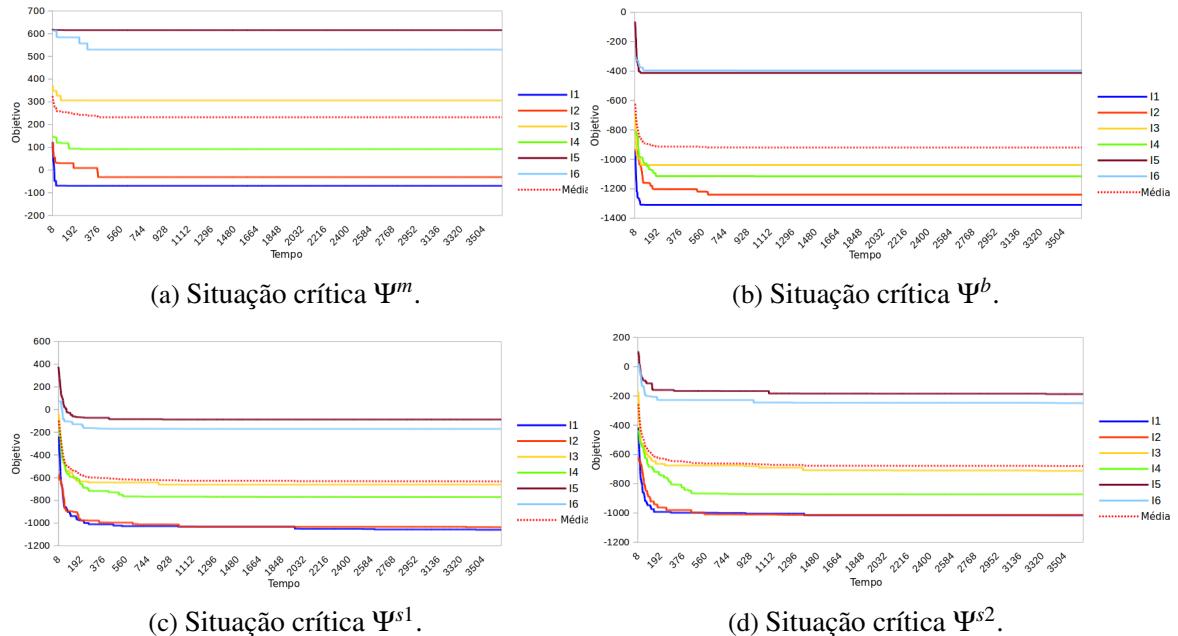
A Tabela 20 apresenta os resultados, onde se pode comparar o tempo que o modelo PLIM2 leva para chegar a primeira e última solução factível. O desvio padrão e intervalo de confiança do tempo também foram mensurados. O intervalo de confiança dessa tabela foi calculado com 95% de confiabilidade.

Analizando a Tabela 20, percebemos que no problema do motor (Ψ^m) e na instância I1 a média de tempo para achar a primeira solução é de 4,76 segundos com um desvio padrão de 1,34. Há um intervalo de confiança entre 4,4 e 5,0 segundos, ou seja, uma solução consideravelmente rápida. O tempo para encontrar a última solução factível é de 63,06 segundos e possui um intervalo de confiança entre 58,9 e 67,1 segundos.

Figura 47 – Evolução do objetivo em função do tempo para as falhas Ψ^m , Ψ^b , Ψ^{s1} e Ψ^{s2} no modelo PLIM1.

Fonte: Elaborada pelo autor.

Figura 48 – Evolução do objetivo em função do tempo para diferentes falhas no modelo PLIM2.



Fonte: Elaborada pelo autor.

Para a utilização do intervalo de confiança necessitamos que os dados sigam uma distribuição normal. Assim, foi feita uma análise considerando a distribuição do tempo para encontrar a primeira solução, ilustrada na Figura 49 (a). Podemos perceber que essa curva se aproxima de uma distribuição normal com média 7,44 e desvio padrão 0,10. Vemos também que tal distribuição é ainda mais conservadora que a normal. Essa análise foi desenvolvida após 5.000 execuções do método PLIM2 em 5.000 mapas gerados do tipo $\mathcal{M}_F^{25\%}$. Como a distribuição do tempo se aproxima de uma distribuição normal, de forma conservativa, então todas as conclusões sobre o intervalo de confiança são também válidas para a Tabela 20 (colunas

Tabela 20 – Análises dos experimentos do PLIM2 para todas as situações críticas e todas a instâncias em relação ao tempo para a primeira e última solução factível.

Ψ	Instância	Tempo para Primeira Solução			Tempo para Última Solução		
		Tempo	σ Tempo	Intervalo de Confiança	Tempo	σ Tempo	Intervalo de Confiança
Ψ^m	I1	4,76	1,34	[4,4 5,0]	63,06	20,90	[58,9 67,1]
	I2	12,36	1,34	[12,1 12,6]	263,08	220,93	[219,7 306,3]
	I3	3,67	0,29	[3,6 3,7]	35,69	13,48	[33,0 38,3]
	I4	15,54	2,55	[15,0 16,0]	185,74	200,02	[146,5 224,9]
	I5	4,66	0,59	[4,5 4,7]	105,88	36,82	[98,6 113,1]
	I6	34,44	8,24	[32,8 36,0]	220,60	108,67	[199,3 241,9]
	Média	11,61	1,64	[11,2 11,9]	141,44	47,95	[132,0 150,8]
Ψ^b	I1	5,79	0,44	[5,7 5,8]	9,32	1,28	[9,0 9,5]
	I2	19,68	2,73	[19,1 20,2]	48,99	11,47	[46,7 51,2]
	I3	7,61	1,08	[7,4 7,8]	12,02	3,21	[11,3 12,6]
	I4	23,20	2,68	[22,6 23,7]	56,13	22,07	[51,8 60,4]
	I5	6,41	0,80	[6,2 6,5]	7,98	1,04	[7,7 8,1]
	I6	19,83	2,16	[19,4 20,2]	60,61	34,71	[53,8 67,4]
	Média	13,74	0,93	[13,5 13,9]	32,46	9,87	[30,5 34,4]
Ψ^{s1}	I1	28,37	21,52	[24,1 32,5]	446,75	358,90	[376,4 517,1]
	I2	28,61	2,90	[28,0 29,1]	402,33	91,68	[384,3 420,3]
	I3	14,66	2,27	[14,2 15,1]	201,98	102,58	[181,8 222,0]
	I4	35,50	5,45	[34,4 36,5]	645,32	168,40	[612,3 678,3]
	I5	10,64	2,28	[10,2 11,0]	229,79	103,23	[209,5 250,0]
	I6	35,02	5,95	[33,8 36,1]	486,76	128,85	[461,5 512,0]
	Média	25,42	4,08	[24,6 26,2]	401,53	55,47	[390,6 412,4]
Ψ^{s2}	I1	9,50	1,13	[9,2 9,7]	426,70	151,38	[397,0 456,3]
	I2	23,45	7,31	[22,0 24,8]	523,16	170,76	[489,6 556,6]
	I3	20,72	5,51	[19,6 21,8]	460,62	122,77	[436,5 484,6]
	I4	49,18	19,54	[45,3 53,0]	591,68	164,82	[559,3 623,9]
	I5	10,72	2,38	[10,2 11,1]	369,07	119,36	[345,6 392,4]
	I6	60,11	10,73	[58,0 62,2]	291,18	266,81	[238,8 343,4]
	Média	28,77	2,78	[28,2 29,3]	444,90	72,99	[430,5 459,2]

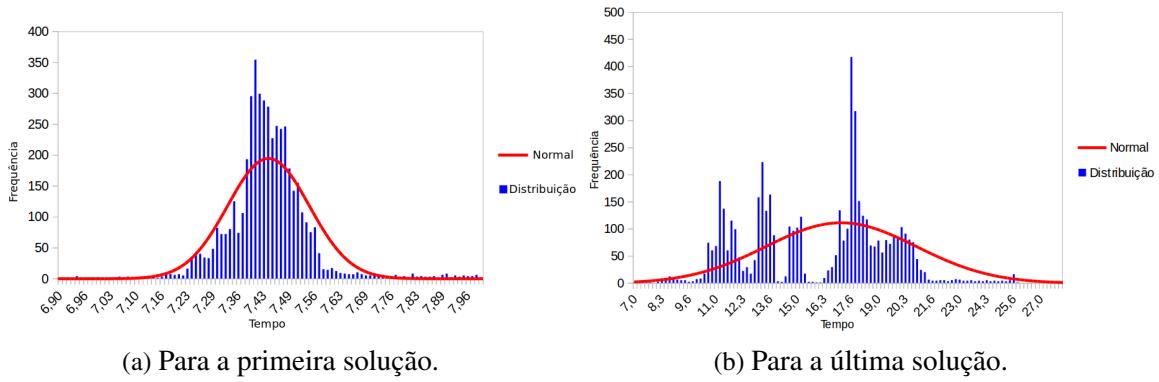
Fonte: Dados da pesquisa.

Tempo para Primeira Solução).

No entanto, a Figura 49 (b) apresenta uma distribuição que não se aproxima de uma distribuição normal. Dessa maneira, as informações sobre o intervalo de confiança na Tabela 20 (colunas Tempo para Última Solução) podem não ser válidas. Essa análise nos apresenta uma informação importante: o método exato fornece a sua primeira solução em uma distribuição de tempo bem comportada, contudo, não existe garantias de quando será encontrado a solução final pelo método.

A Tabela 21 apresenta os resultados do valor da função objetivo da primeira e última solução factível, seu desvio padrão e intervalo de confiança. O intervalo de confiança foi mensurado com 95% de confiabilidade. No problema do motor, o valor médio encontrado para a primeira solução nos grupos I1 até I6 é sempre positivo, ou seja, há penalização indicando que a maioria das soluções pousaram sobre a região remanescente. Enquanto a solução final, em média, recai em zonas bonificadoras em I1 e I2 sinalizada pelo custo negativo. O problema na bateria apresentou um bom resultado já na primeira solução do PLIM2, visto que, em todas as instâncias o custo foi negativo. Já nas falhas Ψ^{s1} e Ψ^{s2} houve em média um custo negativo nas instâncias fácil e normal ainda na primeira solução.

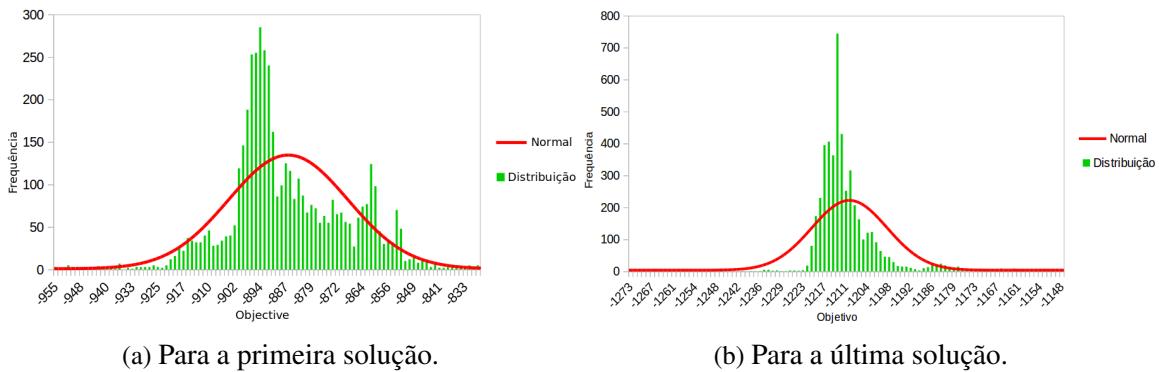
Figura 49 – Análise de distribuição do tempo.



Fonte: Elaborada pelo autor.

Um estudo sobre o formato da distribuição do objetivo para a primeira solução também foi feito. Os dados seguem uma distribuição normal como apresentado na Figura 50 (a). Embora menos conservador do que a distribuição do tempo, pode-se considerar uma distribuição similar a normal. Nesse caso, a média foi de $-885,68$ e o desvio padrão foi de $17,78$. A Figura 50 (b) mostra a distribuição do objetivo para a última solução. Há uma distribuição altamente conservadora em relação a normal. Pela característica dos formatos da distribuição do objetivo, as conclusões sobre o intervalo de confiança utilizadas nessa tabela podem ser consideradas válidas.

