
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: 23 de fevereiro de 2015

Assinatura: _____

Planejamento de Rota para VANTs em Caso de Situação Crítica: Uma Abordagem Baseada em Segurança¹

Jesimar da Silva Arantes

Orientador: *Prof. Dr. Claudio Fabiano Motta Toledo*

Monografia apresenta ao Instituto de Ciências Matemáticas e de Computação – ICMC/USP, para o exame de Qualificação, como parte dos requisitos para obtenção do título de Mestre em Ciência de Computação e Matemática Computacional.

USP - São Carlos
Fevereiro/2015

¹Trabalho Realizado com Auxílio da FAPESP

Resumo

O presente trabalho apresenta uma proposta de dissertação a ser desenvolvida no Programa de Mestrado em Ciência da Computação e Matemática Computacional do ICMC/USP. A segurança envolvida nos voos de Veículos Aéreos não Tripulados (VANTs) é uma importante questão e vem ganhando destaque, devido a uma série de acidentes envolvendo tais aeronaves [**Contexto**]. Nesse contexto o desenvolvimento de algoritmos que efetuem o planejamento de rotas para voo de VANTs na ocorrência de situações críticas é fundamental para obter maior segurança aérea. Eventuais situações de insegurança podem estar relacionadas a uma falha crítica nos equipamentos do VANT que impede a continuação da missão em curso pela aeronave [**Lacuna**]. A pesquisa busca avançar o estado da arte considerando um conceito, chamado *In-Flight Awareness* (IFA), que estabelece consciência situacional em VANTs visando maior segurança de voo. Pretende-se avançar também na elaboração de modelos matemáticos que representem o estado da aeronave avaliada. O propósito principal é viabilizar o pouso emergencial de VANTs em caso de situação crítica, minimizando possíveis danos [**Propósito**]. Este trabalho assim adaptará técnicas de computação evolutiva no tratamento de falhas críticas junto com o conceito de IFA [**Metodologia**]. As soluções serão obtidas através de experimentos realizados *offline* usando os modelos matemáticos desenvolvidos e por fim serão validadas em simulador de voo [**Resultado**]. O trabalho apresentado pretende colaborar com a segurança de veículos aéreos não tripulados e com a proposta de modelos matemáticos que representem a aeronave em caso de situações críticas [**Conclusão**].

Palavras-chave: Computação Evolutiva, Otimização, Veículo Aéreo Não Tripulado, Planejamento de Rota e Segurança de Voo.

Abstract

THis present work presents a dissertation proposal being developed in the Master's Program in Computer Science and Computational Mathematics at ICMC/USP. The security involved in flights of Unmanned Aerial Vehicles (UAVs) is an important issue and has been gaining attention due to a series of accidents involving such aircraft [**Context**]. In this context the development of algorithms that contribute significantly to the route planning for UAS flight in the event of critical situations is essential for greater air safety. Possible situations of insecurity may be related to a critical failure in the equipment of the UAV that prevents the continuation of the ongoing mission the aircraft [**Gap**]. The research seeks to advance the state of the art considering a concept called In-Flight Awareness (IFA), which provides situational awareness in UAVs aiming at greater flight safety. We also intend to advance the mathematical models that represent the state of the aircraft evaluated. The main purpose is to make the emergency landing of UAVs in case of emergency, minimizing possible damage [**Purpose**]. This study thus adapt evolutionary computation techniques in the treatment of critical failures along with the concept of IFA [**Methodology**]. The solutions will be obtained through experiments offline using the developed mathematical models and finally will be validated in a Flight Simulator [**Result**]. The presented work intends to collaborate with the security unmanned aerial vehicles and with the proposed mathematical models that represent the aircraft in case of critical situations [**Conclusion**].

Keywords: Evolutionary Computation, Optimization, Unmanned Aerial Vehicle, Route Planning and Flight Safety

Sumário

Resumo	v
Abstract	vii
Sumário	x
Lista de Figuras	xi
Lista de Tabelas	xiii
Lista de Algoritmos	xv
Lista de Abreviaturas	xviii
Lista de Símbolos	1
1 Introdução	3
1.1 Contextualização	3
1.2 Motivação e Justificativas	4
1.3 Objetivos	5
1.4 Contribuições e Limitações	6
1.5 Organização	7
1.6 Conclusão	7
2 Revisão Bibliográfica	9
2.1 Introdução	9
2.2 Conceitos Importantes	9
2.2.1 Veículos Aéreos Não Tripulados	10
2.2.2 Algoritmos Genéticos	13
2.2.3 In-Flight Awareness	15
2.3 Simuladores de Voo	17
2.3.1 FlightGear	17
2.3.2 X-Plane	18
2.3.3 Microsoft Flight Simulator	19
2.3.4 Comparação dos Simuladores	19
2.4 Trabalhos Relacionados	20
2.5 Conclusão	23

3	Problema Abordado	25
3.1	Introdução	25
3.2	Descrição do Problema	25
3.2.1	Conjunto de Regiões Modeladas	27
3.2.2	Situações Críticas Modeladas	28
3.2.3	Hipóteses sobre Condições de Voo e Ambiente	29
3.3	Modelagem do Problema	29
3.4	Conclusão	32
4	Metodologia	33
4.1	Introdução	33
4.2	Codificação do Problema	34
4.2.1	Função Fitness	35
4.2.2	Operadores Inicialização	36
4.2.3	Operadores Crossover	37
4.2.4	Operadores Mutação	38
4.3	Árvore de Decisão	38
4.4	Heurística Gulosa	39
4.5	Algoritmo Genético	41
4.6	Algoritmo Genético Multi-Populacional	42
4.7	Método Exato	43
4.8	Arquitetura Proposta	44
4.9	Conclusão	45
5	Resultados Preliminares	47
5.1	Introdução	47
5.2	Geração Automática de Mapas	47
5.2.1	Classificação dos Mapas	48
5.2.2	Algoritmo de Geração de Mapas	49
5.2.3	Configurações Usadas na Geração dos Mapas	50
5.3	Modelagem da Dinâmica do VANT	51
5.3.1	Dedução da Constante de Arrasto	54
5.4	Experimentos	56
5.4.1	Configurações Utilizadas	56
5.4.2	Análise de Sensibilidade do AG e AGMP	57
5.4.3	Exemplo de Soluções de Situações Críticas	59
5.4.4	Heurística Gulosa	60
5.4.5	Algoritmo Genético	62
5.4.6	Algoritmo Genético Multi-Populacional	63
5.5	Conclusão	64
6	Cronograma	65
7	Considerações Finais	67
	Referências Bibliográficas	68

Lista de Figuras

2.1	Veículo aéreo não tripulado BQM-1BR. Foto: Rodrigo Zanette.	10
2.2	Veículo aéreo não tripulado Arara M1. Foto: (AGX, 2014).	11
2.3	Veículo aéreo não tripulado Tiriba. Foto: (AGX, 2014).	11
2.4	Esquema da integração entre IFA, piloto automático e MOSA na aeronave. . . .	17
2.5	Simulador de voo FlightGear. Fonte: (FlightGear, 2014b).	18
2.6	Simulador de voo X-Plane. Fonte: (X-Plane, 2014b).	18
2.7	Simulador de voo Microsoft Flight Simulator. Fonte: (Microsoft Flight, 2014).	19
3.1	Cenário ilustrativo para o planejamento de missão.	26
3.2	Representação de regiões factíveis. (a) região factível fora do obstáculo. (b) região factível dentro do obstáculo. Cada região factível foi aproximado por um triângulo e por um conjunto de restrições lineares. Fonte: adaptado de (Ono et al., 2013).	32
4.1	Codificação dos controles do problema.	34
4.2	Árvore de decisão utilizada na escolha de qual algoritmo será executado.	39
4.3	Rotas candidatas usando Heurística Gulosa.	40
4.4	Estrutura hierárquica em árvore do AGMP.	43
4.5	Arquitetura do planejador de rota de pouso proposto.	44
5.1	Mapas gerados automaticamente: Em (a), (c) e (e) a área de cobertura é de 25%. Em (b), (d) e (f) a área de cobertura é de 50%. Os mapas (a) e (b) são fáceis. Os mapas (c) e (d) são normais. Os mapas (e) e (f) são difíceis.	49
5.2	Análise das forças envolvidas na aeronave. Fonte: Adaptado de (Stuart e Dahmen, 2006).	54
5.3	Mudança de velocidade da aeronave partindo do repouso e com aceleração constante.	55
5.4	Exemplo de rotas dada a variação das situações críticas: Em (a), (b), (c) e (d) as situações são ψ_m , ψ_b , ψ_{s^1} e ψ_{s^2} respectivamente.	60

Lista de Tabelas

2.1	Detalhes técnicos da aeronave Tiriba. Fonte: (Branco et al., 2011; Braga et al., 2011).	12
2.2	Detalhes de acidentes envolvendo VANT militares. Fonte: Adaptado de (Dro- neWars, 2014).	12
2.3	Comparações entre os simuladores de voo: FightGear, X-Plane e Microsoft Flight Simulator.	19
5.1	Configurações utilizadas no gerador de mapas para cada tipo de categoria. . . .	51
5.2	Notação utilizada nas instâncias.	51
5.3	Configurações do Mapa, VANT e Pesos das Regiões usados nos experimentos. .	56
5.4	Configurações do computador utilizado nos experimentos.	57
5.5	Análise de sensibilidade do AG e AGMP.	58
5.6	Resultado obtido para diferentes situações críticas enfrentadas pelo VANT. . . .	59
5.7	Resultado obtido para a HG, para todos as falhas e todas as instâncias.	61
5.8	Resultado obtido para o AG1 e AG2, para todos as falhas e todas as instâncias.	62
5.9	Resultado obtido para o AGMP1 e AGMP2, para todos as falhas e todas as instâncias.	63
6.1	Cronograma de execução das atividades.	66

Lista de Algoritmos

1	Heurística Gulosa.	39
2	Algoritmo Genético.	41
3	Algoritmo Genético Multi-Populacional.	42
4	Algoritmo Gerador Automático de Mapas.	50

Lista de Abreviaturas

ACFS	<i>(Advanced Concepts Flight Simulator)</i>
AE	Algoritmo Evolutivo (<i>Evolutionary 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 (<i>Autopilot</i>)
ARARA	Aeronave de Reconhecimento Autônoma e Remotamente Assistida
ATC	Controle de Tráfego (<i>Air Traffic Control</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 Aterragem Convencional (<i>Conventional Take-off and Landing</i>)
DE	Evolução Diferencial (<i>Differential Evolution</i>)
DGPS	Sistema de Posicionamento Global Diferencial (<i>Differential Global Positioning System</i>)
EE	Estratégia Evolutiva (<i>Evolutionary Strategy</i>)
ELP	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>)
GPS	Sistema de Posicionamento Global (<i>Global Positioning System</i>)
HG	Heurística Gulosa (<i>Greedy Heuristic</i>)
ICAO	Organização de Aviação Civil Internacional (<i>International Civil Aviation Organization</i>)
IFA	Consciência em Voo (<i>In-Flight Awareness</i>)
MA	Algoritmo Memético (<i>Memetic Algorithm</i>)
MOSA	(<i>Mission Oriented Sensor Array</i>)
PE	Programação Evolutiva (<i>Evolutionary Programming</i>)
PLIM	Programação Linear Inteira Mista (<i>Mixed-Integer Linear Programming</i>)
PIM	Programação Inteira Mista (<i>Mixed-Integer Programming</i>)
PSO	Otimização por Enxame de Partículas (<i>Particle Swarm Optimization</i>)
PG	Programação Genética (<i>Genetic Programming</i>)
SI	Inteligência de Enxame (<i>Swarm Intelligence</i>)
SISVANT	Sistema de Veículo Aéreo Não Tripulado
S&A	Detectar e Evitar (<i>Sense and Avoid</i>)

USAF Força Aérea dos Estados Unidos (*United States Air Force*)
VANT Veículo Aéreo Não Tripulado (*Unmanned Aerial Vehicle*)

Lista de Símbolos

Φ_j	Conjunto de regiões do mapa $\Phi_j = \{Z_{\Phi_j}^1, Z_{\Phi_j}^2, \dots, Z_{\Phi_j}^{ \Phi_j }\}$ com $j \in \{n, p, b, r\}$
ϕ_n	Conjunto de regiões do tipo não navegável (proibido sobrevoar esta região)
ϕ_p	Conjunto de regiões do tipo penalizadoras (região péssima para pouso)
ϕ_b	Conjunto de regiões do tipo bonificadoras (região boa para pouso)
ϕ_r	Conjunto de regiões do tipo restante (região não classificada)
$Z_{\Phi_j}^i$	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
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
Ψ	Situação crítica ocorrida na aeronave ($\Psi = \{\psi_m, \psi_b, \psi_{s^1}, \psi_{s^2}, \psi_{\emptyset}\}$)
ψ_m	Situação crítica problema no motor
ψ_b	Situação crítica problema na bateria
ψ_{s^1}	Situação crítica problema nas superfícies aerodinâmicas tipo 1
ψ_{s^2}	Situação crítica 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_k com $k \in \{m, b, s^1, s^2, \emptyset\}$
x_t	Conjunto de estados da aeronave no tempo t ($x_t = [p_t^x, p_t^y, v_t, \alpha_t]$)
p^x	Posição no eixo x do VANT
p^y	Posição no eixo y do VANT
v	Velocidade do VANT
α	Ângulo (direção) do VANT
u_t	Conjunto de controles da aeronave no tempo t ($u_t = [a_t, \varepsilon_t]$)
a	Aceleração do VANT
ε	Variação angular (rotação) do VANT em torno do eixo z
ω_t	Perturbação independente do estado no instante t
Σ_t	Incerteza de cada estado da aeronave em cada instante t
Q	Matriz de covariância dos estados da aeronave
F^d	Força do arrasto, ou equação do arrasto, ou ainda, equação da resistência do ar
C_d	Coefficiente de arrasto este valor é particular para cada aeronave
ρ	Massa específica do fluido (ar) que a aeronave está inserida
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>)
ϕ^*	Local do pouso emergencial do VANT associado a uma região ϕ qualquer
\mathcal{W}	<i>Fitness</i> associada a solução
$\mathcal{M}_F^{25\%}$	Mapa com dificuldade Fácil e com área de cobertura de 25%
$\mathcal{M}_F^{50\%}$	Mapa com dificuldade Fácil e com área de cobertura de 50%
$\mathcal{M}_N^{25\%}$	Mapa com dificuldade Normal e com área de cobertura de 25%
$\mathcal{M}_N^{50\%}$	Mapa com dificuldade Normal e com área de cobertura de 50%
$\mathcal{M}_D^{25\%}$	Mapa com dificuldade Difícil e com área de cobertura de 25%
$\mathcal{M}_D^{50\%}$	Mapa com dificuldade Difícil e com área de cobertura de 50%

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 descon-
fiar.”*

Santos Dumont

1.1 Contextualização

A presente dissertação estuda problemas relacionados ao planejamento de rota para Veículos Aéreos Não Tripulados (VANTs) durante o voo, após a ocorrência de uma situação crítica. Nesse caso, a missão em curso pela aeronave precisa ser abortada e um replanejamento da rota se faz necessário, onde o objetivo principal é pousar a mesma sem riscos à segurança de pessoas, propriedades em solo e a própria aeronave. A situação crítica pode ser representada por falhas em sensores do VANT, pane no motor, superaquecimento da bateria, entre outras situações que obriguem ele a retornar à base ou a realizar um pouso emergencial. Logo, diversas falhas que um VANT poderá apresentar serão estudadas e algoritmos que façam o pouso da aeronave com

segurança serão propostos. Modelos matemáticos ou reformulação de modelos existentes serão desenvolvidos para o contexto das situações críticas consideradas.

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 de VANTs ao espaço aéreo, a probabilidade de falha deve ser menor ou igual aquelas 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, segundo Mattei (2014), o que significa capacidade de detectar riscos e tomar ações adequadas visando a segurança. Uma vez que segurança é o fator mais importante na integração dos VANTs ao espaço aéreo, é preciso criar formas de trazer novamente a bordo os recursos que foram eventualmente perdidos com a saída dos pilotos (Mattei et al., 2013; Fonseca et al., 2013; Figueira et al., 2013).

