



Universidade do Estado do Amazonas
Escola Superior de Tecnologia
Engenharia Mecânica



Caio Augusto Mascarenhas Dias Filho

Otimização Bi-Objetivo do Projeto Conceitual de Aeronaves de Baixo Número de Reynolds Utilizando Algoritmos Genéticos

Manaus - Amazonas

2024

Caio Augusto Mascarenhas Dias Filho

**Otimização Bi-Objetivo do Projeto Conceitual de
Aeronaves de Baixo Número de Reynolds Utilizando
Algoritmos Genéticos**

Trabalho de Conclusão de Curso apresentado
como requisito parcial à obtenção do título
de bacharel em Engenharia Mecânica da Uni-
versidade do Estado do Amazonas (UEA).

Universidade do Estado do Amazonas - UEA

Escola Superior de Tecnologia - EST

Engenharia Mecânica

Orientador: Prof. Dr. Rubelmar Maia de Azevedo Cruz Neto

Manaus - Amazonas

2024

Ficha Catalográfica

Ficha catalográfica elaborada automaticamente de acordo com os dados fornecidos pelo(a) autor(a).
Sistema Integrado de Bibliotecas da Universidade do Estado do Amazonas.

D541o

Dias Filho, Caio Augusto Mascarenhas

Otimização Bi-Objetivo do Projeto Conceitual de Aeronaves de Baixo Número de Reynolds Utilizando Algoritmos Genéticos / Caio Augusto Mascarenhas Dias Filho . Manaus : [s.n], 2024.

70 f.: color.; 21,0 cm.

TCC - Graduação em Engenharia Mecânica- Universidade do Estado do Amazonas, Manaus, 2024.

Inclui Bibliografia.

Inclui Apêndice.

Orientador: Cruz Neto, Rubelmar Maia de Azevedo.

1. Projeto Conceitual Aeronáutico. 2. Otimização Multiobjetivo. 3. Algoritmos Evolutivos. 4. SAE Brasil AeroDesign. I. Cruz Neto, Rubelmar Maia de Azevedo (Orient.) II. Universidade do Estado do Amazonas. III. Título

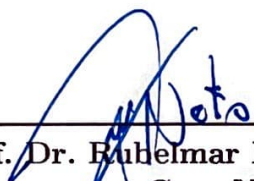
CDU(1997)621

Caio Augusto Mascarenhas Dias Filho

Otimização Bi-Objetivo do Projeto Conceitual de Aeronaves de Baixo Número de Reynolds Utilizando Algoritmos Genéticos

Trabalho de Conclusão de Curso apresentado como requisito parcial à obtenção do título de bacharel em Engenharia Mecânica da Universidade do Estado do Amazonas (UEA).

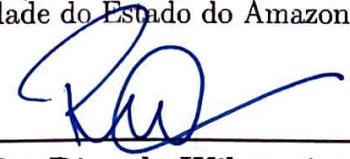
Trabalho aprovado. Manaus, 19 de dezembro de 2024:



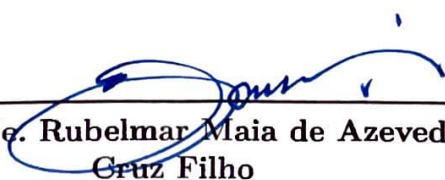
**Prof. Dr. Rubelmar Maia de Azevedo
Cruz Neto**
Universidade do Estado do Amazonas (UEA)



**Prof. Dr. Joao D'Anuzio Lima de
Azevedo**
Universidade do Estado do Amazonas (UEA)



**Prof. Dr. Ricardo Wilson Aguiar da
Cruz**
Universidade do Estado do Amazonas (UEA)



**Prof. Me. Rubelmar Maia de Azevedo
Cruz Filho**
Universidade do Estado do Amazonas (UEA)

Manaus - Amazonas

2024

Agradecimentos

Agradeço à minha família, por todo o apoio, carinho e suporte que viabilizaram a conclusão desta graduação.

Aos meus avós paternos, Hildeberto Côrrea Dias e Edinea Mascarenhas Dias por nutrirem minhas paixões pela aviação e pela área acadêmica.

Ao meu orientador, Prof. Dr. Rubelmar Maia de Azevedo Cruz Neto, pela amizade, conselhos, ensinamentos e orientação ao longo dos últimos três anos.

À minha namorada, Gyovanah Ágata por todo o amor, carinho e paciência.

Aos meus amigos da equipe Urutau AeroDesign, pelos bons momentos compartilhados ao longo dos três anos de projeto que tive o prazer de participar.