Figura 50 – Análise de distribuição do objetivo.



Fonte: Elaborada pelo autor.

Alguns exemplos de rotas geradas pelo PLIM1 são mostradas na Figura 51. Essas rotas descrevem o comportamento da aeronave para diferentes situações críticas. Para facilitar a visualização, as rotas são apresentadas em um cenário bidimensional. A Figura 52 exemplifica a mesma rota da Figura 51 (c) no cenário tridimensional.

Uma etapa importante é efetuar a validação das rotas geradas pelo modelo, mostrando assim, que uma aeronave consegue segui-las e efetuar o pouso com segurança. Uma forma de se efetuar tal análise é através de um simulador de voo como descrito a seguir.

Tabela 21 – Análises dos experimentos do PLIM2 para todas as situações críticas e todas a instâncias em relação ao objetivo para a primeira e última solução factível.

Ψ	Instância	Objetivo para Primeira Solução			Objetivo para Última Solução		
		\mathcal{W}	$\sigma \mathcal{W}$	Intervalo de Confiança	\mathcal{W}	$\sigma \mathcal{W}$	Intervalo de Confiança
Ψ^m	I1	123,03	97,55	[103,9 142,1]	-69,55	123,85	[-93,8 -45,2]
	I2	119,62	170,51	[86,2 153,0]	-31,16	252,96	[-80,7 18,4]
	I3	372,35	123,06	[348,2 396,4]	306,09	114,97	[283,5 328,6]
	I4	147,37	210,24	[106,1 188,5]	91,84	204,80	[51,7 131,9]
	I5	620,52	79,57	[604,9 636,1]	615,19	80,65	[599,3 631,0]
	I6	613,06	91,25	[595,1 630,9]	529,46	94,24	[510,9 547,9]
	Média	325,28	73,95	[310,7 339,7]	232,06	90,19	[214,3 249,7]
Ψ^b	I1	-948,45	204,33	[-988,5 -908,4]	-1309,41	8,84	[-1311,1 -1307,6]
	I2	-935,30	96,69	[-954,2 -916,3]	-1240,47	62,65	[-1252,7 -1228,2]
	I3	-689,49	188,97	[-726,5 -652,4]	-1038,44	108,07	[-1059,6 -1017,2]
	I4	-814,55	122,42	[-838,5 -790,5]	-1115,36	88,38	[-1132,6 -1098,0]
	I5	-66,14	143,26	[-94,2 -38,0]	-412,34	170,18	[-445,6 -378,9]
	I6	-303,47	152,71	[-333,4 -273,5]	-396,07	146,75	[-424,8 -367,3]
	Média	-626,77	70,86	[-640,6 -612,8]	-919,55	54,00	[-930,1 -908,9]
Ψ^{s1}	I1	-242,20	230,79	[-287,4 -196,9]	-1059,11	133,25	[-1085,2 -1033,0]
	I2	-578,46	192,48	[-616,1 -540,7]	-1036,46	88,61	[-1053,8 -1019,0]
	I3	-37,51	179,38	[-72,6 -2,3]	-661,40	87,22	[-678,4 -644,3]
	I4	-183,83	134,80	[-210,2 -157,4]	-770,88	75,11	[-785,6 -756,1]
	I5	377,21	109,23	[355,8 398,6]	-87,68	188,13	[-124,5 -50,8]
	I6	73,10	245,35	[25,0 121,1]	-170,53	263,58	[-222,1 -118,8]
	Média	-99,29	69,39	[-112,8 -85,6]	-633,05	80,70	[-648,8 -617,2]
Ψ^{s2}	I1	-420,83	270,43	[-473,8 -367,8]	-1015,80	174,32	[-1049,9 -981,6]
	I2	-626,96	113,28	[-649,1 -604,7]	-1013,14	147,23	[-1041,9 -984,2]
	I3	-173,96	225,27	[-218,1 -129,8]	-712,05	133,01	[-738,1 -685,9]
	I4	-440,39	152,16	[-470,2 -410,5]	-872,56	165,82	[-905,0 -840,0]
	I5	100,66	122,70	[76,6 124,7]	-187,27	117,92	[-210,3 -164,1]
	I6	13,87	95,65	[-4,8 32,6]	-249,02	133,11	[-275,1 -222,9]
	Média	-260,39	117,80	[-283,4 -237,3]	-678,69	84,57	[-695,2 -662,1]

Fonte: Dados da pesquisa.

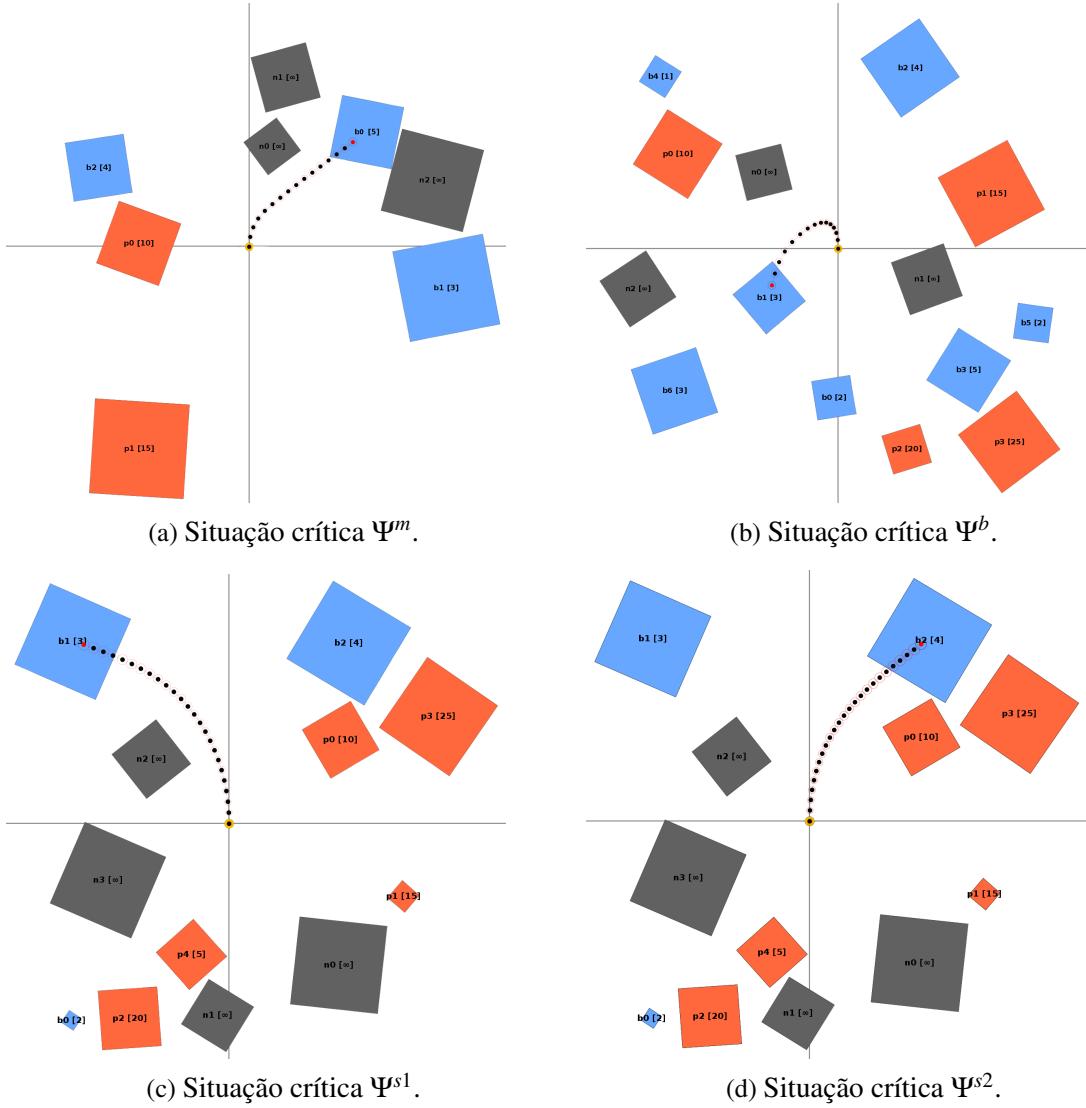
5.3.4 Experimentos Online

Os experimentos desta seção utilizaram o simulador FlightGear (FG). A Figura 53 mostra um voo utilizando o FG a esquerda e a direita um mapa com a rota da aeronave. Todos os experimentos conduzidos nesta seção são relativos ao PLIM2.

Conforme já mencionado, a aeronave Cessna 172 foi utilizada para atuar como VANT e os mapas gerados nos testes anteriores foram ampliados 25 vezes para se adequarem à dinâmica da aeronave. Um mapa de cada conjunto de instância foi selecionado, totalizando assim 6 mapas. Para cada mapa, uma falha diferente foi simulada permitindo executar 24 diferentes experimentos com o FG. A Tabela 22 apresenta os resultados obtidos como o tempo gasto em segundos na simulação, o erro horizontal médio em metros (E_{medio}^h) e o desvio padrão do erro (σE_{medio}^h) também medido em metros. Ainda nessa tabela comparamos os desvios da rota com simulações sem vento e com vento, onde nesse caso o vento estava orientado no sentido norte.

A Figura 54 (a) mostra um exemplo da rota de pouso, onde a posição da aeronave é dada por P_t . O próximo *waypoint* da rota é W_t e o *waypoint* anterior da rota é W_{t-1} . Esses *waypoints* são utilizados pelo controlador (AP) da aeronave. A Figura 54 (b) fornece maiores detalhes da

Figura 51 – Rotas de pouso obtidas para o modelo PLIM2.



Fonte: Elaborada pelo autor.

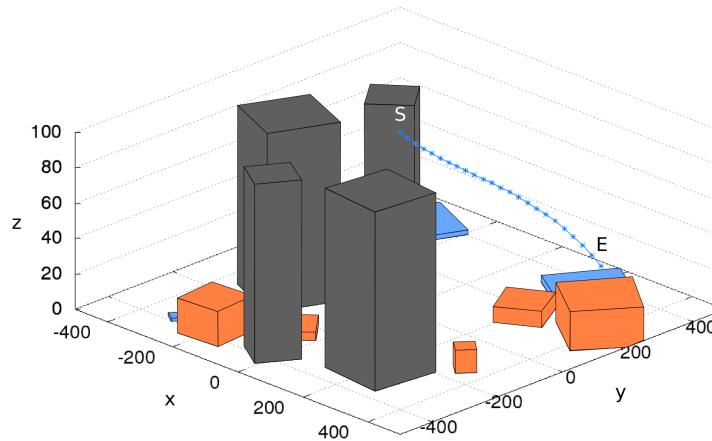
região destacada em (a), onde temos a projeção de P_t sobre o seguimento formado por W_t e W_{t-1} . Essa projeção é formalizada na Equação (5.2), onde os vetores \vec{R}_t e \vec{U}_t estão definidos nas equações (5.3) e (5.4).

$$Q_t = \text{proj}_{\vec{R}_t} \vec{U}_t = \frac{\vec{U}_t \cdot \vec{R}_t}{||\vec{R}_t||^2} \times \vec{R}_t \quad (5.2)$$

$$\vec{R}_t = W_t - W_{t-1} \quad (5.3)$$

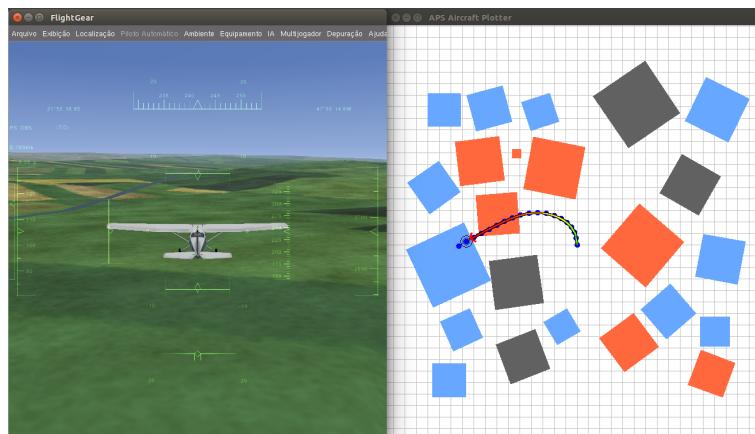
$$\vec{U}_t = P_t - W_{t-1} \quad (5.4)$$

Figura 52 – Rota obtida na situação crítica Ψ^{s1} em ambiente tridimensional.