O voo de um VANT pode ser modelado como um sistema dinâmico e estocástico, influenciado por diversos fatores internos e externos (Li, 2010; Ono et al., 2013; Blackmore et al., 2011; Richards e How, 2002). O objetivo deste trabalho é maximizar a segurança dos voos, através da elaboração de algoritmos que permitam conduzir a aeronave em situação crítica até uma região segura para pouso. Deve-se ressaltar que, mesmo durante situações críticas, o VANT pode ser capaz de decidir ativar procedimentos que permitam retornar à base ou mesmo acionar mecanismos como paraquedas. Assim, os métodos desenvolvidos acionarão os procedimentos mais adequados de forma a privilegiar aspectos de segurança.

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 destes acidentes

com VANTs é ocasionada pela tomada de decisões errada por parte de pilotos localizados em terra. Assim, surge a necessidade de se construir sistemas planejadores de rotas para VANTs capazes de oferecer alternativas ao piloto em solo, ou mesmo executar o planejamento de uma nova rota de forma autônoma.

Uma série de acidentes envolvendo aeronaves do tipo *drones*, usadas para fins recreativos, levaram a queda de tais aeronaves próxima a pessoas, sem deixar vítimas, mas já causaram mortes em alguns casos como reportado em (Smartdrones, 2013a,b,c; Mortimer, 2012). Há também registros de acidentes com VANTS em aplicações militares norte-americanas, onde mais de 400 acidentes em todo o mundo ocorreram desde 2001 (OGlobo, 2013).

Um elemento que justifica o estudo do replanejamento de rotas na ocorrência de situações críticas é a ausência de trabalhos a respeito. O comportamento da aeronave diante de cada tipo de problema exige uma compreensão maior a respeito de como 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 a quedas de aeronaves civis 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). Estas falhas somam juntas mais de 60% dos acidentes envolvendo aeronaves tripuladas.

Portanto, entender cada uma das falhas no contexto dos VANTs e desenvolver algoritmos capazes de lidar com cada uma delas são fatores relevantes que justificam o presente estudo. Estudos deste tipo poderão auxiliar tanto na maior autonomia de voo quanto na melhoria do nível de segurança da aeronave. Logo, pretende-se justificar tal estudo através do desenvolvimento de um planejador de rotas robusto o suficiente para um conjunto de situações críticas.

1.3 Objetivos

O presente trabalho de mestrado visa promover e aplicar maior segurança aos voos de VANTs, projetando algoritmos planejadores de rotas que levem em conta as principais situações críticas que podem ocorrer com tais aeronaves.

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

- Investigar e elencar as principais situações críticas existentes em VANTs e suas causas.
- Desenvolver modelos matemáticos que representem os controles, estados e dinâmica da aeronave, considerando as falhas críticas que podem ocorrer durante seu voo.
- Elaborar cenários (mapas) para voo, através de um gerador automático de mapas.
- Avaliar e implementar Algoritmos Genéticos (AGs) capazes de tratar as principais situações críticas consideradas.
- Avaliar o desempenho e comportamento dos algoritmos em experimentos *offline* e em simulador de voo.
- Comparar os resultados obtidos com outras técnicas da literatura utilizadas para resolver problemas semelhantes.

1.4 Contribuições e Limitações

As contribuições teóricas esperadas com o desenvolvimento da presente dissertação são: obtenção de modelos matemáticos que representem a dinâmica da aeronave avariada e determinem a rota de pouso ótima para determinada situação crítica; contribuir com o estado da arte em planejamento de rotas para VANTs em caso de situações críticas, através do desenvolvimento de algoritmos genéticos para tal planejamento. Uma possível contribuição prática deste trabalho 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. A aeronave precisa conhecer o cenário de voo, onde qualquer eventual mudança no cenário pode afetar os resultados. Logo, um sistema embarcado que possibilite acesso a tais informações precisa ser estabelecido visando aplicações em cenários reais. Uma segunda limitação está na complexidade dos modelos matemáticos capazes de representar a dinâmica da aeronave no contexto considerado. Isso pode limitar o uso de *solvers* como CPLEX capazes de garantir a

obtenção de uma solução ótima. Por último, outra limitação é a dificuldade inerente a realização de experimentos reais em campo. Trata-se de uma tarefa complexa que demanda tempo e recursos financeiros quando o objeto de estudo é um VANT.

1.5 Organização

A organização deste documento foi subdividida da seguinte forma:

- **Capítulo 1:** apresentou a contextualização do problema, trazendo as motivações, justificativas, objetivos, contribuições e limitações do presente trabalho;
- **Capítulo 2:** apresenta uma revisão bibliográfica dos principais conceitos envolvidos em VANTs, AGs e IFA assim como alguns trabalhos relacionados;
- **Capítulo 3:** define o problema a ser estudado, apresentando uma descrição do mesmo, a modelagem matemática inicial 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 ser desenvolvida;
- **Capítulo 5:** apresenta alguns dos resultados preliminares obtidos nesta dissertação como: gerador automático de mapas; modelo de dinâmica do VANT e os experimentos propriamente ditos;
- **Capítulo 6:** mostra o cronograma detalhado das atividades executadas e a serem executadas durante este trabalho;
- **Capítulo 7:** apresenta as considerações finais.

1.6 Conclusão

Vimos neste capítulo que o presente trabalho visa construir um planejador de rotas em caso de emergência que aumente a segurança de voo associada a VANTs, contribuindo para sua inserção dentro do espaço aéreo.

Este capítulo apresentou os primeiros conceitos relacionados ao tema da dissertação e julgados essenciais à compreensão do seu desenvolvimento. Fizemos uma breve contextualização, apresentamos a motivação e justificativa do trabalho, os objetivos a serem alcançados, e ao final listamos possíveis contribuições e limitações do trabalho.

Revisão Bibliográfica

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

Isaac Newton

2.1 Introdução

Este capítulo apresenta uma revisão dos principais conceitos utilizados na compreensão deste 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, será mostrado os trabalhos relacionados a esta dissertação.

2.2 Conceitos Importantes

A presente dissertação utilizará conceitos relacionado aos VANTs para melhor 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 Veículo Aéreo Não Tripulado (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 VANT, os componentes necessários a decolagem, voo, pouso do veículo, os meios necessários para a realização da missão, a estação de pilotagem remota, meios para comunicações e controle, entre outros. De acordo com essa definição, este trabalho se encaixa na categoria de SISVANT, onde aqui serão desenvolvidos componentes que permitam o replanejamento da rota na ocorrência de uma situação crítica.

O primeiro protótipo de VANT desenvolvido no Brasil foi o BQM-1BR, pela extinta Companhia Brasileira de Tratores (CBT) (Paula, 2009). A aeronave teria realizado um voo em 1983, mas com a falência da CBT o protótipo do BQM-1BR ficou preservado no Museu “Asas de Um Sonho” da empresa TAM (Paula, 2009, pág. 2). A Figura 2.1 mostra uma foto do BQM-1BR.



Figura 2.1: Veículo aéreo não tripulado BQM-1BR. Foto: Rodrigo Zanette.

O Projeto Aeronave de Reconhecimento Autônoma e Remotamente Assistida (ARARA) foi desenvolvido numa parceria entre o Instituto de Ciências Matemáticas e Computação da

Universidade de São Paulo (ICMC-USP) e a Empresa Brasileira de Pesquisa Agropecuária (EMBRAPA), voltado especialmente para utilização do VANT em agricultura de precisão. Tal projeto deu origem em 2005 ao primeiro VANT de asa fixa desenvolvido com tecnologia brasileira (Wikipédia, 2014a). O Arara M1 é uma aeronave proveniente dessa parceria que foi utilizada no monitoramento, reconhecimento e vigilância para operações militares (AGX, 2014). A Figura 2.2 apresenta o Arara M1.



Figura 2.2: Veículo aéreo não tripulado Arara M1. Foto: (AGX, 2014).

O projeto da aeronave Tiriba foi desenvolvido em 2011 pela empresa AGX (Figura 2.3) em parceria com o ICMC-USP. O resultado obtido 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, 2014a).



Figura 2.3: Veículo aéreo não tripulado Tiriba. Foto: (AGX, 2014).

A Tabela 2.1 apresenta as principais características técnicas do Tiriba. O Tiriba será tomado como base neste trabalho para a modelagem da dinâmica de um VANT diante de uma situação crítica.

Tabela 2.1: Detalhes técnicos da aeronave Tiriba. Fonte: (Branco et al., 2011; Braga et al., 2011).

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	60Km/h a 100Km/h
Velocidade Terminal Considerada	110Km/h
Velocidade de Sustentação Considerada	40Km/h
Aceleração Máxima Considerada	2m/s ²

2.2.1.1 Situações Críticas em VANTs

Não há muitas fontes que divulguem as causas de acidentes envolvendo aeronaves não tripuladas. A Tabela 2.2 apresenta um conjunto de problemas que ocorrem em VANTs organizado pelo Conselho de Investigação de Acidentes (AIB) da Força Aérea dos Estados Unidos (USAF) (DroneWars, 2014).

Tabela 2.2: Detalhes de acidentes envolvendo VANT militares. Fonte: Adaptado de (DroneWars, 2014).

Identificação da falha		Ano de ocorrência								
Id	Falha ocorrida	2007	2008	2009	2010	2011	2012	2013	2014	Total
1	Falha mecânica		3	5	1		3	2	1	15
2	Falha do motor	3	1	1						5
3	Falha elétrica		2	2			1			5
4	Erro do piloto	1	1		3					5
5	Falha de comunicação ou perda de sinal	1					1			2
6	Falha no sistema de alimentação (bateria)	1								1
7	Falha eletrônica							1		1
8	Colisão com avião de carga					1				1
9	Perda de GPS						1			1
10	Problemas nas hélices							1		1
11	Condições meteorológicas		1							1
12	Colidiu com terminal do aeroporto		1							1
13	Desgaste da fuselagem					1				1
14	VANT abatido			1						1
15	Falhas mecânicas e condições meteorológicas			1						1
16	Desconhecida	2	4	11	15	23	18	18	3	94
	Total de acidentes	8	13	21	19	25	24	22	4	136

As falhas mencionadas servirão de base para os estudos desta dissertação, onde as situações críticas consideradas serão detalhadas no Capítulo 3. Porém, vale ressaltar que falhas da Tabela 2.2 como a Id:4 não ocorreria através do sistema proposto dada a ausência de piloto. A falha Id:14 é complexa e restrita ao ambiente militar, assim não será tratada neste estudo. As

situações id:5, 8, 9, 11, 12, 13 e 15 também não serão inicialmente consideradas por serem específicas e de difícil modelagem. Logo, as falhas consideradas neste trabalho inicialmente são: id:2 e 10 que serão consideradas como falha no motor; id:6 será assumida como problema na bateria; id:1, 3 e 7 serão falhas que impedem o VANT de, por exemplo, 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 o *aileron*, responsáveis pelas curvas à esquerda e à direita.

A Tabela 2.2 relatou um total de 42 falhas conhecidas do total de 136. Considerando as situações críticas descritas acima, consegue-se tratar 28 ocorrências de falhas de um total de 42 conhecidas. Assim, caso o método desenvolvido consiga tratar todas as situações críticas, um tratamento satisfatório será dado a 66,6% das avarias conhecidas representando 20,5% do total.

2.2.1.2 Terminologia

Um conjunto de termos utilizados nesta dissertação sobre a área de veículos aéreos estão definidos a seguir:

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

Aeronave de Asa Rotativa: aeronave em que suas asas são representadas 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 aterragem convencional efetuado sobre pistas. Alguns exemplos desse tipo de pouso são os efetuados por aviões tradicionais.

2.2.2 Algoritmos Genéticos

Os métodos de computação evolutiva são metaheurísticas baseadas em população de soluções (*population-based*). As técnicas para gerar e testar soluções, explorando e intensificando a busca no espaço de soluções do problema, são inspirados 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 e Smith, 2003). No contexto deste projeto, pretende-se desenvolver um método para planejamento de rotas em situação crítica baseado em AGs.

Os AGs foram inicialmente proposto 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 se evolui a população de indivíduos 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 um novo indivíduo é gerado. 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.

2.2.2.1 Terminologia

A seguir serão apresentados alguns conceitos envolvendo AGs utilizados neste trabalho:

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. Por exemplo, o Torneio- q é um operador de seleção e seu funcionamento consiste no sorteio aleatório de q indivíduos da população. Em seguida, é escolhido o melhor indivíduo entre os q selecionados para participar do processo de cruzamento ou para integrar a próxima geração.

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

Mutação: É um operador que altera de forma randômica parte da informação codificado 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.

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 existentes na representação da melhor solução.

2.2.3 In-Flight Awareness

O conceito chamado *In-Flight Awareness* (IFA) está relacionado à ideia de Consciência Situacional (CS) em aeronaves não tripuladas. A partir de eventos ocorridos interna ou externamente à aeronave, o sistema IFA verifica as causas para os sintomas considerando as capacidades da plataforma e os dados dos sensores embarcados, sejam eles dados ambientais, de tráfego ou de funcionamento 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.

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 et al. (2006) usa a definição de Endsley (1988), mas alterada de forma a tornar mais fácil o reconhecimento da presença de grupos de pessoas e de VANTs durante uma operação. O trabalho de Drury et al. (2006) também analisou alguns dos fatores que afetam um voo e, consequente-

mente, 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, Endsley e Jones (2006) apresentaram outra definição para CS: “entendimento compartilhado do subconjunto de informações necessário para cada um de seus objetivos”. Esta 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 de emergência, a informação necessária e correta deve estar disponível no momento certo e no formato certo.

Os sistemas para evitar acidentes envolvem radares *Air Traffic Control* (ATC), sistemas *Sense and Avoid* e os pilotos a bordo como uma medida final. Tendo em conta 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 e representa um componente da mesma, juntamente com o Piloto Automático (AP) e o MOSA (*Mission Oriented Sensor Array*). A Figura 2.4 mostra a relação entre o IFA, piloto automático e MOSA. A principal tarefa do MOSA é o cumprimento da missão, o mesmo possui acesso a um conjunto de sensores da aeronave (Figueira et al., 2013). MOSA é o responsável por todos os aspectos relacionados à missão, controlando não somente a carga útil, mas também a trajetória de voo. O VANT navega usando um piloto automático para manter um plano de voo pré-definido para realizar a missão. Todas as responsabilidades ligadas

as questões de segurança são destinadas ao IFA, assim o mesmo pode utilizar as informações provindas de uma série de sensores da aeronave para garantir a segurança. Este trabalho visa integrar ao IFA um planejador de rotas que leve o VANT até uma região considerada segura na ocorrência de uma situação crítica.

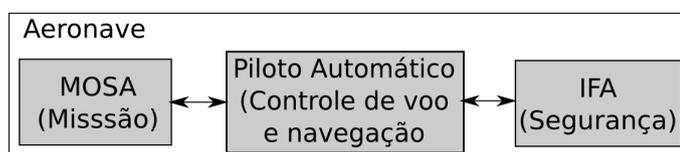


Figura 2.4: Esquema da integração entre IFA, piloto automático e MOSA na aeronave.

2.3 Simuladores de Voo

Segundo Banks (2000), simulação é a imitação da operação de um processo do mundo real ou do sistema ao longo do tempo. 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 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, 2014a). A Figura 2.5 apresenta uma tela do simulador. FlightGear se caracteriza pelo seu realismo e por possuir uma grande variedade de aeronaves, aeroportos e cenários disponíveis para *download*. O desenvolvimento deste 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, 2014b).



Figura 2.5: Simulador de voo FlightGear. Fonte: (FlightGear, 2014b).

O motor de simulação SimGear é utilizado pelo FlightGear, o qual é um software usado para simulações independente do FlightGear (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, 2014c). Esse simulador utiliza a biblioteca OpenGL para desenhar rotinas gráficas 3D (FlightGear, 2014e).

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) (X-Plane, 2014a). O X-Plane implementa um modelo de voo realista baseado nas forças exercidas sobre múltiplas seções de uma aeronave (Wikipédia, 2014c). A Figura 2.6 mostra a tela de interface do simulador X-Plane.



Figura 2.6: Simulador de voo X-Plane. Fonte: (X-Plane, 2014b).