Aos meus amigos do setor de aerodinâmica, com quem tive a satisfação de compartilhar conhecimento, aprender muito e testemunhar o desenvolvimento de excelentes profissionais.

Aos amigos que fiz durante a graduação, e que levarei para toda a vida.

A todos os professores do curso de Engenharia Mecânica, pela excelência no ensino.

*"Heavier than air flying machines are impossible."
(Lord Kelvin, President of the Royal Society)*

Resumo

O projeto conceitual aeronáutico corresponde à etapa inicial do desenvolvimento de uma aeronave, em que diversas possíveis configurações são testadas a fim de identificar o conceito que melhor atenda aos requisitos de projeto. Este processo é demorado e trabalhoso quando realizado por métodos tradicionais baseados em múltiplas iterações humanas. Uma das técnicas mais eficazes para agilizar esse processo consiste na aplicação de uma rotina de otimização de projeto capaz de apontar o conceito ideal com maior precisão e menor tempo, através da otimização de uma função objetivo e de restrições estabelecidas. Neste contexto, o presente trabalho tem como objetivo desenvolver um modelo de otimização bi-objetivo aplicado ao projeto conceitual de uma aeronave de baixo número de Reynolds. Os objetivos estabelecidos consistem na maximização do peso máximo de decolagem e na minimização do peso vazio. O modelo contempla nove variáveis de projeto, abrangendo a escolha do aerofólio, do sistema propulsivo e parâmetros geométricos da asa. Cada configuração é avaliada por uma rotina que integra as análises aerodinâmicas do *software Athena Vortex Lattice* a um algoritmo que realiza uma simulação da decolagem. Os resultados obtidos confirmaram a capacidade do modelo de otimização bi-objetivo de identificar configurações que atendem aos requisitos de projeto apresentando alto desempenho. Além disso, o modelo demonstrou um baixo custo computacional, permitindo a avaliação de mais de 1500 soluções por hora. A solução ótima identificada atendeu aos objetivos estabelecidos para a competição SAE Brasil AeroDesign, contribuindo para a competitividade do projeto e destacando a eficiência da abordagem aplicada.

Palavras-chave: Projeto Conceitual Aeronáutico. Otimização Multiobjetivo. Algoritmos Evolutivos.

Abstract

Aircraft conceptual design constitutes the initial phase of aircraft development, in which numerous potential configurations are tested to identify the concept that best meets the project requirements. This process is both time-consuming and labor-intensive when conducted through traditional methods that rely on multiple human iterations. One of the most effective techniques to expedite this stage involves applying a design optimization routine that can determine the ideal concept with greater accuracy and shorter time cost, through the optimization of an objective function and established constraints. In this context, the present study aims to develop a bi-objective optimization model applied to the conceptual design of a low Reynolds number aircraft. The established objectives are to maximize the maximum takeoff and minimize the empty weight. A total of nine design variables were included in the optimization process, representing the selection of the airfoil, propulsive system, and wing's geometric variables. Each configuration is evaluated through a routine that integrates the aerodynamic analyses of the Athena Vortex Lattice software with an algorithm that simulates takeoff performance. The results showed the capability of the bi-objective optimization model to identify configurations that meet the design requirements while delivering high performance. Additionally, the model demonstrated low computational cost, enabling the evaluation of more than 1500 solutions per hour. The optimal solution identified successfully met the objectives established for the SAE Brasil AeroDesign competition, contributing to the project's competitiveness and highlighting the efficiency of the applied approach.

Keywords: Aircraft Conceptual Design. Multiobjective Optimization. Evolutionary Algorithms.