Fonte: Elaborada pelo autor.

Figura 53 – Simulação usando o FlightGear seguindo uma rota gerada pelo método PLIM.



Fonte: Elaborada pelo autor.

Ainda na Figura 54 (b), temos o erro horizontal E_t^h em um dado instante t , definido na Equação 5.6, que é calculado através da distância entre a posição da aeronave (P_t) e a sua projeção na rota emergencial (P'_t). Por fim, o erro horizontal médio (E_{medio}^h), Equação (5.7), é dado pela média dos erros horizontais em cada instante de tempo (T).

$$P'_t = W_{t-1} + Q_t \quad (5.5)$$

$$E_t^h = dist(P_t, P'_t) = ||P_t - P'_t|| \quad (5.6)$$

$$E_{medio}^h = \frac{1}{T} \cdot \sum_{t=1}^T E_t^h \quad (5.7)$$

Tabela 22 – Resultados obtidos nos experimentos utilizando simulador de voo com o modelo PLIM2.

Ψ	Mapa	Sem Vento			Com Vento [10 knots]		
		Tempo (T)	E_{medio}^h	σE_{medio}^h	Tempo (T)	E_{medio}^h	σE_{medio}^h
Ψ^m	I1-18	161	19,57	14,06	156	22,46	16,78
	I2-13	199	21,68	9,48	178	24,78	9,46
	I3-08	162	19,02	12,84	161	24,41	13,93
	I4-20	212	15,12	12,36	166	21,89	17,06
	I5-24	172	12,10	13,98	168	25,27	12,80
	I6-31	187	9,31	8,63	163	27,39	12,62
	Média	182,17	16,13	-	165,33	24,37	-
Ψ^b	I1-18	121	18,52	14,96	130	18,52	15,36
	I2-13	229	7,80	10,78	221	24,50	14,26
	I3-08	88	15,63	14,79	89	34,89	19,71
	I4-20	133	21,58	20,22	133	23,79	27,72
	I5-24	213	13,46	11,43	224	22,91	13,21
	I6-31	151	8,61	9,31	161	27,89	11,57
	Média	155,83	14,27	-	159,67	25,42	-
Ψ^{s1}	I1-18	149	8,00	8,80	152	28,53	11,33
	I2-13	229	8,16	11,22	222	24,72	14,49
	I3-08	91	15,56	14,23	92	36,01	18,67
	I4-20	113	11,02	11,14	115	33,37	14,34
	I5-24	81	19,85	20,21	76	38,07	25,54
	I6-31	151	8,00	8,57	162	28,18	12,01
	Média	135,67	11,76	-	136,50	31,48	-
Ψ^{s2}	I1-18	119	18,55	14,28	130	27,96	15,29
	I2-13	82	21,96	20,27	88	29,23	24,01
	I3-08	74	25,25	22,14	76	34,03	25,14
	I4-20	139	19,74	15,98	140	21,10	21,77
	I5-24	211	13,30	11,36	224	22,77	13,06
	I6-31	100	25,26	19,81	100	25,74	29,09
	Média	120,83	20,68	-	126,33	26,81	-
	Média Final	148,63	15,71	-	146,96	27,02	-

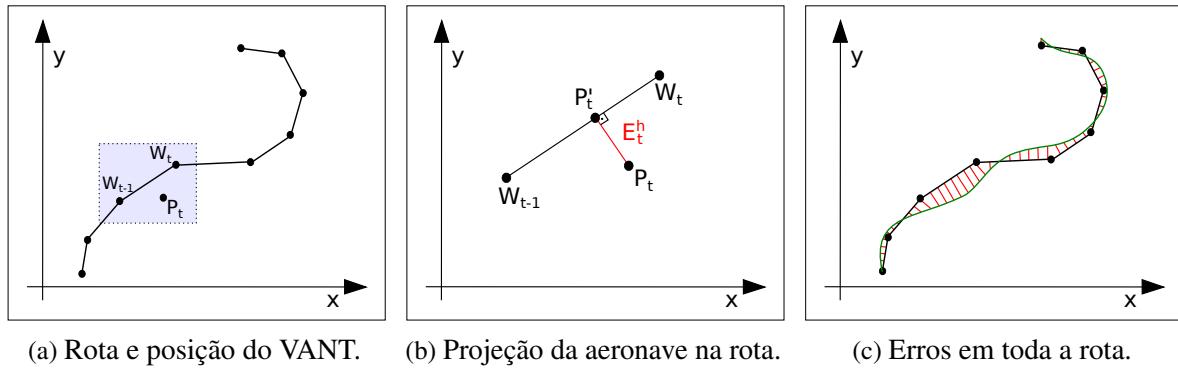
Fonte: Dados da pesquisa.

A Figura 54 (c) apresenta a rota calculada pelo método (rota na cor preta), a rota seguida pela aeronave (rota na cor verde) e os erros horizontais em cada instante. Esses erros são dados pelas distâncias entre essas rotas (seguimentos de reta na cor vermelha).

Com base nessas métricas, podemos compreender melhor a Tabela 22. Nessa tabela, o mapa I1-18 indica que se trata do 18º mapa do grupo de instância I1. Analisando o problema no motor (Ψ^m) nesse mapa, sem a ocorrência de vento, percebemos que o voo durou 161 segundos com um erro horizontal médio de 19,57 metros.

Analizar o resultado da mesma simulação de falha, mas com vento associado, o tempo de voo foi de 156 segundos. O erro horizontal médio aumentou, passando de 19,57 metros para 22,46 metros. Em geral, os erros horizontais sem vento foram sempre menores ou iguais aos erros das simulações com vento. Esse resultado corrobora com o senso comum, onde a existência de vento causa uma perturbação na posição da aeronave que é proporcional a sua intensidade.

Figura 54 – Cálculo do erro horizontal da aeronave em relação a rota gerada.



Fonte: Elaborada pelo autor.

O tempo de voo com e sem vento são bastante similares, com um resultado médio de 148,63 segundos sem vento e 146,96 segundos com vento.

A Figura 55 (a) apresenta a distribuição do erro horizontal obtido nas simulações sem vento. Essa distribuição se comporta de forma conservadora em relação a distribuição Gaussiana. Nessa distribuição, um erro positivo indica um deslocamento da aeronave a direita da rota planejada, já um erro negativo indica um deslocamento a esquerda da rota.

A Figura 55 (b) mostra o formato da distribuição do erro com vento. Nessa simulação, o deslocamento horizontal da aeronave não segue a distribuição normal. Uma das premissas básicas deste trabalho, baseada em Blackmore, Ono e Williams (2011), é que a posição da aeronave segue uma distribuição Gaussiana com $\mathcal{N}(0, Q)$ com média zero e matriz de covariância Q . No entanto, esse experimento revelou que a presença de ventos pode violar tal condição.

Um estudo mais apurado sobre o comportamento da aeronave diante a presença de ventos, com diferentes intensidades e direções, ainda se faz necessário. Apesar disso, os experimentos conseguiram gerar resultados satisfatórios, conforme reportado nas seções anteriores. Essas distribuições foram obtidas a partir do processamento do erro em todas as 24 simulações da Tabela 22.

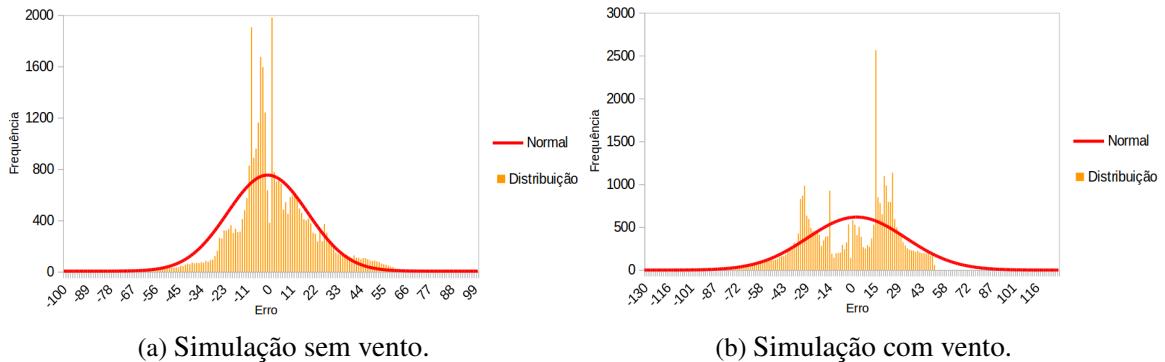
Na Figura 56, são apresentados dois exemplos de rotas obtidos nos experimentos. Mesmo após ampliar essas imagens, não há grandes erros horizontais em relação a rota original. Observe que as duas rotas planejadas e executadas estão sobrepostas. Dois vídeos com as simulações realizadas estão disponíveis no site⁴ e no site⁵.

Os voos utilizando o simulador de voo mostram a robustez das rotas geradas pelo método PLIM, uma vez que a aeronave conseguiu seguir as rotas e manter um erro horizontal reduzido.

⁴ <https://youtu.be/rRJVArhk77c>

⁵ <https://youtu.be/RBZBMkYsCqM>

Figura 55 – Distribuição do erro horizontal obtidos nas simulações.



Fonte: Elaborada pelo autor.

5.3.5 Experimentos com Rotas Pré-Processadas

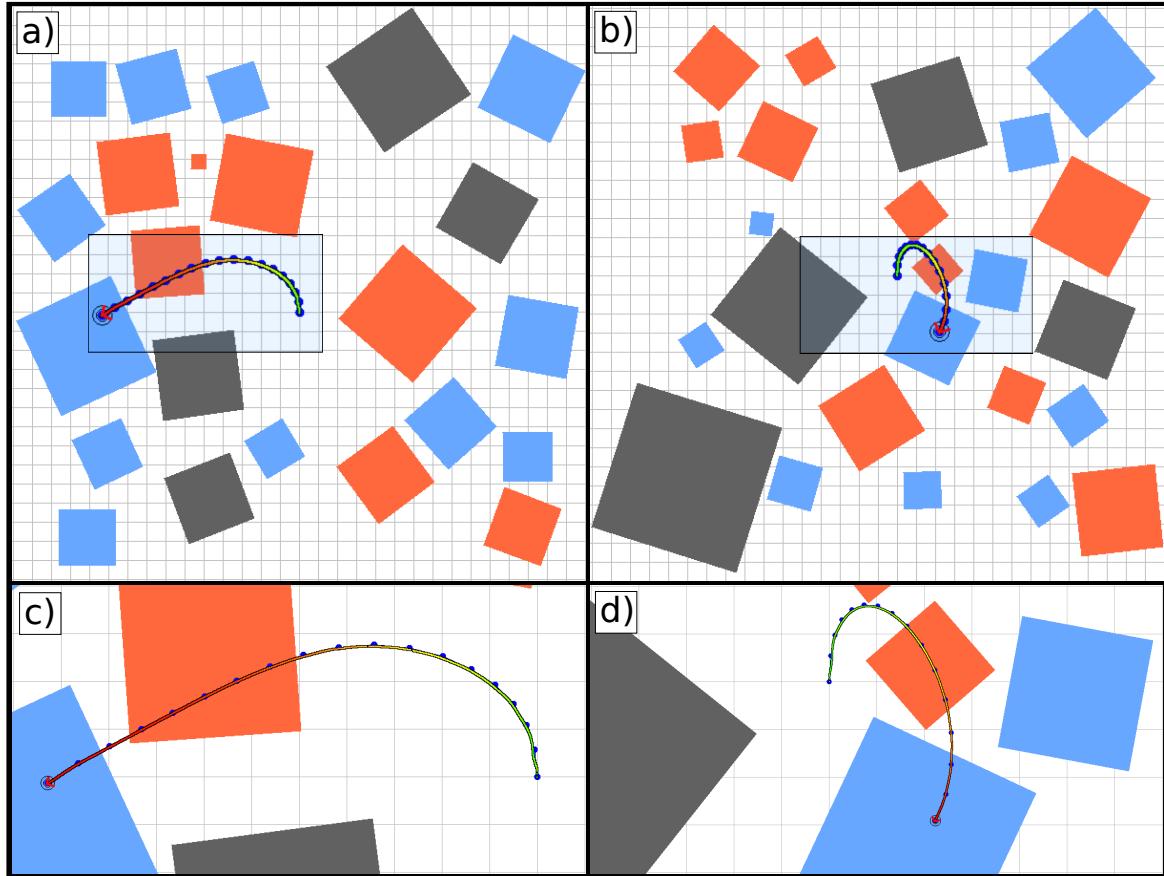
Esta seção descreve os experimentos considerando que as rotas para possíveis pouso emergenciais foram previamente definidas conforme descrito na Seção 4.6. Definiu-se que a missão efetuada pela aeronave seria subdividida em 20 pontos-chave, onde para cada um desses pontos foi avaliada a melhor rota emergencial dada uma situação crítica na aeronave.