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

2.3.3 Microsoft Flight Simulator

Microsoft Flight Simulator é um jogo de computador que também é considerado simulador de voo (Microsoft Flight, 2014). Na Figura 2.7 é apresentada uma tela do simulador de voo Flight Simulator. Ele é distribuído na forma *Free-to-Play*, ou seja, o jogo é gratuito mas 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 o seu desenvolvimento foi cancelado (Wikipédia, 2014d).



Figura 2.7: Simulador de voo Microsoft Flight Simulator. Fonte: (Microsoft Flight, 2014).

2.3.4 Comparação dos Simuladores

A Tabela 2.3 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 que o simulador foi desenvolvido.

Tabela 2.3: Comparações entre os simuladores de voo: FlightGear, X-Plane e Microsoft Flight Simulator.

Simulador	Lançamento	Última Versão	Sistema Operacional	Licença	Distribuição	Linguagem
FlightGear	1997	2014	Windows, Linux, Mac OS X, FreeBSD, Solaris e IRIX	GPL	Grátis	C/C++
X-Plane	1993	2014	Windows, Linux e Mac OS X	Proprietário	Pago	C/C++
MS Flight Simulator	1982	2012	Windows	Proprietário	<i>Free-to-Play</i>	C/C++/C#

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 executando 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. Pretende-se utilizar o simulador FlightGear na etapa de experimentos *online*, pois o mesmo suporta multiplataforma, sua distribuição é gratuita e se trata de um sistema ainda em constante atualização.

2.4 Trabalhos Relacionados

O planejamento de missão para veículos autônomos foi abordado em Li (2010); Blackmore et al. (2011); Ono et al. (2013), onde foi modelado como um problema de Programação Inteira e 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: utiliza uma representação compacta para todos os planos factíveis, sendo o primeiro trabalho a tratar o planejamento para ações contínuas e discretas em conjunto; uma formulação para definir em alto nível o conjunto das ações que o VANT pode tomar e o *scheduling* dos objetivos da missão. Entretanto, no sistema proposto não considerava a alocação de risco na tomada de decisão.

Os autores em Blackmore et al. (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 da missão não é considerada pelos autores, tratando 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 et al. (2013), onde um novo sistema planejador é proposto e melhorias no tratamento de alocação de risco em relação a Blackmore et al. (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. Este trabalho foi validado em dois cenários: Sistema de Transporte Pessoal e Sistema de Transporte de Carga Aeroespacial.

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 crítica do avião, o piloto deve primeiro recuperar o controle da aeronave, 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 (ELP), avalia esses fatores e propõe rotas, bem como possíveis locais de pouso para as aeronaves que utilizando um algoritmo A* híbrido.

Em um trabalho posterior Meuleau et al. (2011), o comportamento do algoritmo A* proposto em Meuleau et al. (2009) foi analisado 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 ELP projetado considera apenas aeroportos e pistas oficialmente reconhecidos. A simulação considera panes que ocorrem entre 1 a 3 minutos de voo, onde o algoritmo ELP é acionado para auxiliar os pilotos. O algoritmo exhibe quais as possíveis rotas de pouso de emergência da aeronave para o piloto da aeronave.

Algoritmos evolutivos tem sido utilizados no planejamento de rotas para VANTs. O autor em Pehlivanoglu (2012) utilizou AG com diagrama de Voronoi para planejamento de rotas de VANTs 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 Mehmet (2012) foi utilizado AG no planejamento de rotas para robôs móveis terrestres. 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 nesse trabalho é representado por um 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 (PSO) e Evolução Diferencial (DE). Os resultados indicaram que o GA foi o melhor método, seguido pelo PSO e DE respectivamente.

Evolução Diferencial (DE) é aplicada em Zhang e Duan (2015) para planejamento de rota para VANT em ambientes tridimensionais. A rota é projetada para ter curta duração e baixa altitude de voo. O VANT deve evitar regiões não navegáveis, regiões com radares, mísseis e armas antiaérea. 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 DE para cenários de busca e resgate em tempo real com VANTs. Esse trabalho lida com um conjunto de VANTs autônomos utilizados em conjunto para tratar a tarefa de resgate em uma grande área. Os resultados são comparados com uma estratégia baseada em Inteligência de Enxame (SI). A abordagem evolutiva mostrou-se melhor que a SI, pois a SI diminui o seu desempenho com o aumento do número de alvos de resgate o que não ocorre com a DE.

O presente trabalho propõe o uso de Algoritmo Genético Multi-Populacional (AGMP), onde indivíduos são hierarquicamente estruturados em árvores. Essa abordagem foi introduzida por França et al. (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 (Moghaddam et al. (2008)), problema do caixeiro viajante (Buriol et al. (2004)), ordenação de dados (Moscatto et al. (2007)), entre outros, com resultados relevantes alcançados.

Os estudos conduzidos em Li (2010); Blackmore et al. (2011); Ono et al. (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 no ambiente. Entretanto, não é considerado o aparecimento de situações críticas, por exemplo o planejamento considera que os equipamentos tenham imprecisões conhecidas, por isso os trabalhos tratam a alocação de risco e não é considerado que possa haver uma falha total em um equipamento. Por exemplo, em Blackmore et al. (2011) o problema era planejar uma rota de um ponto origem até um ponto destino desviando de obstáculos.

Essa dissertação apresenta uma formulação matemática para o problema de replanejamento de caminho baseada no modelo descrito em Blackmore et al. (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 considera-se falhas em aeronaves, o presente estudo se assemelha ao descrito em Meuleau et al. (2009), no entanto, difere do mesmo por lidar com VANTs, possuir um conjunto diferente de falhas associadas e por ser uma falha severa que leva 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, esse trabalho se assemelha com Toledo et al. (2012, 2013) aplicando técnicas de computação evolutiva.

2.5 Conclusão

Este capítulo apresentou os principais conceitos envolvidos com VANTs; algoritmos genéticos e IFA. A relação do IFA com os componentes da aeronave AP e MOSA foi estabelecida e alguns simuladores de voo também foram apresentados. Todos estes conceitos são importantes e dão suporte para a compreensão dos capítulos seguintes. Alguns trabalho relacionados também foram avaliados neste capítulo que servirão de base para o desenvolvimento da modelagem matemática e dos métodos de solução para o problema estudado.

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 definir formalmente os objetivos e restrições considerados neste trabalho.

3.2 Descrição do Problema

O problema está descrito nessa seção a partir da situação ilustrada pela Figura 3.1. Há um cenário composto por duas regiões povoadas (casas), florestas, uma planície, uma área sob tempestade, um aeroporto, uma região observada e uma pista pequenas aeronaves. Suponhamos que a missão a ser realizada pelo Veículo Aéreo Não Tripulado (VANT) seja seguida por uma situação crítica como descrito a seguir, baseado na missão de (Ono et al., 2013):

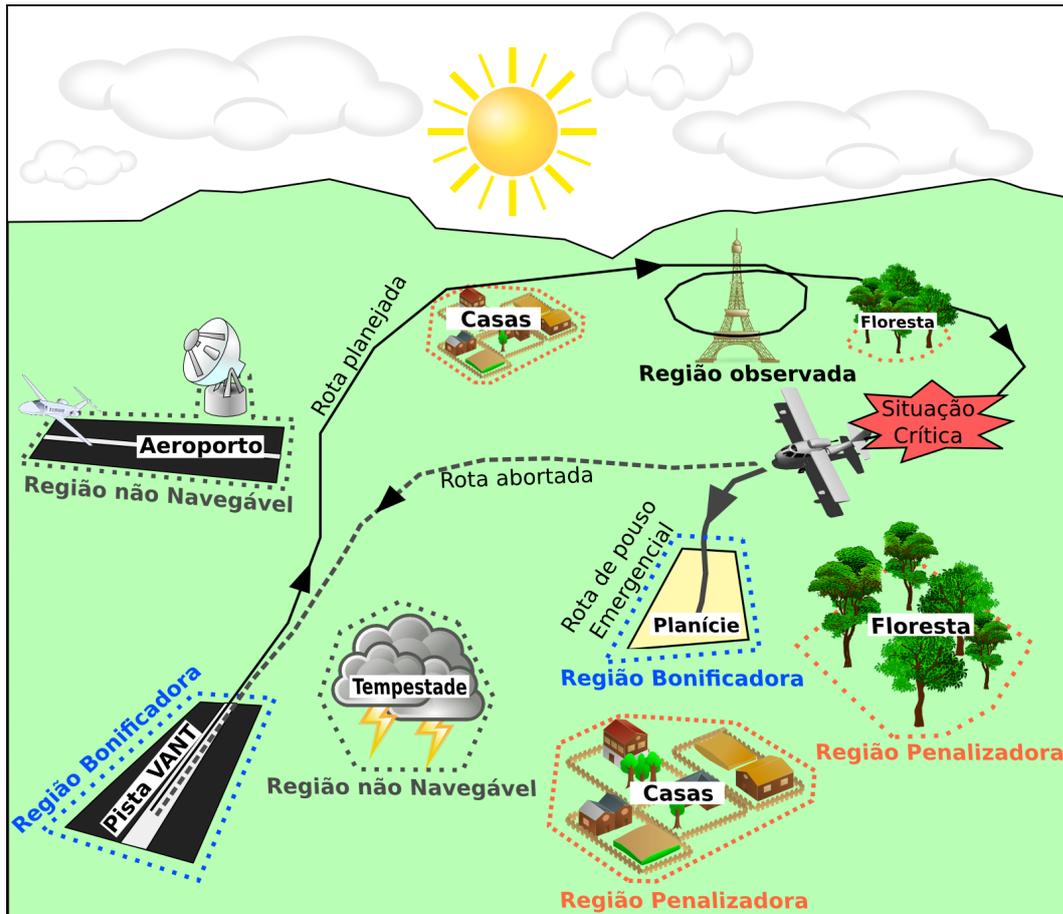


Figura 3.1: Cenário ilustrativo para o planejamento de missão.

Missão: Partindo da pista do VANT, ele deve alcançar em 20 minutos a região observada e permanecer nela retirando fotografias entre 5 e 10 minutos. Em seguida, a aeronave deve retornar à pista de pouso do VANT. Em todo o caminho, ele deve ficar em uma região de segurança evitando regiões não navegáveis, como a tempestade que se aproxima e o aeroporto. A aeronave pode voar sobre áreas povoadas e florestas. A missão deve ser concluída no prazo de 60 minutos.

Situação crítica: Durante a execução da missão, os sistemas da aeronave detectam um determinado problema na aeronave como o superaquecimento da bateria. Tais sistemas são capazes de abortar a missão e acionar o algoritmo para replanejamento da rota atual. Tal replanejamento a partir de agora busca minimizar a probabilidade de danos durante o pouso, considerando as informações sobre as regiões mencionadas e as limitações causadas pelo problema da aeronave.

3.2.1 Conjunto de Regiões Modeladas

Várias regiões podem ser mapeados previamente para a execução da missão e foram classificadas em conjuntos de regiões, baseadas em suas características. Estes conjuntos, representados por Φ , serão separados de acordo com a probabilidade de pouso da aeronave nestas regiões. Um total de quatro conjuntos foram definidos para os tipos de regiões consideradas por este 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, base militar e outras áreas com restrições ao voo VANT. As consequências devido a queda e a penalização por sobrevoar estas regiões é altíssima.
2. **Conjunto Navegável com Penalização (ϕ_p):** O VANT pode sobrevoar regiões deste conjunto, mas não é desejável que o mesmo pouso sobre elas. As regiões sobre este conjunto podem representar áreas povoadas, fábricas, florestas, áreas íngremes, entre outras, onde o voo de aeronaves são permitidos, mas o seu pouso deve ser evitado. As consequências devido a queda da aeronave é bem relevante e pode acarretar prejuízos financeiros ou acidentes envolvendo pessoas. Haverá penalização considerável no caso de queda nestas regiões.
3. **Conjunto Navegável e Bonificador (ϕ_b):** O VANT pode sobrevoar e é desejado que o mesmo pouso numa das regiões deste conjunto. Este conjunto contém regiões planas e adequadas ao pouso como áreas gramadas ou campos com plantações rasteiras. As regiões de origem e destino do VANT também pertencem a este conjunto. As consequências devido a queda em uma dessas regiões são que a aeronave terá grandes chances de ser salva. Neste caso, há uma bonificação ao invés de penalização.
4. **Conjunto Restante (ϕ_r):** A aeronave pode sobrevoar e pousar nessas regiões. Este conjunto representa as áreas restantes que não foram classificados para o pouso. Não há restrições de 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 modeladas 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.

3.2.2 Situações Críticas Modeladas

Algumas situações hipotéticas, que fossem críticas à operação da aeronave, foram definidas com base no levantamento de dados descrito na Seção 2.2.1.1. O algoritmo de replanejamento deve ser informado sobre o tipo de limitação imposta à aeronave pelo problema detectado. Neste trabalho, foram consideradas algumas situações críticas, representadas por Ψ , para a operação de aeronaves:

1. **Problema no Motor (ψ_m):** O motor m apresenta problemas e parou 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 acelerar a aeronave.
2. **Problema na Bateria (ψ_b):** A bateria b tem um superaquecimento, em que todos os controles funcionam, porém o VANT deve pousar o mais rápido possível. Caso o pouso demore a bateria e o VANT podem incendiar, ou ainda, parar de funcionar os componentes eletrônicos o que levaria a aeronave a queda imediata.
3. **Problema nas Superfícies Aerodinâmicas tipo 1 (ψ_{s^1}):** A aeronave apresenta uma falha em uma das asas s^1 que a torna capaz de virar apenas para a esquerda.
4. **Problema nas Superfícies Aerodinâmicas tipo 2 (ψ_{s^2}):** A aeronave apresenta uma falha em uma das asas s^2 que a torna capaz de virar apenas para a direita.
5. **Nenhum Problema (ψ_\emptyset):** A aeronave não apresenta nenhuma falha, logo todos os controles funcionam. Assim quando a aeronave está configurada desta forma o problema se transforma em uma simples missão de pouso sem nenhuma avaliação.

A situação crítica a que foi submetida a aeronave, durante o voo, será representada por ψ^* , onde $\psi^* \in \Psi$.

3.2.3 Hipóteses sobre 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 inicialmente serão consideradas apenas aquelas descritas na Seção 3.2.2. O pouso emergencial da aeronave ocorrerá também sempre a uma taxa constante de descida. Após o VANT atingir o solo, será considerado que o mesmo parou instantaneamente, atingindo uma velocidade abaixo da mínima.

Algumas outras condições assumidas neste trabalho são: velocidade do vento (ar) igual a zero; densidade do ar não muda com a altitude; constante de resistência do ar não muda com o ângulo da aeronave; o VANT possui todos os cenários (mapas) carregados em sua memória (conhecendo assim previamente todas as regiões catalogadas); os mapas são todos estáticos, ou seja, não mudam com o tempo.

Os algoritmos envolvidos na etapa de decolagem não são tratados nesse trabalho. A missão envolvida no momento do voo cruzeiro também não é tratada. A presente dissertação possui como foco o desenvolvimento a partir do momento em que houve alguma falha na aeronave.

Todas as hipóteses consideradas tem como objetivo definir o escopo inicial do trabalho, onde novas condições que permitam avançar no estudo do problema proposto poderão ser consideradas. Todavia, novas situações serão consideradas desde que viáveis dentro da limitação de tempo existente para conclusão desta dissertação.

3.3 Modelagem do Problema

Nós apresentamos uma formulação matemática para resumir todos os aspectos do problema abordado. Esta formulação vem de uma modelagem semelhante descrita em Blackmore et al. (2011) para planejamento de caminho com alocação de riscos. A seguir encontram-se alguns parâmetros e variáveis de decisão utilizados na formulação matemática do problema.

Parâmetros:

- $\Phi_j = \{Z_{\Phi_j}^1, Z_{\Phi_j}^2, \dots, Z_{\Phi_j}^{|\Phi_j|}\}$: Conjunto de regiões com $j \in \{n, p, b, r\}$;

- $Z_{\Phi_j}^i$: i -ésima região do conjunto Φ_j ;
- $|\Phi_j|$: Número de regiões no conjunto Φ_j ;
- C_{Φ_j} : Custo de pousar no conjunto Φ_j ;
- T : Número de passos de tempo para pousar o 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_k com $k \in \{m, b, s^1, s^2, \emptyset\}$;
- ω_t : Perturbação independente do estado, no instante step t ;
- ΔT : Discretização do tempo com intervalo de tempo fixo.