Lista de ilustrações

Figura 1 – Aeronave da equipe Urutau AeroDesign em 2024.	12
Figura 2 – Fluxograma de um projeto aeronáutico.	15
Figura 3 – Peça de uma aeronave nas diferentes etapas de projeto.	16
Figura 4 – Parâmetros geométricos de um aerofólio.	18
Figura 5 – Forças e momento aerodinâmicos atuantes em um aerofólio.	18
Figura 6 – Coeficiente de sustentação em função do ângulo de ataque de um aerofólio.	19
Figura 7 – Parâmetros geométricos de uma asa.	20
Figura 8 – Geração de vórtices na ponta de uma asa finita.	20
Figura 9 – Efeito do alongamento na sustentação de uma asa.	21
Figura 10 – Efeito da razão de afilamento na distribuição de sustentação de uma asa.	22
Figura 11 – Aplicação de <i>washout</i> geométrico em uma asa.	23
Figura 12 – Efeito da torção negativa na distribuição de sustentação de uma asa.	23
Figura 13 – Representação esquemática da corrida de decolagem.	24
Figura 14 – Forças atuantes na aeronave durante a decolagem.	24
Figura 15 – Representação esquemática dos vórtices em asas finitas.	27
Figura 16 – Fluxograma do ciclo de um projeto realizado de duas maneiras distintas.	28
Figura 17 – Três soluções (A, B e C) representadas em função dos objetivos f_1 e f_2	29
Figura 18 – Formação da Fronteira de Pareto pelas soluções não dominadas.	29
Figura 19 – Representação de uma população em um Algoritmo Genético.	30
Figura 20 – Requisitos de decolagem para a Classe Advanced.	31
Figura 21 – Roleta com a torção intermediária correspondente a cada intervalo.	34
Figura 22 – Curvas aerodinâmicas dos aerofólios.	35
Figura 23 – Empuxo em função da velocidade dos sistemas propulsivos.	36
Figura 24 – Análise de convergência de malha.	38
Figura 25 – Análise de convergência de malha.	38
Figura 26 – Representação do funcionamento da seleção pela Roda da Fortuna.	41
Figura 27 – Comportamento do <i>fitness</i> dos indivíduos ao longo das gerações.	43
Figura 28 – Geometria da asa dos melhores indivíduos ao longo das gerações.	44
Figura 29 – Comportamento bi-objetivo do problema de otimização.	45
Figura 30 – Distribuição dos indivíduos em função de suas variáveis de projeto.	46
Figura 31 – Modelagem tridimensional da solução ótima.	48
Figura 32 – Simulação da decolagem da solução ótima em seu <i>MTOW</i>	48

Lista de abreviaturas e siglas

AG	Algoritmo Genético
AVL	<i>Athena Vortex Lattice</i>
GIF	<i>Graphics Interchange Format</i>
VLM	Método <i>Vortex Lattice</i>
MDO	Otimização de Projeto Multidisciplinar
MTOW	Peso Máximo de Decolagem
EW	Peso Vazio
SAE	<i>Society of Automotive Engineers</i>
UruA	<i>Urutau Airfoil</i>

Sumário

1	INTRODUÇÃO	12
1.1	Objetivos	13
1.1.1	Objetivos Específicos	14
1.2	Estrutura do Trabalho	14
2	REFERENCIAL TEÓRICO	15
2.1	Metodologia de um Projeto Aeronáutico	15
2.2	Projeto Aeronáutico Conceitual	16
2.2.1	Projeto Aerodinâmico na Etapa Conceitual	17
2.2.1.1	Aerodinâmica de Aerofólios	17
2.2.1.2	Aerodinâmica de Asas Finitas	19
2.2.2	Performance de Decolagem na Etapa Conceitual	23
2.3	<i>Athena Vortex Lattice (AVL)</i>	26
2.4	Otimização de Projeto	27
2.4.1	Otimização Multiobjetivo	28
2.4.2	Algoritmos Genéticos	30
3	MATERIAIS E MÉTODOS	31
3.1	Requisitos do Projeto	31
3.2	Elaboração da Função Objetivo	32
3.3	Escolha das Variáveis de Projeto	33
3.3.1	Parâmetros Geométricos da Asa	34
3.3.2	Aerofólios	34
3.3.3	Sistemas Propulsivos	36
3.4	Avaliações Aerodinâmicas	37
3.4.1	Criação do Modelo da Asa	37
3.4.2	Execução das Avaliações Aerodinâmicas	38
3.5	Avaliações de Performance de Decolagem	39
3.6	Desenvolvimento do Algoritmo Genético	39
3.6.1	Definição da População Inicial	40
3.6.2	Avaliação dos Indivíduos	40
3.6.3	Operador de Seleção	40
3.6.4	Operador de Cruzamento	41
3.6.5	Operador de Mutação	41
3.6.6	Operador de Elitismo	42

4	RESULTADOS E DISCUSSÃO	43
4.1	Comportamento da Otimização Bi-objetivo	43
4.2	Parâmetros da Solução Ótima	47
5	CONCLUSÃO	49
5.1	Sugestões para Trabalhos Futuros	50
	REFERÊNCIAS	51
	APÊNDICES	53
	APÊNDICE A – CRIAÇÃO DO ARQUIVO DA ASA	54
	APÊNDICE B – AVALIAÇÕES AERODINÂMICAS	56
	APÊNDICE C – SIMULAÇÃO DA DECOLAGEM	58
	APÊNDICE D – PERFORMANCE DE DECOLAGEM	60
	APÊNDICE E – FUNÇÃO <i>FITNESS</i>	61
	APÊNDICE F – RODA DA FORTUNA	63
	APÊNDICE G – OPERADOR DE CRUZAMENTO	64
	APÊNDICE H – ALGORITMO GENÉTICO	65
	APÊNDICE I – GIF DA EVOLUÇÃO	69