A Figura 57 (a) mostra um conjunto de resultados de pouso emergencial no caso de problema na bateria (Ψ^b). As rotas foram calculadas para cada um dos pontos-chave marcados em vermelho. O solver CPLEX conseguiu atingir a solução ótima para cada um dos pontos-chave, pousando sempre em região bonificadora em 20 das 20 simulações dessa falha. A quantidade média de *waypoints* gastos foram de 11,5 para definir a rota emergencial. O tempo gasto pelo método foi de 28 minutos para processar todas as 20 rotas definidas.

A Figura 57 (b) mostra o caso do problema na superfície aerodinâmica tipo 1 (Ψ^{s1}). O modelo conseguiu pousar a aeronave em uma região bonificadora nas 18 das 20 simulações de falha grave. A quantidade média de *waypoints* usados foi de 15,8 e o tempo gasto foi de 83 minutos. O resultado para o problema na superfície aerodinâmica tipo 2 (Ψ^{s2}) é mostrado na Figura 57 (c). Nessa figura, tem-se que cinco soluções caíram em região remanescente. Essas soluções partem do 6º, 7º, 8º, 18º e 19º ponto-chave, respectivamente. O tempo gasto nesse processamento foi de 197 minutos. Em média foram alocados 18,1 *waypoints* até o pouso da aeronave.

O caso de falha motor foi o mais problemático, onde em apenas 5 pontos-chave foi possível trazer a aeronave ao solo e pousar em região bonificadora. Esse resultado pode ser visto na Figura 57 (d). Os outros 15 pontos-chave pousaram em região remanescente. O tempo gasto pelo método foi de 339 minutos. A quantidade de *waypoints* usados foram de 16,4. Estudando as soluções que pousaram em região remanescente, observa-se que há um crescimento da incerteza até atingir Φ_1^b . Ao partir dos pontos-chave nessas rotas, a incerteza da posição cresceu muito ao se aproximar da região bonificadora. As rotas nas figuras mostram apenas a posição esperada do

Figura 56 – Exemplos de rotas obtidas nos experimentos no FlightGear. (a) e (b) rotas em diferentes mapas com falhas na bateria. Em (c) tem-se a rota (a) ampliada. Em (d) tem-se a rota (b) ampliada.



Fonte: Elaborada pelo autor.

VANT, não está sendo mostrado o crescimento da incerteza da aeronave.

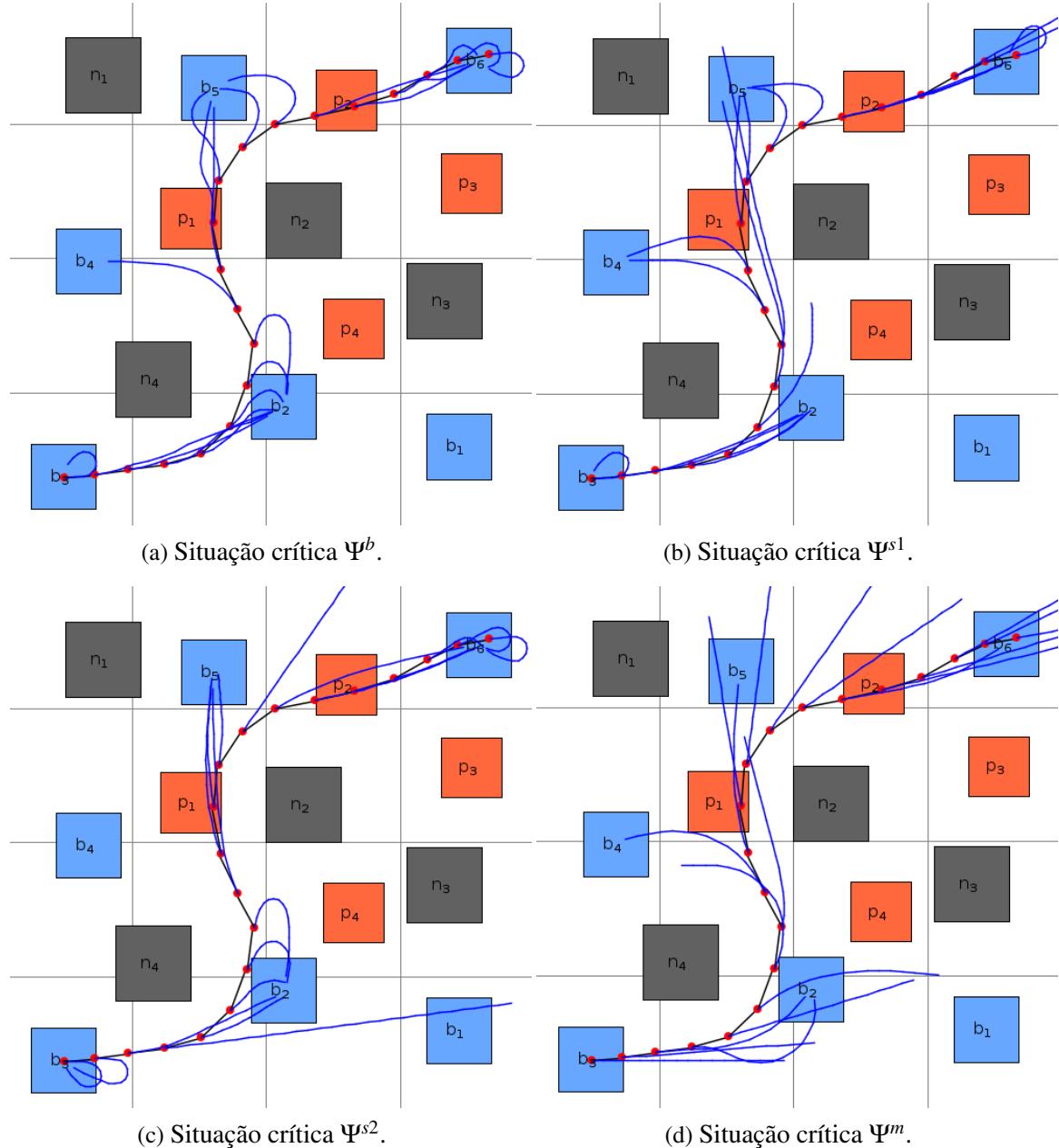
A Tabela 23 mostra uma síntese dos resultados obtidos usando o modelo PLIM2 para as rotas pré-processadas. Essa tabela mostra para cada falha o local de pouso, tempo de processamento, valor do objetivo entre outras informações.

Tabela 23 – Resultados obtidos para as rotas pré-processadas pelo modelo PLIM2.

Ψ	Ótima	Factível	Infactível	Desconhecida	\mathcal{W}	Tempo	Pousou Φ^b	Pousou Φ^r
Ψ^m	16	4	0	0	-408,99	339 min	5	15
Ψ^b	20	0	0	0	-1373,54	28 min	20	0
Ψ^s_1	20	0	0	0	-1655,57	83 min	18	2
Ψ^s_2	17	3	0	0	-1368,27	197 min	15	5
Média	18,25	1,75	0,00	0,00	-1201,59	161 min	14,50	5,50

Fonte: Dados da pesquisa.

Figura 57 – Resultados das simulações pré-processadas em uma série de pontos-chave da rota.



Fonte: Elaborada pelo autor.

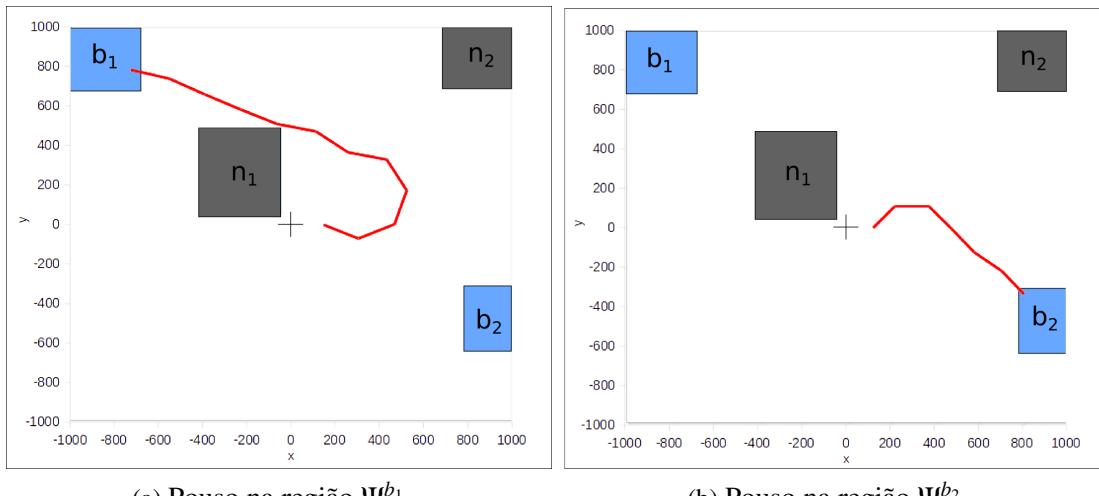
5.4 Experimentos Usando SCADE

Uma implementação preliminar da HG usando SCADE foi realizada em parceria com o pesquisador André Pierre. Esse método foi escolhido por ter uma implementação simples e possuir um baixo tempo de processamento. O SCADE é uma ferramenta da empresa Esterel para desenvolvimento baseado em modelo para softwares críticos (ESTEREL, 2015). Essa ferramenta é capaz de gerar código para a linguagem C através de modelos. O mesmo pode ser integrado com outras aplicações, usado em prototipagem, simulação, verificação, validação e gerador de código certificado com DO-178B (SCADE, 2015).

A Figura 59 mostra um diagrama de estados da implementação efetuada. Nesse diagrama, cada estado representa um operador que, por sua vez, podem ser compostos por outros operadores. Há cinco estados principais que são Setup, RouteCalc, compareFit, Error e End. De forma geral, o estado Setup fornece as configurações e parâmetros de entrada para o método; o estado RouteCalc é o principal responsável pelo funcionamento da HG, calculando e avaliando a rota; o estado compareFit é o responsável pela comparação dos *fitness* e retorna a melhor rota; o estado Error processa possíveis erros de rotas, em geral, rotas que excedem o total de *waypoints*; e o estado End indica o término do processamento na máquina de estados.

Na implementação em SCADE, algumas simplificações foram feitas nessa versão preliminar da HG. Não foram consideradas regiões penalizadoras, apenas não navegáveis e bonificadoras. O número de regiões bonificadoras e não navegáveis foi limitado a quatro. Os polígonos estão restritos a quadrados e o número de *waypoints* foi limitado a 20. Não foi incluída a incerteza associada a aeronave e apenas as equações (4.3), (4.5) e (4.7) da função de avaliação desenvolvida foram implementadas.

Figura 58 – Exemplo de rotas obtidas pelo SCADE.