Variáveis de Decisão:

- x_t : Conjunto de estados do VANT ($x_t = [p_t^x, p_t^y, v_t, \alpha_t]^T$);
- 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 do VANT no instante t ;
- α_t : Ângulo do VANT no instante t sobre o eixo x ;
- u_t : Conjunto de controles do VANT ($u_t = [a_t, \varepsilon_t]^T$);
- a_t : Aceleração do VANT no instante t ;
- ε_t : Variação angular do VANT no instante t sobre o eixo x .

Formulação Matemática:

$$\text{Minimizar } \sum_{i=1}^{|\phi_p|} (C_{\phi_p} \cdot P(x_T \in Z_{\phi_p}^i)) - \sum_{i=1}^{|\phi_b|} (C_{\phi_b} \cdot P(x_T \in Z_{\phi_b}^i)) \quad (3.1)$$

sujeito a:

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

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

Uma série discreta e finita dos passos no tempo $t = 0, 1, \dots, T$ é assumido para pousar a aeronave. A variável de decisão x_t representa o estado do VANT no instante t , que é dada por sua coordenada no plano cartesiano (p_t^x, p_t^y) , velocidade (v_t) e direção angular neste plano (α_t) . A variável u_t representa os controles em t , onde a aceleração (a_t) e variação angular (ε_t) aplicados sobre as aeronaves estão definidos.

A função objetivo (3.1) visa minimizar danos executando um pouso de segurança, assim é formada por duas partes: penalização e bonificação. A primeira parte tem as penalidades (C_{ϕ_p}) , dada por uma rota que pouso 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}) , dada por uma rota que pouso a aeronave dentro das regiões bonificadoras ($x_T \in Z_{\phi_b}^i$). A função $P(\dots)$ utilizada nesta formulação representa a probabilidade da aeronave no estado x_t pertencer ou não a uma região particular em ϕ_i .

A restrição (3.2) descreve as transições de estados do VANT, onde a dinâmica dos estados do VANT no tempo $t + 1$ são definidos a partir de posições e controles aplicados no instante t adicionado da incerteza ω_t . A função F_{Ψ} depende de cada tipo de situação crítica ocorrida (Ψ). Tal como proposto em Blackmore et al. (2011), é assumido que o estado inicial tem uma distribuição gaussiana e há perturbações e incertezas representadas como um ruído gaussiano branco $\omega_t \sim \mathcal{N}(0, Q)$ com matriz de covariância Q . Além disso, o estado futuro também segue uma distribuição gaussiana e x_t é uma variável aleatória.

Assim, a localização do VANT não é precisa e há sempre o risco de se desviar de sua rota e atingir uma região não navegável, mas um limite para esse risco pode 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 .

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 , e tem um

conjunto de *waypoints* para a rota de pouso emergencial (veja Equação 3.4). O local de pouso da aeronave é representado por ϕ^* (veja Equação 3.5).

$$\Omega = \{(p_0^x, p_0^y), (p_1^x, p_1^y), (p_2^x, p_2^y), \dots, (p_T^x, p_T^y)\} \quad (3.4)$$

$$\phi^* = \{\phi \in \Phi \mid (p_T^x, p_T^y) \in Z_\phi\} \quad (3.5)$$

A Figura 3.2 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 3.2(a), utiliza o operador \vee que indica o OR lógico para todas as retas estarem fora do obstáculo. Uma outra situação possível é permanecer dentro de uma área passível de voo Figura 3.2(b), onde o operador \wedge representa o AND lógico para todas as retas permanecerem dentro de um obstáculo. Ambas as situações podem ser descritas como regiões factíveis aproximadas através de restrições lineares.

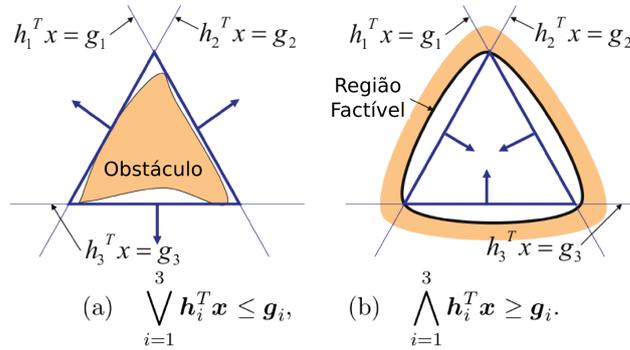


Figura 3.2: Representação de regiões factíveis. (a) região factível fora do obstáculo. (b) região factível dentro do obstáculo. Cada região factível foi aproximado por um triângulo e por um conjunto de restrições lineares. Fonte: adaptado de (Ono et al., 2013).

3.4 Conclusão

Este capítulo descreveu o problema abordado em maiores detalhes, onde foram definidos os conceitos de conjuntos de regiões e os tipos de situações críticas a serem tratadas. A partir desses conceitos, uma formulação matemática foi apresentada que considera tanto o planejamento da missão quanto a alocação de risco na rota planejada. Assim, tal formulação auxiliou no estabelecimento formal dos objetivos e das restrições do problema tratado.

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 a ser seguida no desenvolvimento deste trabalho. Inicialmente é mostrada a codificação do problema, bem como função objetivo, operadores de inicialização, *crossover* e mutação, a árvore de decisão dos algoritmos e sua ordem de execução. Os três algoritmos planejadores de rotas inicialmente propostos são descritos: a Heurística Gulosa (HG), o Algoritmo Genético (AG) e Algoritmo Genético Multi-Populacional (AGMP). Por fim, é mostrada a arquitetura do sistema planejador descrevendo as entradas e saídas do mesmo.

4.2 Codificação do Problema

A codificação do problema é uma etapa importante do desenvolvimento do método, pois uma boa representação da solução impacta na qualidade da busca no espaço de soluções do problema. Como os métodos devem determinar os controles de entrada da aeronave, que a leve a um local de pouso de forma a minimizar danos, uma codificação real foi definida onde os controles são elementos de aceleração e ângulo 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 4.1 exemplifica tal codificação.

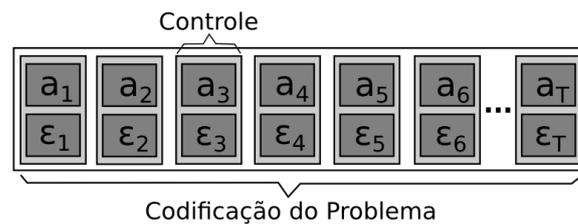


Figura 4.1: Codificação dos controles do problema.

Tal codificação é traduzida como uma rota a partir da decodificação dos controles em um conjunto de *waypoints* do VANT. A função de transição $F_{\Psi}(x_t, u_t)$ é usada para decodificar os controles através da Equação 5.3 melhor descrita na Seção 5.3. Esta função aproxima o movimento de um VANT de asa fixa num espaço 2D.

Dado que x_t é uma variável aleatória ($x_t \sim \mathcal{N}(\bar{x}_t, \Sigma_t)$), os autores em Blackmore et al. (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 pode-se utilizar a função de transição para calcular todos os próximos estados esperados ($\bar{x}_{t+1} = F_{\Psi}(\bar{x}_t, \bar{u}_t)$). No entanto, a incerteza Σ_t em todo estado esperado \bar{x}_t cresce a cada instante de tempo. Os autores em Ono et al. (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 . Neste trabalho, 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 $P(x_t \in Z_\phi^i)$ 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* para conseguir pousar o VANT como descrito pela Restrição (3.2) na Seção 3.3. É possível pousar a aeronave sem gastar todos os passos de tempo disponível, assim pode-se pousar em um tempo K tal que $0 < K \leq T$.

4.2.1 Função Fitness

A função objetivo utilizada pelos métodos propostos é baseada na função objetivo (3.1) da modelagem matemática 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:

$$fitness = f_{Pouso\phi_b} + f_{Pouso\phi_p} + f_{PousoEVoo\phi_n} + f_{Curvas} + f_{DistVANT\phi_b} + f_{ViolouT} + f_{\psi_b} \quad (4.2)$$

Expressões usadas:

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

$$f_{Pouso\phi_b} = -C_{\phi_b} \cdot \sum_{i=1}^{|\phi_b|} (P(x_K \in Z_{\phi_b}^i)) \quad (4.3)$$

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

$$f_{Pouso\phi_p} = C_{\phi_p} \cdot \sum_{i=1}^{|\phi_p|} (P(x_K \in Z_{\phi_p}^i)) \quad (4.4)$$

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

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

A Equação 4.6 prioriza rotas que evitem fazer curvas desnecessárias.

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

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

$$f_{DistVANT\phi_b} = \text{menor } Dist(\bar{x}_K, Z_{\phi_b}^i) \quad (4.7)$$

Se a aeronave tem velocidade final maior do que o seu valor mínimo, não ocorre de fato um pouso. Assim, 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 adicionado à função de *fitness*. O objetivo é reduzir o número de *waypoints*, em vez de gastar muito tempo para o pouso da aeronave.

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

4.2.2 Operadores Inicialização

Os operadores de inicialização criados para as estratégias evolutivas são descritos a seguir.

- **Inicialização Aleatória:** a inicialização aleatória gera valores com distribuição uniforme 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 $\varepsilon_t \in U[0.25\varepsilon_{min}; 0.25\varepsilon_{max}]$. Este

operador dá mais chance de aplicar controles mais suaves para as mudanças de direção na aeronave.

- **Inicialização Aceleração Curta:** o operador aceleração curta faz o contrário, ou seja, gera $\varepsilon_t \in U[\varepsilon_{min}; \varepsilon_{max}]$ e $a_t \in U[0.25a_{min}; 0.25a_{max}]$. Este operador dá mais chance de aplicar controles de acelerações mais suaves a aeronave, fazendo a aeronave pousar mais rapidamente.
- **Inicialização Gulosa:** a inicialização gulosa aplica a HG, explicada na próxima seção, para gerar um conjunto de controles que garantem atingir regiões bonificadoras.

4.2.3 Operadores Crossover

Os operadores de *crossover* implementados para lidar com tal codificação da solução utilizados pelas estratégias evolutivas são:

- **Crossover Média:** neste operador os controles do indivíduo filho é formado pela média aritmética dos controles dos pais.
- **Crossover Aritmético:** neste *crossover* os controles do indivíduo filho é formado pela média ponderada dos controles dos pais.
- **Crossover Geométrico:** nesta recombinação os controles do indivíduo filho é formado pela média geométrica dos controles dos pais.
- **Crossover OX:** no operador OX controles do indivíduo filho é composto pelos controles de ambos os pais tomados aleatoriamente.
- **Crossover BLX- α :** neste operador os controles do indivíduo filho é submetido a expressão: $w_i^{filho} = u_i^{pai1} + \beta(u_i^{pai1} - u_i^{pai2})$, onde $i = 1, 2, \dots, T$ e $\beta = U(-\alpha, 1 + \alpha)$, com U representando uma distribuição uniforme (Eshelman e Schaffer, 1993).

4.2.4 Operadores Mutação

Alguns operadores para mutação considerando representação real (Michalewicz e Schoenauer, 1996; Eiben e Smith, 2003; Herrera et al., 2005) foram utilizados pelas estratégias evolutivas implementadas:

- **Mutação Uniforme:** este operador faz um sorteio aleatório de algum controle e sorteia novos valores para sua aceleração e direção.
- **Mutação Limite:** o operador Limite faz um sorteio de duas variável binária, dependendo dos valores das variáveis os controles vão para o limite (máximo ou mínimo).
- **Mutação Creep:** este operador adiciona ao controle um pequeno valor aleatório. Provocando uma pequena perturbação no controle.

4.3 Árvore de Decisão

Este trabalho pretende desenvolver um conjunto de métodos para tratar o problema de pouso de aeronaves avaliadas. A Figura 4.2 mostra uma árvore de decisão que guia os métodos executados e sua ordem para resolver o problema. Primeiramente, o IFA detecta que a aeronave possui uma avaliação, em seguida, chama a Heurística Gulosa (HG) (passo 1), o método retorna sua melhor rota encontrada (passo 2). Caso a rota gerada não seja considerada factível, o IFA chama um segundo algoritmo, neste caso o Algoritmo Genético (AG) (passo 3) e o mesmo retorna sua melhor solução encontrada (passo 4). Este mesmo raciocínio é aplicado caso um método exato seja integrado ao sistema proposto. O estabelecimento desta ordem é importante, pois a HG tem a vantagem de retorna rapidamente uma solução, seguida pelo genético e pelo método exato que teoricamente seria o mais lento. Alguns resultados preliminares mostraram que a HG retorna uma solução na ordem de 0,05 segundos, enquanto o genético leva cerca de 1,0 segundo. Assim, acredita-se que esta estratégia seja promissora. Por exemplo, o AG pode ser executado em paralelo com a HG, a solução retornada mais rapidamente pela HG é implementada até que a solução do AG seja obtida, a partir da qual uma atualização nos *waypoints* poderia ser feita.

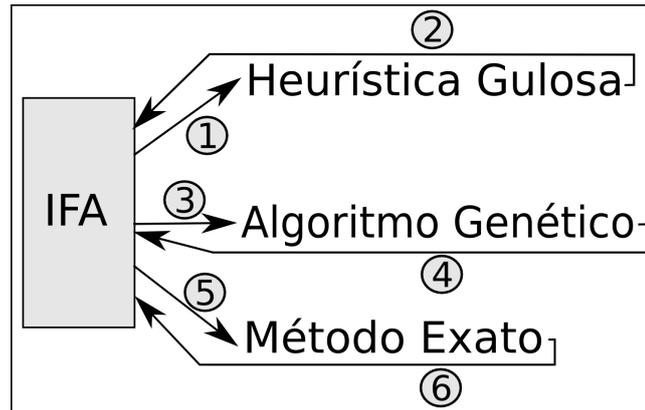


Figura 4.2: Árvore de decisão utilizada na escolha de qual algoritmo será executado.

As próximas seções descrevem os métodos que poderão ser chamados pela árvore de decisão e implementados nesse trabalho.

4.4 Heurística Gulosa

A Heurística Gulosa (HG) foi 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 em situações críticas, quando uma tomada de decisão rápida é necessária e uma solução viável torna-se mais interessante do que a obtenção de soluções próximas ao ótimo. O Algoritmo 1 descreve a HG mostrando as entradas, saídas e funcionamento do mesmo.

Algoritmo 1: Heurística Gulosa.

Entrada: Mapa: *mapa*

Saída: RotaPouso: *melhorRota*

```

1 início
2   RotaPouso rota[] ← instância RotaPouso()[mapa.| $\phi_b$ |];
3   para  $i = 1$  até mapa.| $\phi_b$ | faça
4     inicializar(rota[i], mapa. $Z_{\phi_b}^i$ );
5     avaliar(rota[i]);
6   fim para
7   RotaPouso melhorRota ← obterMelhorRota(rota);
8   retorna melhorRota;
9 fim

```

O HG possui como entrada o mapa contendo informações do cenário, a saída é a melhor rota para pouso encontrada. O funcionamento básico desta heurística é gerar uma solução candidata

para cada uma das regiões bonificadoras ($map.\phi_b$). Esta solução candidata é determinada rotacionando-se a aeronave até que a mesma esteja direcionada para uma região bonificadora ($map.Z_{\phi_b}^i$). Em seguida, calcula-se uma trajetória em linha reta até pousar nesta região. A rotação ocorre para o lado com menor ângulo entre o VANT e uma linha reta para uma região bonificadora. Isto é ilustrado pelos ângulos $\lambda_1 < \lambda_1'$ para a região b_1 na Figura 4.3, onde a rotação ocorre em direção ao ângulo λ_1 .

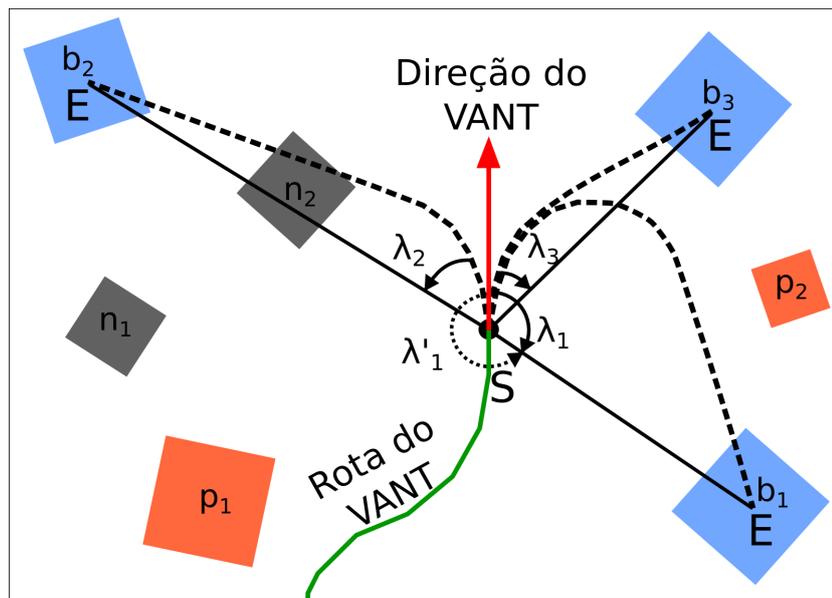


Figura 4.3: Rotas candidatas usando Heurística Gulosa.

A melhor trajetória definida entre todas as regiões bonificadoras é retornada pela HG. As soluções candidatas são avaliadas de acordo com o seu desempenho na Equação 4.2. Na Figura 4.3, a melhor solução é a que pousa na região b_3 , que é a região mais próxima do VANT. A solução que pousa em b_1 tem o desempenho um pouco pior e a solução que pousa em b_2 viola a restrição de não navegabilidade, por isso tem baixa qualidade.

Como foi mencionado, a estratégia gulosa pode ser utilizada como um operador de inicialização para criar indivíduos no AGMP. Neste 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. Assim caso a rota gerada para uma determinada região viole essa condição, essa terá uma grande penalização inviabilizando a rota. Se as soluções testadas para

cada uma das regiões bonificadoras, todas tiverem uma alta penalização, a árvore de decisão indicará que outro método deverá ser executado. Assim um algoritmo mais robusto, como um AG, deverá ser executado de forma a tentar encontrar uma rota mais segura para a aeronave.

4.5 Algoritmo Genético

Um Algoritmo Genético (AG) foi desenvolvido como alternativa ao HG descrito como apresentado no Algoritmo 2 a seguir.

Algoritmo 2: Algoritmo Genético.

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

Saída: RotaPouso: *melhorRota*

```

1 início
2   criarPopulação(rotas);
3   inicializar(rotas);
4   avaliar(rotas);
5   enquanto nãoAtingir(critérioDeParada) faça
6     enquanto nãoConvergir() faça
7       para  $i = 1$  até  $taxaCross \times tamPop$  faça
8         selecionar(pais);
9         filho ← crossover(pais);
10        mutação(filho);
11        adicionar(filho);
12      fim para
13    fim enqto
14    reiniciar(rotas);
15  fim enqto
16  RotaPouso melhorRota ← obterMelhorRota(rotas);
17  retorna melhorRota;
18 fim

```

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 de *crossover* e mutação a cada repetição. A convergência ocorre quando nenhum novo indivíduo é inserido na população. Caso tenha convergido, a população será reiniciada onde é mantido apenas o melhor indivíduo (melhor solução). Esse processo é repetido até o critério de parada ser atingido.

4.6 Algoritmo Genético Multi-Populacional

Uma outra estratégia evolutiva utilizada nesse trabalho é o AGMP. O mesmo é similar ao AG descrito na seção anterior, entretanto, o mesmo trabalha com multi-populações, migrações entre as populações e indivíduos organizados em forma de árvore. O Algoritmo 3 descreve o AGMP proposto nesse trabalho.

Algoritmo 3: Algoritmo Genético Multi-Populacional.

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

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));
7       fim para
8     estruturar(pop(i));
9     repita
10      para  $j = 1$  até  $taxaCross \times numIndivíduos$  faça
11        selecionar(pais);
12        filho  $\leftarrow$  crossover(pais);
13        mutação(filho);
14        avaliar(filho);
15        adicionar(filho, pop(i));
16      fim para
17      estruturar(pop(i));
18    até convergir(pop(i));
19  fim para
20  para  $i = 1$  até numPopulações faça
21    migrar(pop(i));
22  fim para
23  até atingir(critérioDeParada);
24  RotaPouso melhorRota  $\leftarrow$  obterMelhorRota(pop);
25  retorna melhorRota;
26 fim

```

A população primeiramente é inicializada e todos os indivíduos são avaliados (conforme Expressão 4.2). Em seguida, esses indivíduos são estruturados na árvore como ilustrado pela Figura 4.4. A hierarquia é representada pela posição dos indivíduos (nós) nos *clusters* da árvore. Há um líder em cada *cluster*, que é o melhor indivíduo do *cluster*, e seus seguidores com menor

valor de *fitness*. O melhor indivíduo é representado pelo nó raiz e as piores indivíduos são os nós folha.

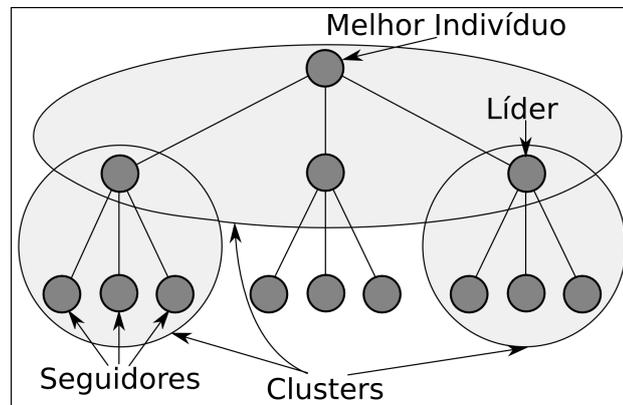


Figura 4.4: Estrutura hierárquica em árvore do AGMP.

O processo de evolução seleciona aleatoriamente dois indivíduos para reprodução, que são sempre um nó líder e um dos seus seguidores, respectivamente. O novo indivíduo gerado substitui o pior pai, se ele tem 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 $taxaCross \times numIndividuos$ tentativas. Quando todas as populações convergem, um operador de migração é executado e elas são reinicializadas, exceto pelo melhor indivíduo de cada população. O operador migração envia uma cópia do melhor indivíduo da população i para a população $i + 1$. 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.

4.7 Método Exato

Um método exato para resolver este problema ainda será estudado. O primeiro passo para isso é avançar na formulação matemática na tentativa de definir um modelo de Programação Inteira Mista (PIM) para o problema. Nesse caso, um pacote computacional como o CPLEX poderá ser utilizado na resolução de instâncias do problema. Todavia, a formulação final também poderá ser não-linear o que levaria a outras possibilidades de resolução via modelagem matemática.

4.8 Arquitetura Proposta

Esta seção mostra a arquitetura proposta para o algoritmo planejador de rota de pouso com todas as suas entradas e saídas. O mesmo busca encontrar rotas seguras para pouso e utiliza como entrada algumas informações fornecidas pelo IFA. Isso resulta em uma saída contendo um conjunto de *waypoints*, local do pouso e *fitness*. A Figura 4.5 apresenta essa arquitetura.

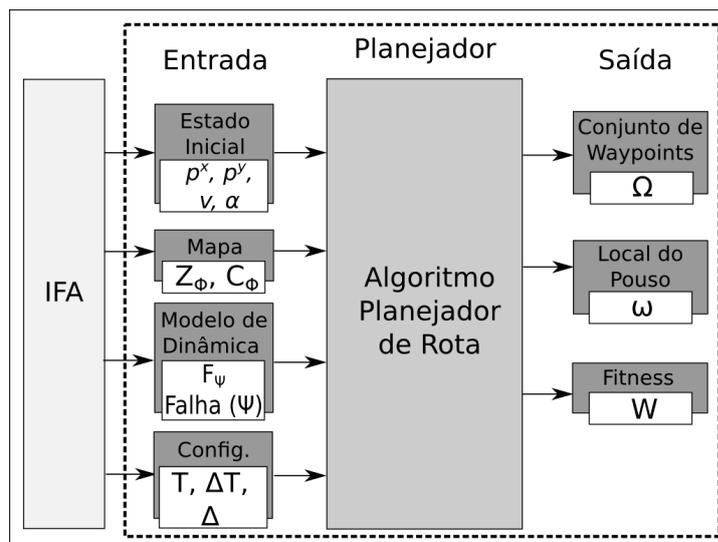


Figura 4.5: Arquitetura do planejador de rota de pouso proposto.

O sistema *In-Flight Awareness* possui acesso a uma série informações como sensores da aeronave, falhas críticas, cenário (mapa), modelo de dinâmica do VANT, etc. O algoritmo planejador é acionado no momento em que o IFA detecta a falha e usa os parâmetros de entrada descritos:

1. **Estado Inicial:** o estado inicial do VANT contém dados como posição cartesiana (p^x , p^y), velocidade (v) e ângulo (α) que serão obtidas por sensores GPS, velocímetro e giroscópio 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 mapeadas.
3. **Modelo de Dinâmica:** o modelo de dinâmica do VANT será obtida pelo IFA, que descreve o tipo de falha da aeronave (Ψ) e a função de transição de estados (F_Ψ) associada a cada situação crítica.

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

O algoritmo planejador executará usando 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. **Fitness (\mathcal{W}):** o algoritmo planejador calculará um custo associado a queda da aeronave (*fitness*) para a situação crítica ocorrida, representado por \mathcal{W} .

4.9 Conclusão

O presente capítulo apresentou a codificação do problema, definiu a função objetivo e os operadores implementados. A árvore de decisão utilizada para selecionar os métodos que serão executados foi descrita. Os métodos proposto para o planejamento da rota HG, AG e AGMP são mostrados com seu respectivo pseudocódigo. Por fim, foi apresentada a arquitetura do sistema que está em desenvolvimento com suas entradas e saídas.

Resultados Preliminares

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

Mahatma Gandhi

5.1 Introdução

Este capítulo apresenta os resultados preliminares obtidos até o presente momento. A próxima seção descreve o gerador automático de cenários desenvolvido assim como a classificação dos mapas gerados. Posteriormente, mostra-se os modelos de dinâmica para Veículos Aéreos Não Tripulados (VANTs). Por fim, são mostrados os experimentos efetuados para a Heurística Gulosa (HG), Algoritmo Genético (AG) e Algoritmo Genético Multi-Populacional (AGMP). Por enquanto, apenas experimentos *offline* foram executados.

5.2 Geração Automática de Mapas

O conjunto de mapas (cenários) a serem utilizados nas simulações e a forma como os mesmos são gerados é uma etapa importante neste trabalho. Esses cenários serão usados para

efetuar a validação do planejador de rotas de pouso. O gerador de mapas automáticos implementado será discutido nesta seção. Inicialmente, buscando atingir uma maior compreensão de como os cenários foram gerados, algumas categorias de mapas possíveis foram definidas. Nesta etapa inicial, não foi considerada a topologia do terreno com relevos 3D, ou seja, considera-se apenas cenários com topologias planas 2D.

5.2.1 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, onde três níveis de dificuldade foram definidos.

Nível de Dificuldade dos Mapas:

1. *Mapa Fácil*: Esta categoria de mapa é considerada a mais fácil. Neste tipo de cenário, pretende-se colocar um nível maior de regiões do tipo bonificadora (ϕ_b), um nível mediano do tipo penalizadora (ϕ_p) e poucas não navegáveis (ϕ_n).
2. *Mapa Normal*: Nesta categoria de mapa a dificuldade é considerada normal (mediana), ou seja, há um equilíbrio na proporção das regiões do tipo ϕ_n , ϕ_p e ϕ_b .
3. *Mapa Difícil*: Há um nível maior de regiões do tipo ϕ_n , um nível mediano do tipo ϕ_p e poucas ϕ_b .

Neste trabalho, quando se afirma que uma mapa possui um nível maior de regiões de determinado tipo significa dizer que a mesma possui maior área de cobertura sobre o território cadastrado.

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.

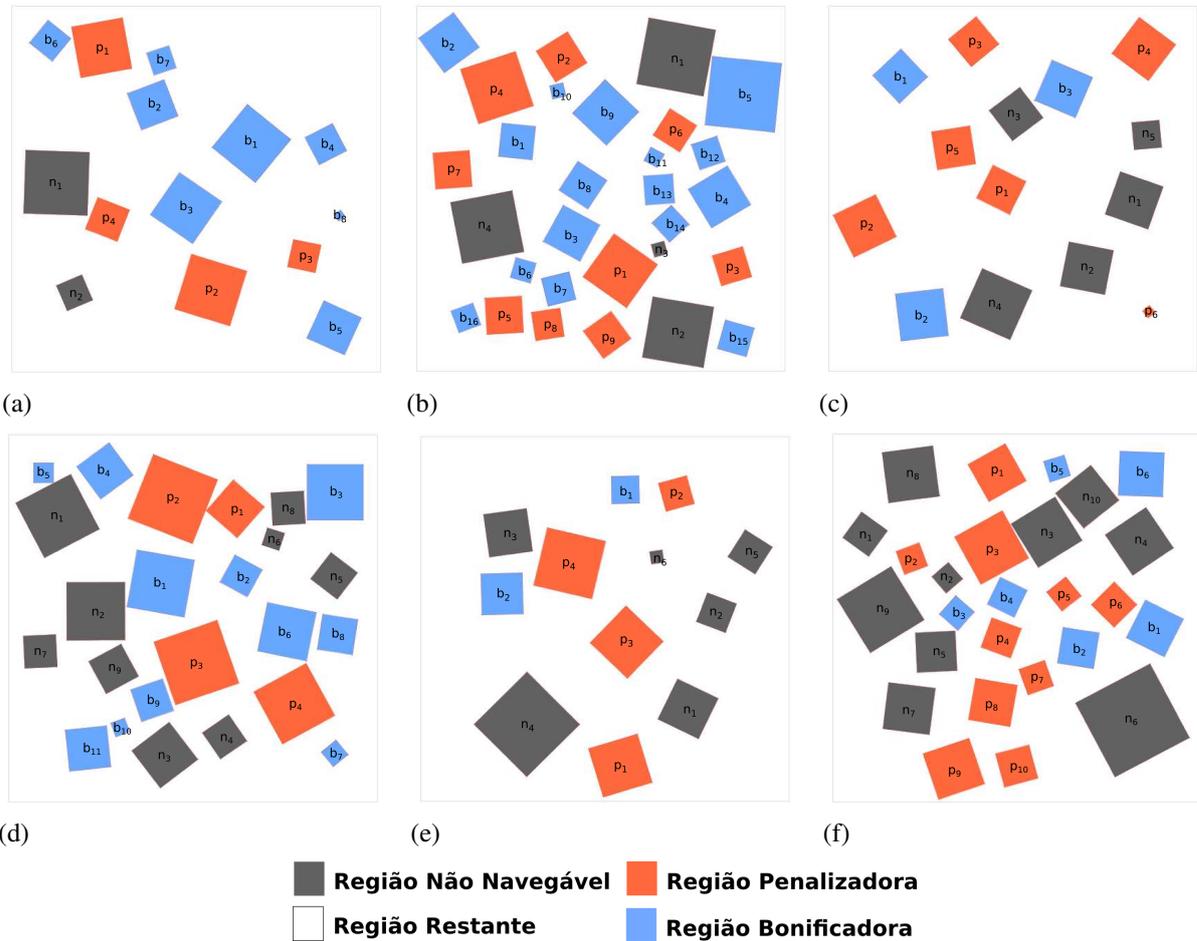


Figura 5.1: Mapas gerados automaticamente: Em (a), (c) e (e) a área de cobertura é de 25%. Em (b), (d) e (f) a área de cobertura é de 50%. Os mapas (a) e (b) são fáceis. Os mapas (c) e (d) são normais. Os mapas (e) e (f) são difíceis.

A Figura 5.1 apresenta um total de seis tipos de mapas gerados de acordo com as combinações das classificações descritas anteriormente.

5.2.2 Algoritmo de Geração de Mapas

O algoritmo para geração automática de mapas que foi implementado será descrito nesta seção. O Algoritmo 4 mostra, de forma geral, a geração dos mapas aleatórios, onde os mesmos são criados sem interseção de regiões diferentes.

Algoritmo 4: Algoritmo Gerador Automático de Mapas.**Entrada:** Numérico: *taxaOcupaçãoÁrea*, *dimXMap*, *dimYMap***Saída:** Mapa: *mapa*