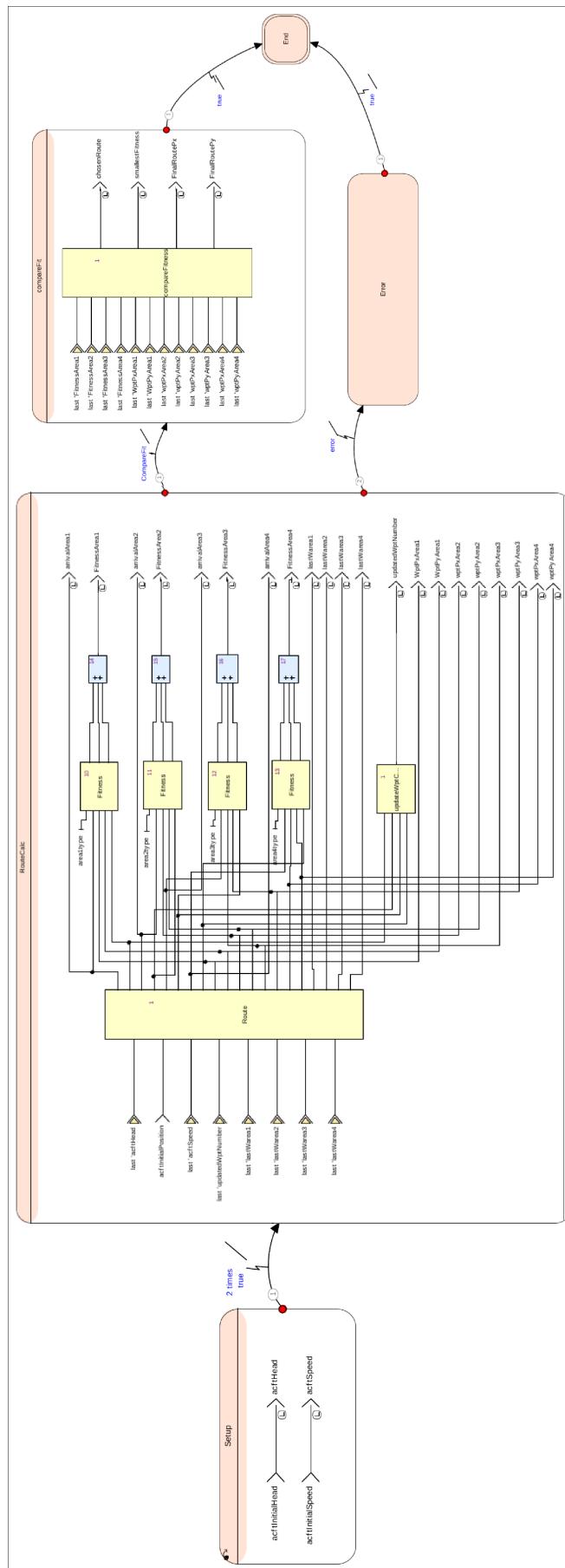


Fonte: Elaborada pelo autor.

A Figura 58 apresenta alguns exemplos de rotas obtidas com a implementação em SCADE. Após a avaliação dessas duas rotas, a melhor é escolhida pelo método. Nesse exemplo, rota da Figura 58 (b) foi selecionada já que é a menor e leva a uma região bonificadora, sem violar zonas não navegáveis. As configurações iniciais da simulação são velocidade inicial $10m/s$ e posição inicial de $(125, 0)$. Um total de oito simulações foram efetuadas usando essa ferramenta. Assim, uma validação preliminar da HG foi conduzida variando os parâmetros de velocidade inicial e posição inicial da aeronave.

Esses experimentos efetuados abrem caminho para a implementação embarcada, uma vez que o SCADE é destinado a aplicações embarcadas.

Figura 59 – Máquina de estados implementada usando o SCADE.



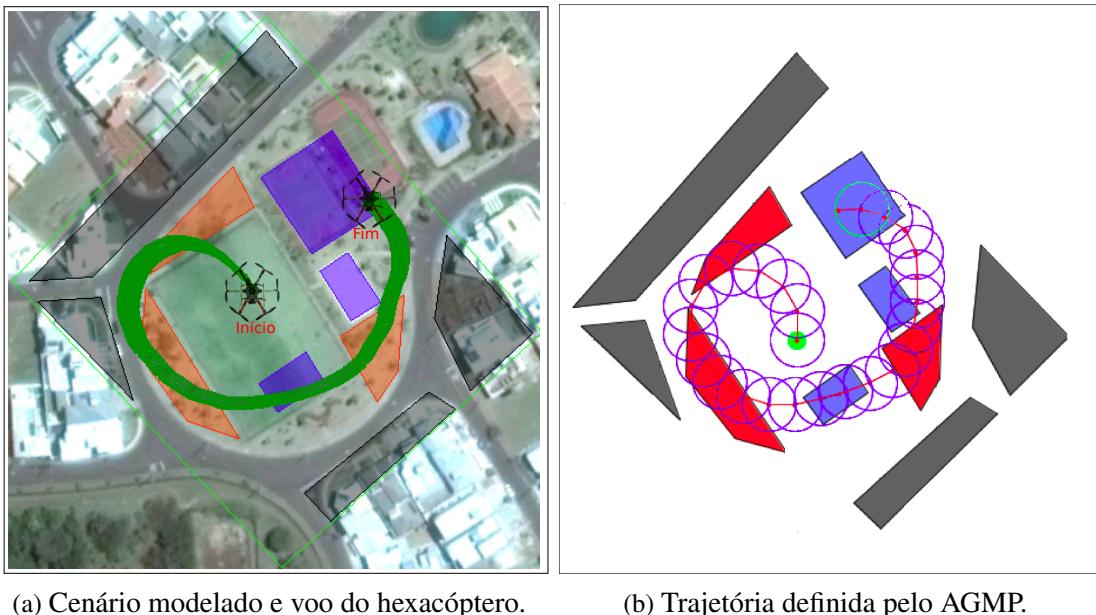
Fonte: Elaborada pelo pesquisador André Pierre.

5.5 Experimentos com Voo Real

Alguns experimentos preliminares envolvendo um hexacóptero foram conduzidos. A Figura 60 (a) mostra o cenário, modelado no Google Earth, que foi utilizado no voo real. As regiões em cinza representam zonas não navegáveis, em laranja temos regiões penalizadoras e em azul regiões bonificadoras. A rota de fato seguida pela aeronave está destacada em verde.

A Figura 60 (b) exibe as regiões do mapa projetadas na Figura 60 (a). A trajetória que a aeronave deveria seguir, retornada pelo Algoritmo Genético Multi-Populacional (AGMP) desenvolvido, também é descrita. A rota definida pelo AGMP realiza o pouso considerando uma falha simulada no hexacóptero, ou seja, essa falha não ocorreu de fato. Nessa falha, é suposto que a aeronave consegue fazer curvas somente para a esquerda Ψ^s ¹. O método AGMP com inicialização gulosa (AGMP2) foi o método escolhido nesse experimento por apresentar bons resultados, no entanto, os outros métodos poderiam também ser utilizados.

Figura 60 – Experimento com voo real usando um Hexacóptero.



(a) Cenário modelado e voo do hexacóptero.

(b) Trajetória definida pelo AGMP.

Fonte: Elaborada pelo autor.

Ao analisar as rotas das Figuras 60 (a) e (b), a aeronave é capaz de seguir de forma bastante aproximada a trajetória definida pelo AGMP. Os erros da trajetória, em geral, são devido a incertezas associadas à posição da aeronave, obtidas pelo sensor GPS.

Esse experimento foi realizado utilizando um VANT de asa rotativa (hexacóptero), no entanto, o modelo de dinâmica foi construído para um VANT de asa fixa. Apesar disso, tal experimento pode ser justificado, pois um hexacóptero consegue simular a dinâmica de um VANT de asa fixa, embora o contrário não seja verdade.

5.6 Síntese dos Resultados

Uma síntese dos resultados gerais obtidas nesta dissertação é descrita nesta seção. A Tabela 24 agrupa as soluções obtidas pelos diferentes métodos desenvolvidos. Os resultados são subdivididos em duas categorias de acordo com a modelagem restrita e completa. De forma geral, estão sintetizados os resultados médios dos experimentos. A coluna Tempo está em segundos (s), a coluna Pouso Φ^b apresenta a porcentagem de pouso sobre regiões bonificadoras, a coluna Ótima lista os métodos que retornam soluções ótimas, a coluna Determinístico lista aqueles com tal característica e a coluna Tipo contem informações sobre os algoritmos implementados.

Tabela 24 – Síntese dos resultados para todos os métodos desenvolvidos na modelagem restrita e completa.

Modelagem	Método	Tempo	Pousou Φ^b	Ótima	Determinístico	Tipo
Restrita	HG	0,071	77,83	Não	Sim	Heurístico
	AG1	0,955	77,96	Não	Não	Metaheurístico
	AG2	0,912	89,86	Não	Não	Metaheurístico
	AGMP1	1,017	80,13	Não	Não	Metaheurístico
	AGMP2	0,874	89,53	Não	Não	Metaheurístico
Completa	PLIM1	1043,64	60,38	Sim	Sim	Exato
	PLIM2	255,31	60,29	Sim	Sim	Exato

Fonte: Dados da pesquisa.

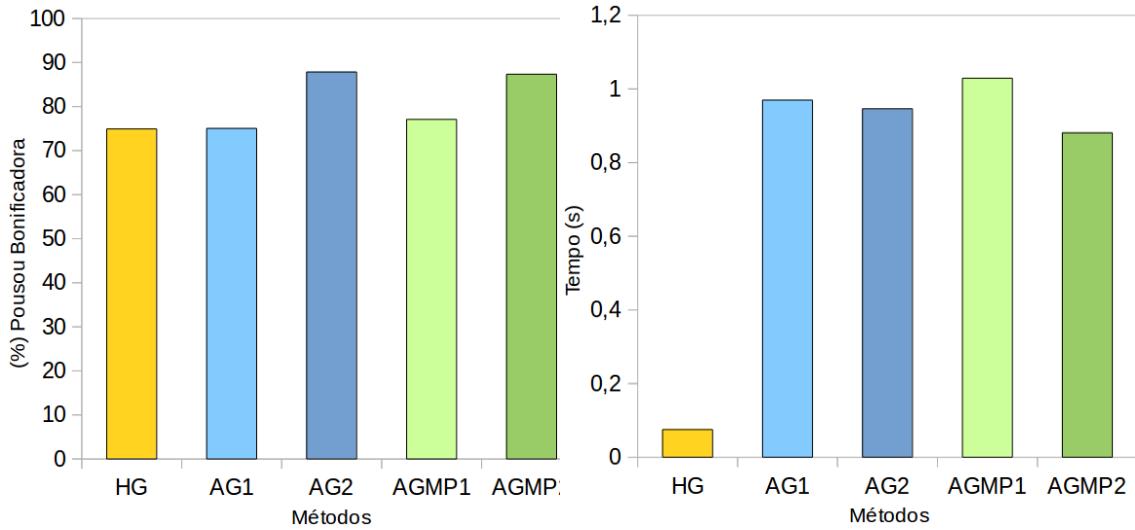
Os resultados obtidos nessa dissertação formam também sintetizados através de gráficos, onde consegue-se melhor visualizar as informações a respeito dos métodos. A Figura 61 agrupa os resultados obtidos para a modelagem restrita do problema. Já na Figura 62 estão reunidos os resultados da modelagem completa do problema. De forma geral, na modelagem restrita, percebe-se o ganho de tempo pela HG, já em termos de qualidade de solução tem-se um melhor desempenho pelos métodos que incorporaram a inicialização gulosa (AG2 e AGMP2). Em geral, na modelagem completa, houve um ganho de tempo pelo PLIM2, no entanto, no quesito qualidade da solução houve um empate entre o método PLIM1 e PLIM2.

5.7 Conclusão

Este capítulo apresentou os resultados obtidos nesta dissertação. De forma geral, a HG apresentou o melhor resultado em termos de tempo computacional, mas o pior em termos de qualidade de solução. Ocorreu um desempenho bastante similar entre os métodos AG e AGMP. Portanto, a ideia de ter múltiplas populações e migração entre elas não auxilia na melhoria da qualidade das soluções. No entanto, investir em diferentes operadores pode dar bons resultados como mostrado pela inicialização gulosa.