```

1 início
2   listaPolígono  $\leftarrow \emptyset$ ;
3   ocupação  $\leftarrow 0$ ;
4   enquanto (ocupação < taxaOcupaçãoÁrea) faça
5     repita
6       posX  $\leftarrow$  posXRegiãoAleatório();
7       posY  $\leftarrow$  posYRegiãoAleatório();
8       comprimento  $\leftarrow$  comprimentoRegiãoAleatório();
9       ângulo  $\leftarrow$  ânguloRegiãoAleatório();
10      polígono  $\leftarrow$  criarPolígono(posX, posY, comprimento, ângulo);
11      até (nãoTerIntercessão(listaPolígono, polígono));
12      listaPolígono.adiciona(polígono);
13      ocupação  $\leftarrow$  ocupacao + comprimento2 / (dimXMap  $\times$  dimYMap);
14    fim enqto
15    mapa  $\leftarrow$  criarMapa(listaPolígono);
16    retorna mapa;
17 fim

```

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. Inicialmente, é criada uma lista de polígonos que representará o cenário. 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 cobertura desejada seja atingida. Dentro desta 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.

5.2.3 Configurações Usadas na Geração dos Mapas

As especificações das configurações dos mapas utilizados em cada uma das categorias propostos é dada na Tabela 5.1. Essas configurações foram utilizadas para gerar um conjunto de mapas aleatórios que serão utilizados nos experimentos para validar o algoritmo planejador. Observando esta 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 um total de 600 mapas.

Tabela 5.1: Configurações utilizadas no 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%

Neste trabalho foi assumido 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 aeronave terá um grau maior de liberdade e poderá ir para cada uma das direções possíveis. A Tabela 5.2 apresenta a notação para descrição das instâncias utilizadas nos experimentos.

Tabela 5.2: Notação utilizada nas instâncias.

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

5.3 Modelagem da Dinâmica do VANT

Os modelos de dinâmica desenvolvidos consideram um VANT de asa fixa com sistema de decolagem e aterrissagem convencional sobre pistas (CTOL). A princípio, definiu-se os estados (x_t) do VANT dado por:

$$x_t = [p_t^x, p_t^y, v_t, \alpha_t]^T \quad (5.1)$$

onde, p_t^x e p_t^y representam as posições (x, y) da aeronave no instante t , v_t é a velocidade da mesma no instante t e, por fim, α_t representa o ângulo de orientação do VANT em relação ao sistema de coordenadas no instante t . Os controles dados ao veículo aéreo são representados por:

$$u_t = [a_t, \varepsilon_t]^T \quad (5.2)$$

onde, a_t representa a aceleração do VANT no instante t e ε_t indica a variação angular sofrida pela aeronave, rotacionando a mesma em torno do eixo de coordenadas z . A expressão geral que representa as mudanças de estados do VANT é dada por:

$$x_{t+1} = F_{\Psi}(x_t, u_t) \Leftrightarrow \begin{bmatrix} p_{t+1}^x \\ p_{t+1}^y \\ v_{t+1} \\ \alpha_{t+1} \end{bmatrix} = \begin{bmatrix} p_t^x + v_t \cdot \cos(\alpha_t) \cdot \Delta T + a_t \cdot \cos(\alpha_t) \cdot (\Delta T)^2 / 2 \\ p_t^y + v_t \cdot \sin(\alpha_t) \cdot \Delta T + a_t \cdot \sin(\alpha_t) \cdot (\Delta T)^2 / 2 \\ v_t + a_t \cdot \Delta T - \frac{F_t^d}{m} \cdot \Delta T \\ \alpha_t + \varepsilon_t \cdot \Delta T \end{bmatrix} \quad (5.3)$$

Este sistema não linear $F_{\Psi}(x_t, u_t)$ define as transições de estados. O valor do parâmetro ΔT representa a discretização do tempo. Assim, 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, representada na variável F^d (veja a Equação 5.4), 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 5.4 que possui como parâmetros: C_d (coeficiente de arrasto que é específico para cada aeronave), ρ (massa específica do fluido), A (área de referência) (Stuart e Dahmen, 2006).

$$F_t^d = \frac{1}{2} \cdot C_d \cdot \rho \cdot A \cdot v_t^2 \quad (5.4)$$

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

$$x_{t+1} = F_{\psi_m}(x_t, u_t) \Leftrightarrow \begin{bmatrix} p_{t+1}^x \\ p_{t+1}^y \\ v_{t+1} \\ \alpha_{t+1} \end{bmatrix} = \begin{bmatrix} p_t^x + v_t \cdot \cos(\alpha_t) \cdot \Delta T \\ p_t^y + v_t \cdot \sin(\alpha_t) \cdot \Delta T \\ v_t - \frac{F_t^d}{m} \cdot \Delta T \\ \alpha_t + \varepsilon_t \cdot \Delta T \end{bmatrix} \quad (5.5)$$

$$x_{t+1} = F_{\psi_{s1}}(x_t, u_t) \Leftrightarrow \begin{bmatrix} p_{t+1}^x \\ p_{t+1}^y \\ v_{t+1} \\ \alpha_{t+1} \end{bmatrix} = \begin{bmatrix} p_t^x + v_t \cdot \cos(\alpha_t) \cdot \Delta T + a_t \cdot \cos(\alpha_t) \cdot (\Delta T)^2/2 \\ p_t^y + v_t \cdot \sin(\alpha_t) \cdot \Delta T + a_t \cdot \sin(\alpha_t) \cdot (\Delta T)^2/2 \\ v_t + a_t \cdot \Delta T - \frac{F_t^d}{m} \cdot \Delta T \\ \alpha_t + \max\{0, \varepsilon_t\} \cdot \Delta T \end{bmatrix} \quad (5.6)$$

$$x_{t+1} = F_{\psi_{s2}}(x_t, u_t) \Leftrightarrow \begin{bmatrix} p_{t+1}^x \\ p_{t+1}^y \\ v_{t+1} \\ \alpha_{t+1} \end{bmatrix} = \begin{bmatrix} p_t^x + v_t \cdot \cos(\alpha_t) \cdot \Delta T + a_t \cdot \cos(\alpha_t) \cdot (\Delta T)^2/2 \\ p_t^y + v_t \cdot \sin(\alpha_t) \cdot \Delta T + a_t \cdot \sin(\alpha_t) \cdot (\Delta T)^2/2 \\ v_t + a_t \cdot \Delta T - \frac{F_t^d}{m} \cdot \Delta T \\ \alpha_t + \min\{0, \varepsilon_t\} \cdot \Delta T \end{bmatrix} \quad (5.7)$$

A primeira modelagem apresentada F_{Ψ} é equivalente a aeronave voando sem falha associada $F_{\psi_{\emptyset}}$. O modelo que representa o problema no motor (F_{ψ_m}) 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 ($F_{\psi_{s1}}$), o VANT não pode efetuar manobra à esquerda. No problema da superfície aerodinâmica 2 ($F_{\psi_{s2}}$), o veículo aéreo não pode efetuar manobra à direita. 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.

5.3.1 Dedução da Constante de Arrasto

A modelagem da dinâmica de voo do VANT depende de uma série de fatores, onde um deles é a resistência do ar. Algumas deduções matemáticas foram realizadas para determinar um valor adequado para a constante de arrasto. A Equação 5.4 possui alguns parâmetros que são constantes como: coeficiente de arrasto (C_d), massa específica do fluido (ρ) e a área de referência (A).

Seja K_d , uma constante de arrasto associada a cada aeronave, onde $K_d = C_d \cdot \rho \cdot A/2$. Assim a equação de arrasto pode ser reescrita como $F_t^d = K_d \cdot v_t^2$. 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 A e C_d associado ao VANT. Esta seção visa encontrar o valor de K_d que se ajuste ao VANT Tiriba usado neste trabalho.

A Figura 5.2 mostra uma aeronave e as forças envolvidas na mesma. Para uma aeronave em voo de cruzeiro, as forças $T = R$ e $S = P$, ou seja, o VANT está em equilíbrio.

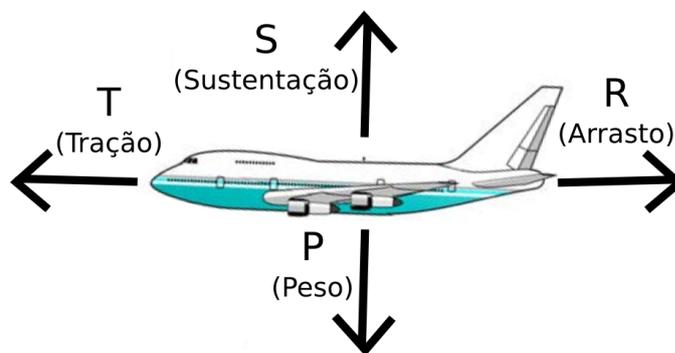


Figura 5.2: Análise das forças envolvidas na aeronave. Fonte: Adaptado de (Stuart e Dahmen, 2006).

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 os dados do Tiriba temos:

$$T_{max} = R_{max} \quad (5.8)$$

$$m \cdot a_{max} = K_d \cdot v_{term}^2 \quad (5.9)$$

$$K_d = \frac{m \cdot a_{max}}{v_{term}^2} \quad (5.10)$$

$$K_d = \frac{3,7 \cdot 2,0}{30,55^2} = 0.007925 \quad (5.11)$$

Na Expressão 5.11 é mostrado 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 2.1. A Figura 5.3 apresenta um gráfico ilustrando a modelagem feita para o Tiriba. Observa-se sua mudança de velocidade, partindo do repouso e com aceleração máxima. Após o instante 31, a velocidade da aeronave satura, ou seja, atingiu a velocidade terminal. Esse valor da contante $K_d = 0.007925$ será utilizado em todos os experimentos da dissertação.

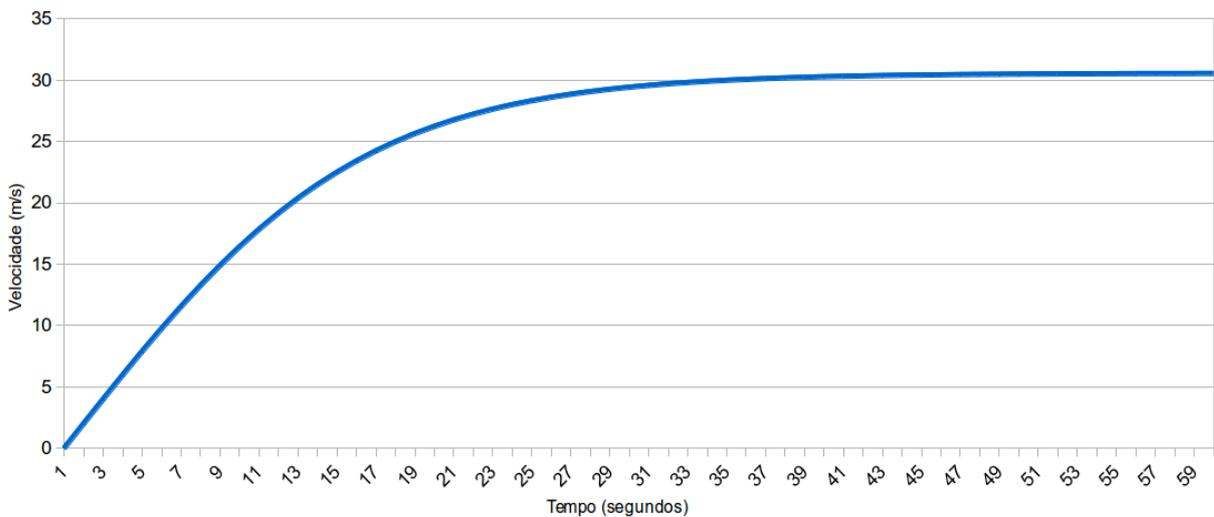


Figura 5.3: Mudança de velocidade da aeronave partindo do repouso e com aceleração constante.

5.4 Experimentos

A seguir são apresentados resultados preliminares obtidos pelos métodos inicialmente propostos.

5.4.1 Configurações Utilizadas

Uma série de experimentos foram feitos para analisar e decidir os melhores parâmetros, configurações e comportamento dos métodos. As configurações gerais do VANT e pesos das penalizações utilizadas nas simulações foram definidas conforme a Tabela 5.3.

Tabela 5.3: Configurações do Mapa, VANT e Pesos das Regiões usados nos experimentos.

Modelo	Parâmetro	Valor
Mapa	Dimensões X [m]	1000
	Dimensões Y [m]	1000
VANT	Posição inicial (p_0^x, p_0^y) [m]	(0; 0)
	Velocidade inicial (v_0) [m/s]	24
	Ângulo inicial (α_0) [°]	90
	Velocidade ($v_{min}; v_{max}$) [m/s]	[11, 1; 30, 5]
	Velocidade angular ($\varepsilon_{min}; \varepsilon_{max}$) [°/s]	[-3; 3]
	Aceleração ($a_{min}; a_{max}$) [m/s ²]	[0, 0; 2, 0]
	Tempo máximo para queda (T) [s]	60
	Discretização do tempo (ΔT) [s]	1
	Probabilidade de violar a região ϕ_n (Δ)	0,001
Pesos das Regiões	C_{ϕ_b}	2000
	C_{ϕ_p}	8000
	C_{ϕ_n}	100000
	C_{ϕ_r}	0

As configurações utilizadas no VANT foram baseadas em dados relativo ao Tiriba. Observando esta tabela, tem-se que a velocidade inicial no momento da falha é de $v_0 = 24m/s$, o ângulo da aeronave é de $\alpha_0 = 90^\circ$, 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 8000 quando associada a queda da aeronave na região do tipo penalizadora. Vale ressaltar que o peso C_{ϕ_b} de 2000 não indica penalização nas equações, mas sim um bônus. Os experimentos foram executados no computador com as especificações da Tabela 5.4.

Tabela 5.4: 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
Velocidade de Clock	1.80 GHz
Memória RAM	4 GB

5.4.2 Análise de Sensibilidade do AG e AGMP

Esta seção apresenta uma análise de sensibilidade dos parâmetros do AG e do AGMP. A Tabela 5.5 contém o conjunto de parâmetros analisados pelo algoritmo. Por exemplo, o número de populações testados 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 todos ativados, onde o método aplica um deles ao indivíduo via seleção aleatória. Essa tabela apresenta também alguns valores considerados bons pela 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 experimentos. Assim, quando se encontrou os melhores resultados para (P^*), este valor foi utilizado e então prosseguiu-se na ordem estabelecida na tabela a fim de se encontrar os próximos valores.

As condições em que foram submetidos os experimentos são descritas: selecionou-se arbitrariamente um mapa de cada conjunto de instância ($I1, I2, \dots, I6$), totalizando seis mapas, um para cada categoria. Um total de 50 execuções foi realizado para cada mapa; o algoritmo para após executar 10000 avaliações; o VANT efetuou o pouso sem falha associada, assim descobriu-se os melhores parâmetros independentemente de situação crítica.

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, devido a considerável complexidade do problema tratado, indicando que o método se comporta de forma parecida independente de parâmetros. Analisando a Tabela 5.5 na coluna P^* e a linha do N° de Ramificação nota-se

Tabela 5.5: Análise de sensibilidade do AG e AGMP.

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	{ <u>1</u> ; 2; 3; 4; 5}	3	3^L
	3	Nº de Indivíduos	{6; 13; <u>20</u> ; 27; 34}	13	13^L
	4	Taxa de Crossover	{0, 5; 1, 0; <u>1, 5</u> ; 2, 0; 2, 5}	1,5	0, 5
	5	Taxa de Mutação	{0, 0; 0, 25; 0, 5; <u>0, 75</u> ; 1, 0}	0,75	$0, 75^L$
	6	Operad. de Crossover	{ <i>Aritmético</i> ; <i>Média</i> ; <i>Geométrico</i> ; <i>OX</i> ; <i>BLX-α</i> ; <i>Todos</i> }	-	<i>Todos</i>
	7	Operad. de Mutação	{ <i>Uniforme</i> ; <i>Limite</i> ; <i>Creep</i> ; <i>Todos</i> }	-	<i>Todos</i>
	8	Operad. de Inicialização	{ <i>Aleatório</i> ; <i>CurvaCurta</i> ; <i>Aceler.Curta</i> ; <i>Gulosa</i> ; <i>Todos</i> }	-	<i>Todos</i>
AG	9	Nº de Indivíduos	{13; 39}	-	39
	10	Tamanho do Torneio	{2; 3; 4; 5}	-	3

a escolha de 3^L . Esse tipo de resultado indica que houve empate técnico entre os diferentes parâmetros, entretanto, usou-se como desempate o valor usado na literatura 3^L .

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. Entretanto, 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 decidido escolher o número de indivíduos igual a 39 e tamanho do torneio de 3. Todos os resultados da análise sensibilidade de parâmetro efetuada encontra-se disponível no site¹.