O método baseado em PLIM apresentou um custo elevado em termos de tempo computacional. Porém, uma melhoria no seu desempenho foi alcançada ao mudar a função objetivo para

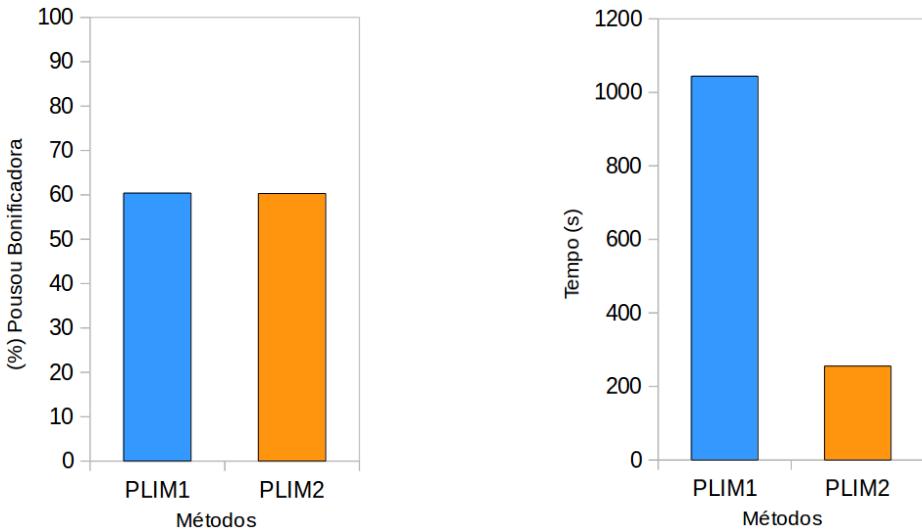
Figura 61 – Síntese dos resultados para a modelagem restrita do problema.



- (a) Comparação da qualidade da solução em termos de pouso em região bonificadora. (b) Comparação da qualidade da solução em termos de tempo de processamento.

Fonte: Elaborada pelo autor.

Figura 62 – Síntese dos resultados para a modelagem completa do problema.



- (a) Comparação da qualidade da solução em termos de pouso em região bonificadora. (b) Comparação da qualidade da solução em termos de tempo de processamento.

Fonte: Elaborada pelo autor.

reduzir a alocação de *waypoints*. Apesar da melhoria alcançada, por se tratar de uma aplicação crítica, resultados em baixo tempo computacional são necessários. Nesse sentido, a ideia de calcular antecipadamente as rotas usando a estratégia exata foi proposta.

Devido as diferentes formas de tratar a dinâmica da aeronave na modelagem restrita e completa, não se pode fazer uma comparação direta entre a abordagem exata e a abordagem heurística.



CONSIDERAÇÕES FINAIS

“O futuro dependerá daquilo que fazemos no presente.”

Mohandas Karamchand Gandhi

6.1 Conclusões

Este trabalho apresentou um problema de replanejamento de caminho para pousar um Veículo Aéreo Não Tripulado (VANT) com algum tipo de falha nos equipamentos, denominada aqui como situação crítica. Assume-se que a aeronave executa uma missão dentro de um cenário a partir do qual uma situação crítica acontece. Logo, um algoritmo planejador de rotas é acionado com o propósito de estabelecer uma rota que minimize danos e aumente a segurança durante o pouso emergencial.

Quatro estratégias para o planejador de rotas foram apresentadas: uma Heurística Gulosa (HG), duas metaheurísticas evolutivas, Algoritmo Genético (AG) e Algoritmo Genético Multi-Populacional (AGMP), e um abordagem exata a partir de um modelo de Programação Linear Inteira Mista (PLIM). Duas versões para cada método evolutivo foram criadas que se diferenciam por ativar ou não um operador de inicialização dos indivíduos através da HG. HG, AG e AGMP utilizam uma modelagem restrita do problema ao efetuarem o pouso. PLIM utiliza a modelagem completa. Duas versão do modelo PLIM utilizando diferentes funções objetivo foram desenvolvidas. Um conjunto de 600 mapas foram gerados com o objetivo de definir alguns níveis de dificuldade e avaliar os métodos propostos diante de diferentes cenários. A modelagem da dinâmica do VANT diante de algumas situações críticas também foi efetuada.

A estratégia de supervisão dos estados da aeronave, chamada *In-Flight Awareness*, foi aplicada neste trabalho. O IFA fornece ao algoritmo planejador informações importantes sobre as condições da situação crítica encontrada, o conjunto de estados da aeronave no momento da falha e o cenário do voo. Essas informações são utilizadas pelo método que planeja a trajetória para encontrar uma rota de pouso emergencial.

Os resultados obtidos com a HG mostram que no pior caso de situação crítica (problema no motor), ainda assim, consegue-se efetuar o pouso em região bonificadora em cerca de 67% dos mapas. No melhor caso, problema na bateria, consegue-se efetuar o pouso em 89% dos mapas. O tempo médio para execução da HG é de 0,07 segundo. Esses resultados em termos de tempo computacional revelam a viabilidade de se tentar embarcar a HG em um VANT real. A facilidade de implementação da HG também auxiliaria nesse processo. Por isso, alguns experimentos utilizando a ferramenta SCADE para o embarque da HG foram iniciados.

Os métodos AG e AGMP obtiveram melhores resultados do que a HG em termos de qualidade de solução. O AG1 e AGMP1 apresentaram resultados similares pousando o VANT em região bonificadora em aproximadamente 78% e 80% dos mapas, respectivamente. Os métodos AG2 e AGMP2 efetuaram ambos o pouso sobre essas regiões em 89% dos mapas. Logo, as abordagens que utilizam a inicialização gulosa (AG2 e AGMP2) são melhores que as que não o utilizam (AG1 e AGMP1). Em termos de tempo computacional, tanto o AG e AGMP gastaram na ordem de 1 segundo para encontrar a solução. Devido a grande semelhança, em termos de resultados, pode-se considerar equivalentes as duas abordagens evolutivas. No entanto, ambos os métodos evolutivos, ao utilizar a inicialização gulosa, possuem melhores qualidades da solução que a abordagem HG. Considerando o tempo computacional, a HG foi cerca de 14 vezes mais rápida do que as metaheurísticas evolutivas. Apesar do tempo de resposta do AG e AGMP serem maiores, ainda assim, essa estratégia poderia ser embarcada em um VANT.

Os resultados para a modelagem completa foram gerados através de duas estratégias. A primeira estratégia (PLIM1) utiliza uma função objetivo que não penaliza a quantidade de *waypoints* alocados. Os resultados para o PLIM1 gastaram, em geral, cerca de 1043 segundos (~ 17 minutos) e conseguiram pousar o VANT em região bonificadora em 60% dos mapas. A estratégia PLIM2 leva cerca de 255 segundos (~ 4 minutos) para gerar uma solução com qualidade similar a PLIM1. Sendo assim, a estratégia PLIM2 se mostrou bem mais promissora. Apesar dessa melhoria, esse tempo de processamento é ainda muito alto ao se tratar uma aplicação crítica. Assim, a estratégia PLIM não poderia ser embarcada em um VANT real. Afim de contornar essa situação, um estudo envolvendo um conjunto de rotas pré-calculadas para pousar o VANT foi realizado. Dessa forma, apenas as rotas emergenciais previamente calculadas seriam embarcadas no VANT, o que tornaria a estratégia exata adequada a aplicação crítica tratada.

Alguns resultados envolvendo simulador de voo foram efetuados tanto para o AGMP e PLIM. Esses resultados mostraram que as trajetórias de pouso geradas podem ser seguidas por uma aeronave de grande porte como o Cessna 172. Nesses experimentos, diferentes condições de vento foram testadas e, mesmo assim, os controladores da aeronave conseguiram traçar a rota dentro da margem de segurança especificada.

Por fim, um pequeno experimento de voo real foi efetuado em um hexacóptero. O voo mostrou a viabilidade de executar as trajetórias definidas no trabalho nesse tipo de aeronave. Isso

mostra que, apesar das rotas planejadas nesta dissertação serem inspiradas em uma VANT de asa fixa, uma aeronave de asa rotativa poderia ser utilizada.

6.2 Artigos

Durante a elaboração desta dissertação os seguintes artigos foram escritos:

A Multi-Population Genetic Algorithm for UAV Path Re-Planning under Critical Situation:

Publicado no *ICTAI - International Conference on Tools with Artificial Intelligence* em 2015. Qualis A2.

A Hybrid Multi-Population Genetic Algorithm for UAV Path Planning:

Publicado no *GECCO - Genetic and Evolutionary Computation Conference* em 2016. Qualis A1.

A Mathematical Formulation for UAV Path Re-Planning under Critical Situation:

A ser submetido em periódico internacional em 2016.

6.3 Trabalhos Futuros

Como trabalhos futuros pretende-se avançar em experimentos usando *Hardware-In-the-Loop* (HIL). E também obter avanços em experimentos evolvendo aeronaves reais, afim de validar melhor os algoritmos propostos. Nesse sentido, almeja-se embarcar os algoritmos de pouso propostos, o sistema supervisor IFA e o sistema de monitoramento da missão MOSA no projeto de doutorado do autor.

Um outro trabalho a ser realizado é o desenvolvimento da HG, AG e AGMP para o modelo completo utilizando a mesma equação de dinâmica. Dessa forma, uma comparação mais direta desses métodos poderiam ser efetuadas. Trabalhos envolvendo mapas mais complexos contendo relevos, uma diversidade maior de regiões e com formatos diversos poderiam ser também efetuados.

REFERÊNCIAS

- AGX. **AGX Tecnologia.** <http://www.agx.com.br/n2/pages/index.php?opt=airplanehistory>: [s.n.], 2014. Citado na página 38.
- ANAC. **Instrução Suplementar - IS: IS n 21-002 Revisão A.** [S.l.], 2012. 21 p. Citado na página 37.
- _____. **Proposta de Instrução Suplementar, Intitulada Emissão de Certificado de Autorização de Voo Experimental para Sistemas de Veículo Aéreo Não Tripulado.** [S.l.], 2012. 5 p. Citado na página 37.
- ARANTES, J. d. S.; ARANTES, M. d. S.; TOLEDO, C. F. M.; WILLIAMS, B. C. A multi-population genetic algorithm for uav path re-planning under critical situation. In: **Tools with Artificial Intelligence (ICTAI), 2015 IEEE 27th International Conference on.** [S.l.: s.n.], 2015. p. 486–493. ISSN 1082-3409. Citado 2 vezes nas páginas 113 e 117.
- ARANTES, M. d. S.; TOLEDO, C. F. M.; WILLIAMS, B. C. Improving encoding and solution approaches for the chance-constrained non convex path planning problem. In: **Não Publicado.** [S.l.: s.n.], 2016. Citado na página 67.
- ARANTES, M. N. **Hybrid Qualitative State Plan Problem e o Planejamento de Missão com VANTs.** Tese (Qualificação de Doutorado) — Universidade de São Paulo (USP), set 2015. São Carlos, SP. Citado na página 32.
- ARANTES, M. S. **Ambiente para desenvolvimento de métodos aplicados a problemas de otimização.** Dissertação (Mestrado) — Universidade de São Paulo (USP/ICMC), 2014. Citado na página 102.
- ASTUTI, G.; CALTABIANO, D.; GIUDICE, G.; LONGO, D.; MELITA, D.; MUSCATO, G.; ORLANDO, A. “hardware in the loop” tuning for a volcanic gas sampling uav. In: **Valavanis K. P. (Ed). Advances in Unmanned Aerial Vehicles.** Springer, v. 33, p. 473–493, 2007. Citado na página 44.
- BANKS, J. Discrete event simulation. In: **Winter Simulation Conference.** [S.l.: s.n.], 2000. Citado na página 43.
- BESADA-PORTAS, E.; TORRE, L. D. L.; MORENO, A.; RISCO-MARTÍN, J. L. On the performance comparison of multi-objective evolutionary uav path planners. **Inf. Sci.**, Elsevier Science Inc., New York, NY, USA, v. 238, p. 111–125, jul 2013. ISSN 0020-0255. Disponível em: <<http://dx.doi.org/10.1016/j.ins.2013.02.022>>. Citado 2 vezes nas páginas 32 e 48.
- BLACKMORE, L.; ONO, M.; WILLIAMS, B. C. **Chance-Constrained Optimal Path Planning with Obstacles.** 2011. IEEE Press. Citado 10 vezes nas páginas 32, 34, 46, 49, 55, 57, 59, 76, 91 e 136.
- BRAGA, R. T. V.; BRANCO, K. R. L. J. C.; JÚNIOR, O. T.; GIMENES, I. M. S. Evolving tiriba design towards a product line of small electric-powered uavs. In: **Brazilian Conference on Critical Embedded Systems.** [S.l.: s.n.], 2011. p. p. 67–72. Citado na página 39.

BRANCO, K.; PELIZZONI, J.; NERIS, L. O.; TRINDADE, O.; OSÓRIO, F.; WOLF, D. Tiriba - a new approach of uav based on model driven development and multiprocessors. In: **ICRA**. [S.I.]: IEEE, 2011. p. 1–4. ISSN 1050-4729. Citado na página 39.

BURIOL, L.; FRANCA, P. M.; MOSCATO, P. A new memetic algorithm for the asymmetric traveling salesman problem. **Journal of Heuristics**, Kluwer Academic Publishers, Hingham, MA, USA, v. 10, n. 5, p. 483–506, set 2004. ISSN 1381-1231. Disponível em: <<http://dx.doi.org/10.1023/B:HEUR.0000045321.59202.52>>. Citado na página 49.

CENIPA. **FCA 58-1: Panorama Estatístico da Aviação Civil Brasileira**. Brasília, DF, Brasil, 2012. 81 p. Citado na página 34.

DRONEWARS. **Drone Crash Database - Drone Wars UK**. <http://dronewars.net/drone-crash-database/>: [s.n.], 2016. Citado 2 vezes nas páginas 40 e 41.

DRURY, J. L.; RIEK, L.; RACKLIFFE, N. A decomposition of uav-related situation awareness. In: **Proceedings of the 1st ACM SIGCHI/SIGART Conference on Human-robot Interaction**. New York, NY, USA: ACM, 2006. (HRI '06), p. 88–94. ISBN 1-59593-294-1. Disponível em: <<http://doi.acm.org/10.1145/1121241.1121258>>. Citado na página 42.

EASA. **Acceptable Means of Compliance (AMC) and Guidance Material (GM) to Part SPA**. European, 2012. 109 p. Citado na página 96.

EIBEN, A. E.; SMITH, J. E. **Introduction to Evolutionary Computing**. [S.I.]: SpringerVerlag, 2003. ISBN 3540401849. Citado 2 vezes nas páginas 80 e 82.

ENDSLEY, M. R. Design and evaluation for situation awareness enhancement. **Human Factors and Ergonomics Society Annual Meeting Proceedings**, Human Factors and Ergonomics Society, p. 97–101, 1988. ISSN 1071-1813. Disponível em: <<http://www.ingentaconnect.com/content/hfes/hfproc/1988/00000032/00000002/art00021>>. Citado na página 42.

ENDSLEY, M. R.; JONES, G. D. **Designing for Situation Awareness: An Approach to User-Centered Design**. 2. ed. [S.I.]: CRC Press, 2006. 396 p. Citado na página 42.

ESHELMAN, L. J.; SCHAFER, J. D. Real-coded genetic algorithms and interval-schemata. In: WHITLEY, L. D. (Ed.). **FOGA**. Morgan Kaufmann, 1992. p. 187–202. ISBN 1-55860-263-1. Disponível em: <<http://dblp.uni-trier.de/db/conf/foga/foga1992.html#EshelmanS92>>. Citado na página 82.

ESTEREL. **SCADE Suite**. <http://www.esterel-technologies.com/products/scade-suite/>: [s.n.], 2015. Citado na página 139.

FAA. **FAA Modernization and Reform Act of 2012**. Washington, DC, USA, 2012. 296 p. Citado na página 31.

FIGUEIRA, M. N. **Arranjos de sensores orientados a missão para a geração automática de mapas temáticos em VANTs**. Tese (Tese de Doutorado) — Universidade de São Paulo (USP), nov 2015. São Carlos, SP. Citado na página 32.

FIGUEIRA, N.; TRINDADE, O.; MATTEI, A. L. P.; NERIS, L. Mission oriented sensor arrays – an approach towards uas usability improvement in practical applications. In: **5th European Conference for Aeronautics and Space Sciences (EUCASS)**. [S.I.: s.n.], 2013. Citado 2 vezes nas páginas 32 e 42.

- FLIGHTGEAR. **FlightGear Flight Simulator**. <http://www.flightgear.org/>: [s.n.], 2014. Citado na página 44.
- _____. **FlightGear Wiki**. <http://wiki.flightgear.org/>: [s.n.], 2014. Citado na página 43.
- _____. **OpenGL - FlightGear Wiki**. <http://wiki.flightgear.org/OpenGL>: [s.n.], 2014. Citado na página 44.
- _____. **SimGear - FlightGear Wiki**. <http://wiki.flightgear.org/SimGear>: [s.n.], 2014. Citado na página 43.
- _____. **TerraGear - FlightGear Wiki**. <http://wiki.flightgear.org/TerraGear>: [s.n.], 2014. Citado na página 43.
- FONSECA, E.; MATTEI, A. L. P.; CUNHA, W. C. Adaptative integration systems using fpga cots devices. In: **5th European Conference for Aeronautics and Space Sciences (EUCASS)**. Munich: [s.n.], 2013. Citado na página 32.
- FRANCA, P. M.; MENDES, A.; MOSCATO, P. A memetic algorithm for the total tardiness single machine scheduling problem. **European Journal of Operational Research**, v. 132, n. 1, p. 224 – 242, 2001. ISSN 0377-2217. Disponível em: <<http://www.sciencedirect.com/science/article/pii/S0377221700001405>>. Citado na página 48.
- GISA. **Grupo de Interesse em Sistemas Autônomos e Aplicações**. www.gisa.icmc.usp.br: [s.n.], 2015. Citado na página 38.
- HERMANSSON, J.; GISING, A.; SKOGLUND, M.; SCHON, T. B. Autonomous landing of an unmanned aerial vehicle. In: **Reglermote (Swedish Control Conference), Sweden**. [S.l.: s.n.], 2010. Citado 2 vezes nas páginas 47 e 89.
- HERRERA, F.; LOZANO, M.; SÁNCHEZ, A. M. Hybrid crossover operators for real-coded genetic algorithms: an experimental study. **Soft Comput.**, v. 9, n. 4, p. 280–298, 2005. Disponível em: <<http://dblp.uni-trier.de/db/journals/soco/soco9.html#HerreraLS05>>. Citado na página 82.
- HOLLAND, J. H. **Adaptation in Natural and Artificial Systems**. Ann Arbor, MI, USA: University of Michigan Press, 1975. Citado na página 80.
- HOSSOMI, M. Y. B. **Framework para Desenvolvimento de Ambientes de Simulação e Controle**. Dissertação (Monografia de Graduação) — Universidade de São Paulo (USP), nov 2015. São Carlos, SP. Citado na página 98.
- ICAO, I. C. A. O. **Doc 9859: Safety Management Manual (SMM)**. Edition 3. [S.l.], 2013. 251 p. Citado na página 31.
- JIN, Y. Surrogate-assisted evolutionary computation: Recent advances and future challenges. **Swarm and Evolutionary Computation**, v. 1, n. 2, p. 61–70, 2011. Disponível em: <<http://dblp.uni-trier.de/db/journals/swevo/swevo1.html#Jin11>>. Citado na página 80.
- JONG, K. A. D. **Evolutionary computation - a unified approach**. [S.l.]: MIT Press, 2006. I-IX, 1-256 p. ISBN 978-0-262-04194-2. Citado na página 80.
- KIM, H.; KIM, M.; LIM, H.; PARK, C.; YOON, S.; LEE, D.; CHOI, H.; OH, G.; PARK, J.; KIM, Y. Fully autonomous vision-based net-recovery landing system for a fixed-wing uav. **Mechatronics, IEEE/ASME Transactions on**, v. 18, n. 4, p. 1320–1333, Aug 2013. ISSN 1083-4435. Citado 2 vezes nas páginas 47 e 89.

LEITE, O. **Projeto de Lei Nº 5942.** [S.l.], 2013. 11 p. Citado na página 37.

LI, H. X. **Kongming: A Generative Planner for Hybrid Systems with Temporally Extended Goals.** Tese (237 p.) — Massachusetts Institute of Technology, 2010. Citado 4 vezes nas páginas 32, 34, 46 e 49.

LI, P.; CHEN, X.; LI, C. Emergency landing control technology for uav. In: **Guidance, Navigation and Control Conference (CGNCC), 2014 IEEE Chinese.** [S.l.: s.n.], 2014. p. 2359–2362. Citado 2 vezes nas páginas 35 e 47.

LI, X. A software scheme for uav's safe landing area discovery. **{AASRI} Procedia**, v. 4, p. 230 – 235, 2013. ISSN 2212-6716. 2013 {AASRI} Conference on Intelligent Systems and Control. Disponível em: <<http://www.sciencedirect.com/science/article/pii/S221267161300036X>>. Citado 3 vezes nas páginas 35, 47 e 89.

MASUKO, K.; TAKAHASHI, I.; OGAWA, S.; WU, M.-H.; OOSEDŌ, A.; MATSUMOTO, T.; GO, K.; SUGAI, F.; KONNO, A.; UCHIYAMA, M. Autonomous takeoff and landing of an unmanned aerial vehicle. In: **System Integration (SII) 2010 IEEE SICE International Symposium on.** [S.l.: s.n.], 2010. p. 248–253. Citado na página 89.

MATTEI, A. L. P. **Consciência situacional em voo de sistemas aéreos não tripulados.** Tese (Tese de Doutorado) — Universidade de São Paulo (USP), ago 2015. São Carlos, SP. Citado 3 vezes nas páginas 31, 32 e 41.

MATTEI, A. L. P.; CUNHA, A. M.; DIAS, L. A. V.; EUPHRASIO, P. C. S.; TRINDADE, O.; TOLEDO, C. M. Ifa2s – in-flight awareness augmentation systems. In: **Information Technology - New Generations (ITNG), 2015 12th International Conference on.** [S.l.: s.n.], 2015. p. 95–100. Citado na página 41.

MATTEI, A. L. P.; FONSECA, E.; FIGUEIRA, N. M.; TRINDADE, O.; VAZ, F. Uav in-flight awareness: A tool to improve safety. In: **5th European Conference for Aeronautics and Space Sciences (EUCASS).** Munich: [s.n.], 2013. Citado 2 vezes nas páginas 32 e 41.

MEULEAU, N.; NEUKOM, C.; PLAUNT, C.; SMITH, D. E.; SMITH, T. The emergency landing planner experiment. In: **21st International Conference on Automated Planning and Scheduling.** [S.l.: s.n.], 2011. Citado 2 vezes nas páginas 35 e 47.

MEULEAU, N.; PLAUNT, C.; SMITH, D. E.; SMITH, T. B. An emergency landing planner for damaged aircraft. In: HAIGH, K. Z.; RYCHTYCKYJ, N. (Ed.). **IAAI.** AAAI, 2009. Disponível em: <<http://dblp.uni-trier.de/db/conf/iaai/iaai2009.html#MeuleauPSS09>>. Citado 4 vezes nas páginas 35, 46, 47 e 49.

MICHALEWICZ, Z.; SCHOENAUER, M. Evolutionary algorithms for constrained parameter optimization problems. **Evolutionary Computation**, v. 4, p. 1–32, 1996. Citado na página 82.

MICROSOFT. **Microsoft Flight.** <http://www.microsoft.com/games/flight/>: [s.n.], 2014. Citado 2 vezes nas páginas 44 e 45.

MORTIMER, G. **sUAS News: Schiebel S-100 crash kills engineer in South Korea.** <http://www.suasnews.com/2012/05/15515/schiebel-s-100-crash-kills-engineer-in-south-korea>: [s.n.], 2012. Citado na página 33.