As seções seguintes apresentarão os resultados obtidos para cada um dos métodos implementados. Algumas considerações a serem feitas antes de tais resultados dos experimentos são importante. A seguir quando se falar que uma solução (rota) for factível, significará que a mesma não violou nenhuma das três restrições de Pousos (ϕ_b), de Velocidade (v_{min}) e Risco (Δ). Restrições de Pousos quer dizer que o VANT encontrou uma região bonificadora e conseguiu pousar na mesma; restrições de Velocidade quer dizer que a aeronave não conseguiu pousar no último instante, pois estava acima da velocidade mínima; restrições de Risco indica que o VANT violou o risco, ou seja, invadiu o espaço aéreo não navegável (ϕ_n).

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

5.4.3 Exemplo de Soluções de Situações Críticas

Essa seção apresenta alguns exemplos de soluções obtidas para diferentes situações críticas em um mesmo cenário. 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 objetivo é apenas ilustrar as rotas seguidas pela aeronave, assim um algoritmo mais robusto como o AGMP foi utilizado. A Tabela 5.6 apresenta os resultados obtidos para um mapa do conjunto de instâncias. A primeira linha dessa tabela mostra algumas informações sobre o pouso dado o problema no motor (ψ_m), 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 na região bonificadora (ϕ_b).

Tabela 5.6: 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_{s^1}	37,80	60	ϕ_b
ψ_{s^2}	28,10	48	ϕ_b
ψ_{\emptyset}	13,40	34	ϕ_b

A Figura 5.4 mostra as trajetórias seguidas nos pousos do VANT para cada uma das situações críticas estabelecidas neste trabalho. Na Figura 5.4 (a) percebe-se que a aeronave pousou bem rapidamente devido ao problema no motor e alcançou uma região bonificadora. Já na Figura 5.4 (b) o veículo aéreo com problema na bateria decidiu por percorrer uma distância maior até chegar na região bonificadora. Nesse caso, seu *fitness* foi penalizado em relação ao problema no motor por percorrer maior distância com uma falha na bateria. Na Figura 5.4 (c) é mostrado que 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 5.4 (d) mostra que a aeronave é capaz de girar somente para a direita e conseguir encontrar uma região de pouso bonificadora. Devido a restrição de manobra, o VANT teve que efetuar um laço para pousar.

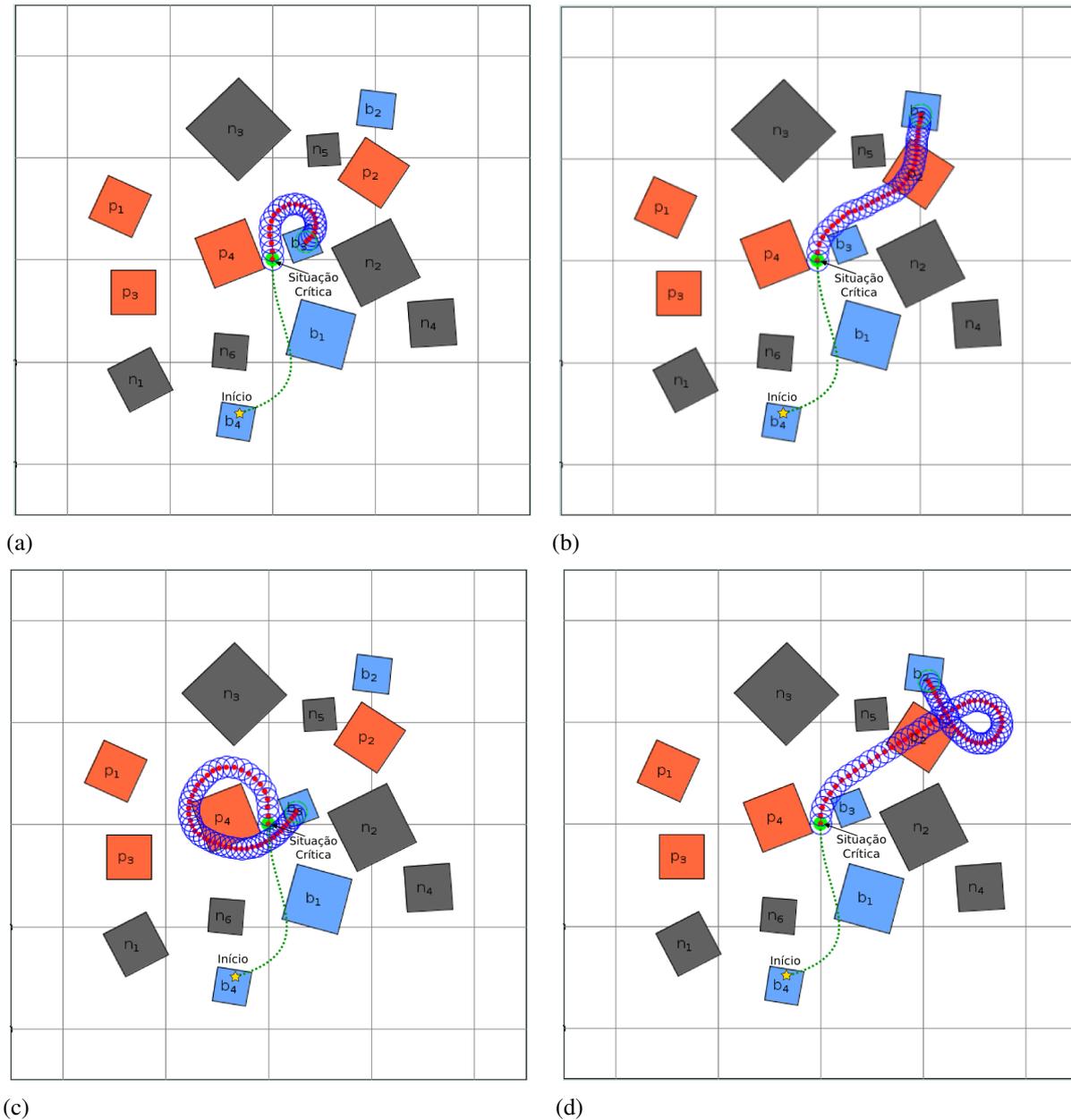


Figura 5.4: Exemplo de rotas dada a variação das situações críticas: Em (a), (b), (c) e (d) as situações são ψ_m , ψ_b , ψ_{s1} e ψ_{s2} respectivamente.

5.4.4 Heurística Gulosa

Essa seção, apresenta os resultados obtidos pela Heurística Gulosa (HG) que podem ser vistos na Tabela 5.7. As condições iniciais que foram submetidos os experimentos são: todos os mapas criados (I_1, I_2, \dots, I_6) foram avaliados pela HG uma vez; o critério de parada do algoritmo ocorre após 10000 avaliações; todas as situações críticas modeladas foram associadas ao VANT, conforme apresentado na tabela.

Tabela 5.7: Resultado obtido para a HG, para todas as falhas 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	0,025	67	5	28
	I4	0,065	83	0	17
	I5	0,037	93	3	4
	I6	0,085	98	0	2
	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	

Essa Tabela 5.7 mostra o tempo médio (dado em segundos) para cada tipo de instância. As colunas indicadas por *Pousou ϕ_b* e *Pousou ϕ_r* indicam a quantidade de soluções que pousaram em regiões bonificadoras e restantes com segurança, respectivamente. A coluna *Infactível* indica a quantidade de soluções que violaram ou a condição de não navegabilidade, pousaram em regiões penalizadoras, ou ainda, violou a condição de velocidade mínima para o pouso. Como para cada tipo de instância tem-se 100 mapas o somatório das colunas *Pousou ϕ_b* , *Pousou ϕ_r* e *Infactível* totaliza 100. Por exemplo, para a situação crítica ψ_\emptyset a instância I1 obteve o melhor resultado com 99% de soluções que levam a rotas seguras de voo e pousaram na região bonificadora. Nessa mesma situação, a instância I3 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 segun-

dos, em geral, sendo aproximadamente 0,071 segundos. O baixo custo computacional justifica executar a HG antes de qualquer outra técnica mais complexa que eventualmente levará um tempo maior. Ressalta-se também que a taxa de sucesso é de 77,83%, considerando todas as situações de pouso descritas. A HG mostrou-se ruim em 13,27% dos mapas testados, assim será necessário executar outro algoritmo de planejamento de pouso, quando se aplicar o conceito de árvore de decisão.

5.4.5 Algoritmo Genético

Os resultados obtidos pelo Algoritmo Genético (AG) estão agrupados na Tabela 5.8. Duas versões do AG foram criadas uma sem a inicialização gulosa e outra com, chamadas AG1 e AG2 respectivamente.

Tabela 5.8: Resultado obtido para o AG1 e AG2, para todos as falhas 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	

Analisando os dados da Tabela 5.8, percebe-se que o AG1 teve um desempenho semelhante a HG resolvendo 77,96% dos mapas. O AG com o operador de inicialização gulosa AG2 teve um desempenho bem melhor resolvendo 89,86% das instâncias. Assim o AG2 resolveu cerca de 12,0% mais instâncias que o AG1. No quesito tempo gasto as duas versões acabaram gastando tempos semelhantes 0,955 e 0,912 segundos. Comparando o tempo gasto do AG2 com a HG, percebe-se que o tempo da heurística é 12 vezes mais rápido que o método evolutivo.

5.4.6 Algoritmo Genético Multi-Populacional

Os resultados obtidos com o Algoritmo Genético Multi-Populacional (AGMP) estão na Tabela 5.9. Este algoritmo evolutivo também teve duas versões utilizadas nos experimentos uma sem a inicialização gulosa e outra com, chamadas AGMP1 e AGMP2 respectivamente.

Tabela 5.9: Resultado obtido para o AGMP1 e AGMP2, para todas as falhas 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	

Ao observar a Tabela 5.9, nota-se que a taxa de mapas solucionados foi de 89,53% com a versão AGMP2 contra 80,13% do AGMP1. O tempo gasto no AGMP2 (0,874 segundos) também foi ligeiramente menor do que o AGMP1 (1,017 segundos).

Uma análise visando verificar a existência ou inexistência de mapas inactíveis revelou que apenas oito mapas do total 600 não foram resolvidos por nenhum dos três algoritmos (HG, AG e AGMP). É importante lembrar que isso não prova que esses oito mapas não tenham solução, porém tem-se a certeza de que a maior parte (592 mapas) é factível. Assim, ao se combinar os três métodos anteriores poderão ser resolvidos pelo menos 98,66% dos mapas gerados.

5.5 Conclusão

Este capítulo apresentou os resultados preliminares obtidos até o presente momento. Uma etapa importante foi a definição do conjunto de instâncias, o algoritmo gerador de mapas, e os resultados dos mapas gerados.

A modelagem da dinâmica da aeronave é um importante ponto de contribuição deste trabalho, o qual busca desenvolver uma dinâmica para cada situação crítica ocorrida. Por fim, foram apresentados alguns resultados utilizando a HG, AG1, AG2, AGMP1 e AGMP2. A HG resolveu cerca de 77,83% das instâncias, o AG1 e AG2 resolveram respectivamente cerca de 77,96% e 89,86% de todos os 600 cenários considerados, por fim o AGMP1 e AGMP2 resolveram 80,13% e 89,53% das instâncias. Portanto, houve um desempenho equivalente entre o AG2 e AGMP2, mostrando que ambos os métodos são adequados para solucionar o problema quando a inicialização gulosa esta ativada.

Cronograma

“Toda e qualquer conquista de vida requer planejamento e estratégia.”

Thaianne Venâncio de Farias

As atividades definidas e o cronograma previsto para a dissertação são apresentados a seguir.

- A) Disciplinas:** Cumprimento do número mínimo de créditos em disciplinas exigido pelo programa de pós-graduação do ICMC/USP.
- B) Revisão bibliográfica:** Revisão da literatura e atualização da mesma ao longo do desenvolvimento da pesquisa.
- C) Pesquisa situações críticas:** Levantamento e estudo das situações críticas envolvendo Veículos Aéreos Não Tripulados (VANTs), e a sua relação com a dinâmica da aeronave.
- D) Desenvolvimento de mapas para voo:** Elaboração de algoritmos para criar um conjunto instâncias de cenários (mapas) diferentes.
- E) Desenvolvimento de algoritmos para pouso considerando em 2D:** Elaboração de algoritmos para o pouso forçado do VANT considerando o cenário e dinâmica em 2D.
- F) Análise de sensibilidade:** Testes com os algoritmos baseados em populações.

- G) Qualificação e proficiência:** Ao final do primeiro ano o candidato deverá redigir e apresentar seu exame de qualificação e um teste de proficiência em língua inglesa.
- H) Desenvolvimento de modelo e método:** Definição de um modelo de programação matemática para obtenção de soluções ótimas ou limitantes inferiores de boa qualidade para o problema. Implementação de novos métodos baseados em algoritmos evolutivos.
- I) Simulações e comparações dos resultados:** Os algoritmos elaborados serão avaliados, procurando os melhores parâmetros. Comparações dos resultados com outras técnicas.
- J) Validação em simulador de voo:** Os algoritmos elaborados serão validados no simulador de voo FlighGear.
- K) Elaboração de artigos:** Escrita de artigos científicos para conferências e periódicos.
- L) Elaboração da dissertação:** O trabalho desenvolvido será documentado apropriadamente na dissertação de mestrado, com defesa prevista para Fevereiro de 2016.

Tabela 6.1: Cronograma de execução das atividades.

Atividades	Meses							
	01-03	04-06	07-09	10-12	13-15	16-18	19-21	22-24
A)	✓	✓	✓					
B)	✓	✓	✓	✓	●	*	●	●
C)		✓	✓	✓				
D)			✓	✓				
E)		✓	✓	✓				
F)			✓	✓				
G)				✓				
H)					●	*	●	
I)			✓	✓	●	*	●	
J)						*	●	
K)				✓	●	*	●	●
L)			✓	✓	●	*	●	●

O cronograma para execução das atividades previstas segue na Tabela 6.1. As atividades marcadas com o símbolo (✓) foram executadas e concluídas, as atividades indicadas por (*) estão previstas para serem executadas em estágio no exterior no *Computer Science and Artificial Intelligence Laboratory/Massachusetts Institute of Technology (CSAIL/MIT)*, e por fim as marcadas por (●) ainda estão para serem feitas no ICMC/USP.

Considerações Finais

*“O futuro dependerá daquilo que fazemos
no presente.”*

Mohandas Karamchand Gandhi

Este trabalho apresentou um problema de replanejamento de caminho para pousar um Veículo Aéreo Não Tripulado (VANT) sob algum tipo de falha nos equipamentos, denominado aqui como situação crítica. Assume-se que a aeronave executa uma missão dentro de um cenário em que a situação crítica acontece, de modo que algoritmo planejador de rotas é acionado e o objetivo torna-se a minimizar os danos e aumentar a segurança durante o pouso emergencial.

Foram apresentadas três estratégias para o planejador de rotas, sendo uma heurística e duas metaheurísticas evolutivas: Heurística Gulosa (HG), Algoritmo Genético (AG) e Algoritmo Genético Multi-Populacional (AGMP). Duas versões para cada método evolutivo foram criadas ativando um operador de inicialização dos indivíduos com a heurística gulosa. Um conjunto de 600 mapas foram gerados com o objetivo de definir alguns níveis de dificuldade e avaliar os métodos propostos. A modelagem da dinâmica do VANT diante algumas situações críticas também foi efetuada.

Uma estratégia chamada *In-Flight Awareness* será utilizada neste trabalho. O IFA fornece ao algoritmo planejador informações importante sobre as condições da situação crítica encon-

trada, os estados da aeronave no momento da falha e o cenário de voo. Estas informações são utilizadas pelo método que planeja a trajetória para encontrar uma rota de pouso emergencial.

Os resultados indicam que todas as abordagens são capazes de pousar a aeronave para mais de 67% de mapas, em média, em todas as situações críticas. O AG2 e AGMP2 são os métodos mais promissores, uma vez que aproveitou para evoluir indivíduos inicializados pela HG, sendo capaz de pousar o VANT em regiões bonificadoras para mais de 78% e 79% de mapas respectivamente. Os métodos foram melhor lidando com o pouso sem falhas seguida com o problema de bateria. A HG retorna soluções rápidas, mas pode falhar mais do que as outras abordagens de pouso em regiões bonificadoras. As abordagens baseadas em soluções dos métodos evolutivos levaram cerca de 1 segundo, enquanto o HG gastou cerca de 0,1 segundo. Para resumir, os métodos propostos são capazes de retornar soluções de boa qualidade dentro de um curto tempo computacional, o que é relevante para tal problema.