- MOSCATO, P.; MENDES, A.; BERRETTA, R. Benchmarking a memetic algorithm for ordering microarray data. **Biosystems**, v. 88, n. 1-2, p. 56–75, 2007. Disponível em: <<http://dblp.uni-trier.de/db/journals/biosystems/biosystems88.html#MoscatoMB07>>. Citado na página 49.
- NATO, N. A. T. O. **STANAG 4671: UAV Systems Airworthiness Requirements (USAR) for North Atlantic Treaty Organization (NATO) Military UAV Systems**. 1. ed. [S.l.], 2007. 205 p. Citado na página 31.
- NIKOLOS, I. K.; TSOURVELOUDIS, N. C.; VALAVANIS, K. P. Evolutionary algorithm based path planning for multiple uav cooperation. **Springer**, v. 33, p. 310–340, 2007. Citado na página 48.
- OGLOBO. **Uma queda drone e bater um corredor, seu operador evoca pirataria**. <http://oglobo.globo.com/mundo/eua-registraram-mais-de-400-acidentes-com-drones-militares-12950003>: [s.n.], 2014. Citado na página 33.
- ONO, M.; WILLIAMS, B. C.; BLACKMORE, L. Probabilistic planning for continuous dynamic systems under bounded risk. **J. Artif. Int. Res.**, AI Access Foundation, USA, v. 46, n. 1, p. 511–577, jan 2013. ISSN 1076-9757. Disponível em: <<http://dl.acm.org/citation.cfm?id=2512538.2512551>>. Citado 8 vezes nas páginas 32, 34, 46, 49, 59, 60, 67 e 76.
- PEHLIVANOGLU, Y. V. A new vibrational genetic algorithm enhanced with a voronoi diagram for path planning of autonomous uav. **Aerospace Science and Technology**, v. 16, n. 1, p. 47 – 55, 2012. ISSN 1270-9638. Disponível em: <<http://www.sciencedirect.com/science/article/pii/S1270963811000356>>. Citado 2 vezes nas páginas 32 e 47.
- RICHARDS, A.; HOW, J. P. Aircraft trajectory planning with collision avoidance using mixed integer linear programming. In: **American Control Conference (ACC)**. [S.l.: s.n.], 2002. v. 3, p. 1936–1941 vol.3. Citado na página 32.
- RODRIGUES, D.; PIRES, R. de M.; ESTRELLA, J.; MARCONATO, E.; TRINDADE, O.; BRANCO, K. Using soa in critical-embedded systems. In: **Internet of Things (iThings/CPS-Com), 2011 International Conference on and 4th International Conference on Cyber, Physical and Social Computing**. [S.l.: s.n.], 2011. p. 733–738. Citado na página 31.
- RODRIGUES, L. E. M. J. **Fundamentos da Engenharia Aeronáutica Aplicações ao Projeto SAE - Aerodesign**. [S.l.]: 1 ed, 2011. v. 2. Citado 2 vezes nas páginas 39 e 40.
- SARIPALLI, S.; MONTGOMERY, J.; SUKHATME, G. Vision-based autonomous landing of an unmanned aerial vehicle. In: **Robotics and Automation, 2002. Proceedings. ICRA '02. IEEE International Conference on**. [S.l.: s.n.], 2002. v. 3, p. 2799–2804. Citado 2 vezes nas páginas 47 e 89.
- SCADE. **Technical Data Sheet - SCADE System R16**. [S.l.], 2015. 5 p. Citado na página 139.
- SEPPÄNEN, H.; MÄKELÄ, J.; LUOKKALA, P.; VIRRANTAUS, K. Developing shared situational awareness for emergency management. In: **Safety Science**. [S.l.: s.n.], 2013. v. 55, p. 1–9. Citado na página 42.
- SMARTDROONES. **Suíça: um operador de robô foi morto por sua própria unidade**. <http://www.smartdrones.fr/pt/Operador-su%C3%ADo-de-um-drone-tinha-sido-morto-por-sua-pr%C3%B3pria-unidade/00921>: [s.n.], 2014. Citado na página 33.

_____. **Um jovem é morto por seu próprio robô.** <http://www.smartdrones.fr/pt/morto-por-seu-pr%C3%B3prio-rob%C3%A9-em-um-parque-no-Brooklyn/001266>: [s.n.], 2014. Citado na página 33.

_____. **Uma queda drone e bater um corredor, seu operador evoca pirataria.** <http://www.smartdrones.fr/pt/zang%C3%A3o-%C3%A9-acidente-pode-sempre-dizer-que-ele-era-Hacke/002771>: [s.n.], 2014. Citado na página 33.

STUDART, N.; DAHMEN, S. R. A física do voo na sala de aula. In: **Física na Escola**. [S.l.: s.n.], 2006. v. 7, n. 2, p. 1–7. Citado 2 vezes nas páginas 58 e 101.

TAVAKKOLI-MOGHADDAM, R.; GHOLIPOUR-KANANI, Y.; CHERAGHALIZADEH, R. A genetic algorithm and memetic algorithm to sequencing and scheduling of cellular manufacturing systems. **International Journal of Management Science and Engineering Management**, p. 119–130, 2008. Citado na página 49.

TOLEDO, C.; FRANCA, P.; KIMMS, A.; MORABITO, R. A multi-population genetic algorithm approach to solve the synchronized and integrated two-level lot sizing and scheduling problem. **International Journal of Production Research**, v. 47, p. 3097–3119, 2009. ISSN 0020-7543. Citado na página 83.

TOLEDO, C. F. M.; ARANTES, M.; OLIVEIRA, R. R. R. de; ALMADA-LOBO, B. Glass container production scheduling through hybrid multi-population based evolutionary algorithm. **Appl. Soft Comput.**, v. 13, n. 3, p. 1352–1364, 2013. Disponível em: <<http://dblp.uni-trier.de/db/journals/asc/asc13.html#ToledoAOA13>>. Citado 2 vezes nas páginas 48 e 49.

TOLEDO, C. F. M.; LIMA, J. M. G.; ARANTES, M. da S. A multi-population genetic algorithm approach for pid controller auto-tuning. In: **ETFA**. [S.l.: s.n.], 2012. p. 1–8. Citado na página 49.

TUNCER, A.; YILDIRIM, M. Dynamic path planning of mobile robots with improved genetic algorithm. **Comput. Electr. Eng.**, Pergamon Press, Inc., Tarrytown, NY, USA, v. 38, n. 6, p. 1564–1572, nov 2012. ISSN 0045-7906. Disponível em: <<http://dx.doi.org/10.1016/j.compeleceng.2012.06.016>>. Citado 2 vezes nas páginas 32 e 47.

VARELA, G.; NO, P. C.; ORJALES, F.; DEIBE, A.; NA, F. L.-P.; DURO, R. J. Autonomous uav based search operations using constrained sampling evolutionary algorithms. **Neurocomputing**, v. 132, n. 0, p. 54 – 67, 2014. ISSN 0925-2312. Disponível em: <<http://www.sciencedirect.com/science/article/pii/S0925231213010904>>. Citado 2 vezes nas páginas 32 e 48.

WIKIPÉDIA. **FlightGear** - Wikipédia, a encyclopédia livre. <http://pt.wikipedia.org/wiki/FlightGear>: [s.n.], 2014. Citado na página 43.

_____. **Microsoft Flight** - Wikipédia, a encyclopédia livre. http://en.wikipedia.org/wiki/Microsoft_Flight: [s.n.], 2014. Citado na página 45.

_____. **Veículo aéreo não tripulado** - Wikipédia, a encyclopédia livre. https://pt.wikipedia.org/wiki/Ve%C3%A7ulo_a%C3%A9reo_n%C3%A3o_tripulado: [s.n.], 2014. Citado na página 38.

X-PLANE. **Plane Maker App Manual**. [S.l.], 2013. 105 p. Citado na página 44.

_____. **FAA-Certified X-Plane**. <http://www.x-plane.com/pro/certified/>: [s.n.], 2014. Citado na página 44.

- _____. **World Editor Manual - X-Plane.** <http://developer.x-plane.com/manuals/wed/>: [s.n.], 2014. Citado na página 44.
- _____. **X-Plane 10 Global.** <http://www.x-plane.com/desktop/home/>: [s.n.], 2014. Citado na página 44.
- ZHANG, X.; DUAN, H. An improved constrained differential evolution algorithm for unmanned aerial vehicle global route planning. In: . [S.l.: s.n.], 2015. p. 270–284. Citado 2 vezes nas páginas 32 e 48.
- ZHAO, Y.; PEI, H. An improved vision-based algorithm for unmanned aerial vehicles autonomous landing. **Physics Procedia**, v. 33, p. 935 – 941, 2012. ISSN 1875-3892. 2012 International Conference on Medical Physics and Biomedical Engineering (ICMPBE2012). Disponível em: <<http://www.sciencedirect.com/science/article/pii/S1875389212014708>>. Citado 2 vezes nas páginas 47 e 89.



GLOSSÁRIO

Aeronave de Asa Fixa: aeronave capaz de voar usando a asa fixada em seu chassi. Alguns exemplos dessas aeronaves são: Tiriba, Ararinha, Arara M1, BQM-1BR e aviões em geral.

Aeronave de Asa Fixa: aeronave capaz de voar usando a asa fixada em seu chassi. Alguns exemplos dessas aeronaves são: Tiriba, Ararinha, Arara M1, BQM-1BR e aviões em geral.

Aeronave de Asa Rotativa: aeronave em que sua asa é representada por hélices que giram em torno de um eixo vertical. Alguns exemplos dessas aeronaves são: helicópteros, autogiros, quadricópteros, hexacópteros e multicópteros.

Conventional Take-off and Landing (CTOL): aeronave que possui sistema de decolagem e aterrissagem convencional efetuado sobre pistas. Alguns exemplos desse tipo de voo são os efetuados por aviões tradicionais.

Vertical Take-off and Landing (VTOL): veículo aéreo que possui sistema de decolagem e aterrissagem na vertical. Alguns exemplos de tal tipo de pouso são helicópteros, quadricópteros, autogiros e balões a gás.

Roll: representa a rotação do VANT ao redor do eixo longitudinal da aeronave.

Pitch: representa a rotação do VANT ao redor do eixo lateral da aeronave.

Yaw: representa a rotação do VANT ao redor do eixo vertical da aeronave.

Aileron: componente da aeronave, sendo o principal responsável pela rotação *roll*.

Profundor: componente da aeronave, sendo o principal responsável pela rotação *pitch*.

Leme: componente da aeronave, sendo o principal responsável pela rotação *yaw*.

População: Conjunto de indivíduos que representam várias soluções possíveis para o problema.

Indivíduo: Um simples membro da população. Um indivíduo representa uma solução para o problema proposto.

Fitness: Tem por finalidade avaliar as soluções encontradas pelos indivíduos, associando a cada um deles um valor de acordo com sua qualidade.

Seleção: Seleciona indivíduos durante o processo de cruzamento ou para definir quais integram a população na próxima geração. Os operadores de seleção se baseiam no valor do *fitness* dos indivíduos como critério para sua escolha.

Crossover: Dois indivíduos são selecionados para cruzamento, ou seja, realiza-se uma troca entre as informações codificadas nestes indivíduos gerando novos indivíduos.

Mutação: É um operador que altera de forma randômica parte da informação codificada no indivíduo. Isso pode levar a maior variabilidade na população, impedindo que a exploração do espaço de busca das soluções fique estagnada em um mínimo local.

Critério de Parada: Critério estabelecido para finalização da execução do algoritmo. Alguns exemplos desses critérios são: número de avaliações, número de gerações, convergência da população e tempo.

Gerações: Número de vezes que uma nova população será gerada.

Elitismo: A ideia do elitismo é manter as melhores soluções na geração seguinte, assim, uma cópia dos melhores indivíduos é feita para compor a próxima geração. Estratégias elitistas podem garantir que não ocorra perda de informações relevantes na representação da melhor solução.

Simulação Online: simulações que ocorrem quando o algoritmo planejador de rotas está sendo executado junto a um simulador de voo.

Simulação Offline: simulações em que os algoritmos responsáveis pelo pouso da aeronave não estão sendo executados juntamente com o simulador de voo.