Como trabalhos futuros, o modelo matemático será melhorado com o objetivo de descrever este problema como um modelo de Programação Linear Inteira Mista (PLIM) e resolvê-lo usando métodos exatos. Os métodos propostos serão avaliados, em seguida, usando um simulador de voo. Finalmente, outras situações críticas poderão ser incorporadas ao problema.

Referências Bibliográficas

AGX. **AGX Tecnologia**. AGX. Disponível em: <http://www.agx.com.br/n2/pages/index.php?opt=airplane_history>. Acesso em: 23 set. 2014.

ANAC, **Instrução Suplementar - IS: IS nº21-002 Revisão A**, Agência Nacional de Aviação Civil, 2012. 21 p.

ANAC, **Proposta de Instrução Suplementar, Intitulada “Emissão de Certificado de Autorização de Voo Experimental para Sistemas de Veículo Aéreo Não Tripulado”**, Agência Nacional de Aviação Civil - Superintendência de Aeronavegabilidade, 2012. 5 p.

BANKS, J. Discrete Event Simulation, **Winter Simulation Conference**, 2000.

BESADA-PORTAS, E.; DE LA TORRE, 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, 2013.

BLACKMORE, L.; ONO, M.; WILLIAMS, B. C. Chance-Constrained Optimal Path Planning With Obstacles, **IEEE Press**, v. 27, p. 1080-1094, 2011. ISBN 1552-3098.

BRAGA, R. T. V.; BRANCO, K. R. L. J. C.; TRINDADE JÚNIOR, O.; GIMENES, I. M. S. Evolving tiriba design towards a product line of small electric-powered UAVs, **Brazilian Conference on Critical Embedded Systems**, p. 67-72, 2011.

- BRANCO, K. R. L. J. C.; PELIZZONI, J. M.; NERIS, L.; TRINDADE JÚNIOR, O.; OSÓRIO, F. S.; WOLF, D. F. Tiriba - a new approach of UAV based on model driven development and multiprocessors, **IEEE International Conference on Robotics and Automation**, Shanghai, p. 9-13, 2011.
- BURIOL, L.; FRANÇA, P. M.; MOSCATO, P. A New Memetic Algorithm for the Asymmetric Traveling Salesman Problem, **Journal of Heuristics**, v. 10, p. 483-506, 2004.
- CENIPA, **FCA 58-1: Panorama Estatístico da Aviação Civil Brasileira**, Centro de Investigação e Prevenção de Acidentes Aeronáuticos, Brasília, DF, Brasil, 2012. 81 p.
- DRONEWARS. **Drone Crash Database - Drone Wars UK**. Dronewars. Disponível em: <<http://dronewars.net/drone-crash-database/>>. Acesso em: 25 set. 2014.
- DRURY, J. L.; RIEK, L.; RACKLIFFE, N. A Decomposition of UAV-Related Situation Awareness, **Proceedings of the 1st ACM SIGCHI/SIGART conference on Human-robot interaction**, Salt Lake City, Utah, 2006.
- EIBEN, A. E.; SMITH, J. E. **Introduction to Evolutionary Computing**, Edition 1, Springer, 2003, ISBN 3-540-40184-9.
- ENDSLEY, M. R. Design and evaluation for situation awareness enhancement, **Proceedings of the Human Factors Society 32nd Annual Meeting**, Santa Monica, CA, 1988.
- ENDSLEY, M. R.; JONES, G. D. **Designing for Situation Awareness: An Approach to User-Centered Design**, Edition 2, CRC Press, 2006. 396 p. ISBN 978-1420063554.
- ESHELMAN, L. J.; SCHAFFER, J. D. Real-coded genetic algorithms and interval-schema, **Foundations of Genetic Algorithms 2**, **Morgan Kaufmann Publishers**, p. 187-202, 1993.
- FAA, **FAA Modernization and Reform Act of 2012**, Federal Aviation Administration, Washington, DC, USA, 2012. 296 p.
- FIGUEIRA, N.; TRINDADE JÚNIOR, O.; MATTEI, A. L. P.; NERIS, L. Mission Oriented Sensor Arrays – An Approach towards UAS Usability Improvement in Practical Applications, **5th European Conference for Aeronautics and Space Sciences (EUCASS)**, Munich, 2013.

- FLIGHTGEAR. **FlightGear Wiki**. FlightGear. Disponível em: <<http://wiki.flightgear.org/>>. Acesso em: 23 out. 2014.
- FLIGHTGEAR. **FlightGear Flight Simulator**. FlightGear. Disponível em: <<http://www.flightgear.org/>>. Acesso em: 23 out. 2014.
- FLIGHTGEAR. **TerraGear - FlightGear Wiki**. FlightGear. Disponível em: <<http://wiki.flightgear.org/TerraGear>>. Acesso em: 08 dez. 2014.
- FLIGHTGEAR. **SimGear - FlightGear Wiki**. FlightGear. Disponível em: <<http://wiki.flightgear.org/SimGear>>. Acesso em: 08 dez. 2014.
- FLIGHTGEAR. **OpenGL - FlightGear Wiki**. FlightGear. Disponível em: <<http://wiki.flightgear.org/OpenGL>>. Acesso em: 08 dez. 2014.
- FONSECA, E.; MATTEI, A. L. P.; CUNHA, W. C. Adaptive integration systems using FPGA COTS devices, **5th European Conference for Aeronautics and Space Sciences (EUCASS)**, Munich, 2013.
- FRANÇA, P. M.; MENDES, A.; MOSCATO, P. A memetic algorithm for the total tardiness single machine scheduling problem, **European Journal of Operational Research**, v. 132, p. 224 - 242, 2001.
- HERRERA, F.; LOZANO, M.; SÁNCHEZ, A. M. Hybrid crossover operators for real-coded genetic algorithms: an experimental study, **Soft Comput.**, v. 9, p. 280-298, 2005.
- HOLLAND, J. H. **Adaptation in Natural and Artificial Systems**, University of Michigan Press, Ann Arbor, MI, USA, 1975.
- ICAO, **Doc 9859: Safety Management Manual (SMM)**, International Civil Aviation Organization, University Street, Montréal, Quebec, Canada, Edition 3. 2013. 251 p. ISBN 978-92-9249-214-4.
- JIN, Y. Surrogate-assisted evolutionary computation: Recent advances and future challenges, **Swarm and Evolutionary Computation**, v. 1, p. 61-70, 2011.

- JONG, K. A. **Evolutionary Computation: A Unified Approach**, Cambridge, MA, USA, MIT Press, Massachusetts Institute of Technology, 2006. 256 p. ISBN 0-262-04194-4.
- LAMINAR RESEARCH, **Plane Maker App Manual**, Laminar Research, 2013. 105 p.
- LEITE, O. **Projeto de Lei Nº 5942**”, Câmara dos Deputados, 2013. 11 p.
- LI, H. X. **Kongming: A Generative Planner for Hybrid Systems with Temporally Extended Goals**. 2010, PhD thesis - Massachusetts Institute of Technology, 2010. 237 p.
- MATTEI, A. L. P.; FONSECA, E.; FIGUEIRA, N. M.; TRINDADE JÚNIOR, O.; VAZ, F. UAV In-Flight Awareness: A Tool to Improve Safety, **5th European Conference for Aeronautics and Space Sciences (EUCASS)**, Munich, 2013.
- MATTEI, A. L. P. **Consciência Situacional em Voo de Sistemas Aéreos não Tripulados**. 2014. p. 123. Qualificação de Doutorado. Universidade de São Paulo (USP), São Carlos, SP, 2014.
- MEULEAU, N.; PLAUNT, C.; SMITH, D. E.; SMITH, T. An Emergency Landing Planner for Damaged Aircraft, **Proceedings of the Twenty-First Innovative Applications of Artificial Intelligence Conference**, 2009.
- MEULEAU, N.; NEUKOM, C.; PLAUNT, C.; SMITH, D. E.; SMITH, T. The Emergency Landing Planner Experiment, **21st International Conference on Automated Planning and Scheduling**, 2011.
- MICHALEWICZ, Z.; SCHOENAUER, M. Evolutionary Algorithms for Constrained Parameter Optimization Problems, **Evolutionary Computation**, v. 4, p. 1-32, 1996.
- MICROSOFT FLIGHT. **Microsoft Flight**. Microsoft. Disponível em: <<http://www.microsoft.com/games/flight/>>. Acesso em: 23 out. 2014.
- TAVAKKOLI-MOGHADDAM, R.; GHOLIPOUR-KANINI, 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.

- MORTIMER, G. **Schiebel S-100 crash kills engineer in South Korea**. sUAS News. 2012. Disponível em: <<http://www.suasnews.com/2012/05/15515/schiebel-s-100-crash-kills-engineer-in-south-korea>>. Acesso em: 17 set. 2014.
- MOSCATO, P.; MENDES, A.; BERRETTA, R. Benchmarking a memetic algorithm for ordering microarray data, **Biosystems**, v. 88, p. 56-75, 2007.
- NATO, **STANAG 4671: UAV Systems Airworthiness Requirements (USAR) for North Atlantic Treaty Organization (NATO) Military UAV Systems**, North Atlantic Treaty Organization, Edition 1, 2007. 205 p.
- OGLOBO. **Uma queda drone e bater um corredor, seu operador evoca pirataria**. O Globo. 2013 . Disponível em: <<http://oglobo.globo.com/mundo/eua-registram-mais-de-400-acidentes-com-drones-militares-12950003>>. Acesso em: 17 set. 2014.
- ONO, M.; WILLIAMS, B. C.; BLACKMORE, L. Probabilistic Planning for Continuous Dynamic Systems under Bounded Risk, **Journal of Artificial Intelligence Research**, v. 46, p. 511-577, 2013.
- PARAIZO, D.; FERIGATO, G. **Uso de drones se populariza no jornalismo, mas divide opiniões pelo risco de acidentes**. Portal Imprensa. 2013. Disponível em: <http://www.portalimprensa.com.br/revista_imprensa/conteudo-extra/67222/uso+de+drones+se+populariza+no+jornalismo+mas+divide+opinioes+pelo+risco+de+acidentes>. Acesso em: 17 set. 2014.
- PAULA, V. M. G. **BQM-1BR: O VANT à Jato Brasileiro**, Relatório Técnico, Universidade Federal de Juiz de Fora, 2009, 3 p.
- 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, p. 47 - 55, 2012.
- RICHARDS, A.; HOW, J. Aircraft trajectory planning with collision avoidance using mixed integer linear programming, **Proceedings of American Control Conference**, 2002.

- RODRIGUES, D.; PIRES, R. d. M.; ESTRELLA, J. C.; MARCONATO, E. A.; TRINDADE JÚNIOR, O.; BRANCO, K. R. L. J. C. Using SOA in Critical-Embedded Systems, **IEEE International Conferences on Internet of Things, and Cyber, Physical and Social Computing**, 2011.
- SEPPÄNEN, H.; MÄKELÄ, J.; LUOKKALA, P.; VIRRANTAU, K. Developing shared situational awareness for emergency management, **Safety Science**, v. 55, p. 1-9, 2013.
- SMARTDRONES. **Uma queda drone e bater um corredor, seu operador evoca pirataria**. Smartdrones. 2013 . Disponível em: <<http://www.smartdrones.fr/pt/zang%C3%A3o-%C3%A9-acidente-pode-sempre-dizer-que-ele-era-Hacke/002771>>. Acesso em: 17 set. 2014.
- SMARTDRONES. **Um jovem é morto por seu próprio robô**. Smartdrones. 2013. Disponível em: <<http://www.smartdrones.fr/pt/morto-por-seu-pr%C3%B3prio-rob%C3%B4-em-um-parque-no-Brooklyn/001266>>. Acesso em: 17 set. 2014.
- SMARTDRONES. **Suíça: um operador de robô foi morto por sua própria unidade**. Smartdrones. 2013 . Disponível em: <<http://www.smartdrones.fr/pt/Operador-su%C3%AD%C3%A7o-de-um-drone-tinha-sido-morto-por-sua-pr%C3%B3pria-unidade/00921>>. Acesso em: 17 set. 2014.
- SMARTDRONES. **Um zangão cai em multidão na Catalunha**. Smartdrones. 2013. Disponível em: <<http://www.smartdrones.fr/pt/secrase-um-drone-no-meio-da-multid%C3%A3o-na-Catalunha/001381>>. Acesso em: 17 set. 2014.
- STUDART, N.; DAHMEN, S. R.; A Física do Voo na Sala de Aula, **Física na Escola**, v. 7, n. 2, p. 1-7, 2006.
- TOLEDO, C. F. M.; LIMA, J. M. G.; ARANTES, M. S. A multi-population genetic algorithm approach for PID controller auto-tuning, **ETFA**, p. 1-8, 2012.
- TOLEDO, C. F. M.; ARANTES, M. S.; OLIVEIRA, R. R. R.; ALMADA-LOBO, B. Glass container production scheduling through hybrid multi-population based evolutionary algorithm, **Appl. Soft Comput**, 2013.

- TUNCER, A.; MEHMET, Y. Dynamic Path Planning of Mobile Robots with Improved Genetic Algorithm. **Comput. Electr. Eng.**, Pergamon Press, v. 38, p. 1564-1572, 2012.
- VARELA, G.; CAAMAÑO, P.; ORJALES, F.; DEIBE, A.; LÓPEZ-PENA, F.; DURO, R. J. Autonomous UAV based search operations using Constrained Sampling Evolutionary Algorithms, **Neurocomputing**, v. 132, p. 54-67, 2014.
- WIKIPÉDIA. **Veículo aéreo não tripulado - Wikipédia, a enciclopédia livre**. Wikipédia. 2014. Disponível em: <http://pt.wikipedia.org/wiki/Ve%C3%ADculo_a%C3%A9reo_n%C3%A3o_tripulado>. Acesso em: 23 set. 2014.
- WIKIPÉDIA. **FlightGear - Wikipédia, a enciclopédia livre**. Wikipédia. Disponível em: <<http://pt.wikipedia.org/wiki/FlightGear>>. Acesso em: 23 out. 2014.
- WIKIPÉDIA. **X-Plane - Wikipédia, a enciclopédia livre**. Wikipédia. Disponível em: <<http://pt.wikipedia.org/wiki/X-Plane>>. Acesso em: 23 out. 2014.
- WIKIPÉDIA. **Microsoft Flight - Wikipédia, a enciclopédia livre**. Wikipédia. Disponível em: <http://en.wikipedia.org/wiki/Microsoft_Flight>. Acesso em: 11 nov. 2014.
- X-PLANE. **FAA-Certified X-Plane**. X-Plane. Disponível em: <<http://www.x-plane.com/pro/certified/>>. Acesso em: 09 dez. 2014.
- X-PLANE. **X-Plane 10 Global**. X-Plane. Disponível em: <<http://www.x-plane.com/desktop/home/>>. Acesso em: 23 out. 2014.
- X-PLANE. **World Editor Manual - X-Plane**. X-Plane. Disponível em: <<http://developer.x-plane.com/manuals/wed/>>. Acesso em: 23 out. 2014.
- ZHANG, X.; DUAN, H. An improved constrained differential evolution algorithm for unmanned aerial vehicle global route planning, **Appl. Soft Comput.**, p. 270-284, 2015.