1 Introdução

Nos primórdios do século XX, acreditava-se que uma máquina mais pesada que o ar jamais conseguiria voar. Nesse contexto, surgiram dois projetos pioneiros obstinados a mudar essa visão: o *14-bis*, desenvolvido pelo brasileiro Alberto Santos Dumont, e o *Flyer I*, criado pelos estadunidenses Orville e Wilbur Wright. Embora ambos surgissem com o mesmo objetivo de provar que seria possível construir uma aeronave capaz de realizar um voo sustentado e controlado, cada projeto teve sucesso graças a soluções distintas.

Tanto o *14-bis* quanto o *Flyer I* foram resultados de estudos em aerodinâmica, particularmente no que diz respeito aos conceitos de sustentação e arrasto. Entretanto, cada aeronave teve seus triunfos. No caso do *14-bis*, destaca-se a aplicação eficaz do diedro e da incidência da asa. Já o *Flyer I* demonstrou avanços consideráveis em termos de aerofólios, alcançados por meio de ensaios em túnel de vento (Anderson, 2024).

O princípio de que diferentes soluções podem atender ao mesmo objetivo permanece atual, figurando de forma particularmente evidente para alunos universitários na Competição SAE BRASIL AeroDesign. Desde 1999, esse evento integra o calendário da SAE BRASIL, promovendo o intercâmbio de conhecimentos voltados à engenharia aeronáutica por meio de um projeto real e competitivo (Ribeiro, 2023). O desenvolvimento do cenário de competição leva as equipes participantes a buscarem aeronaves cada vez mais competitivas, seja através do aumento de carga paga, redução de peso vazio, dentre outras possibilidades (Dias Filho *et al.*, 2024). Na Figura 1, é apresentado o resultado de um projeto competitivo desenvolvido pela equipe Urutau AeroDesign, da Universidade do Estado do Amazonas (UEA). Este projeto conquistou a 3ª Colocação da Classe *Advanced* na Competição SAE Brasil AeroDesign 2024.

Figura 1 – Aeronave da equipe Urutau AeroDesign em 2024.



Fonte: (Fujiki, 2024).

No cenário atual, a aplicação de técnicas de otimização no projeto conceitual se mostra fundamental, especialmente para as equipes da Classe *Advanced* da competição, categoria destinada a projetos mais ousados e inovadores. Essas técnicas permitem analisar escolhas necessárias entre características conflitantes do projeto, com maior acurácia e em menor tempo. Essas escolhas, chamadas de *trade-offs*, ocorrem quando melhorar um aspecto do projeto implica em sacrificar outro, e a aplicação de algoritmos de otimização permite encontrar o melhor equilíbrio entre esses fatores. Vários trabalhos demonstram o potencial desses algoritmos para o desenvolvimento de aeronaves de AeroDesign.

Bortolete (2017), por exemplo, apresentou um algoritmo de otimização aerodinâmica da geometria de asas para a Classe *Advanced* da competição SAE AeroDesign utilizando o *modeFrontier*[®]. Esse estudo aumentou a eficiência do projeto de asas da equipe Urutau AeroDesign, reduzindo o tempo de uma abordagem iterativa que levava 14 dias para apenas 1 dia. No entanto, ainda notou-se a necessidade de algumas aplicações, como a inclusão de novas variáveis, o refinamento das análises e a diminuição do custo computacional.

Já Vilela e Silva (2019) propuseram um modelo de otimização com algoritmos genéticos para maximizar a pontuação na competição, focando nas variáveis geométricas da asa da Classe *Advanced*. De forma semelhante, Ribeiro (2023) desenvolveu um algoritmo de otimização multidisciplinar, também baseado em algoritmos genéticos, para o dimensionamento de uma aeronave para competições de AeroDesign. Esse trabalho incorporou uma gama maior de variáveis de projeto, incluindo a geometria da empenagem, a escolha do aerofólio e o sistema propulsivo. Contudo, tanto em Vilela e Silva (2019) quanto em Ribeiro (2023), foi resolvido um problema de otimização mono-objetivo, focando na maximização da pontuação na competição e do peso máximo de decolagem da aeronave, respectivamente, sem considerar a minimização do peso vazio como fator relevante.

Dessa forma, o objetivo deste trabalho é desenvolver um modelo de otimização bi-objetivo, que utiliza a técnica dos Algoritmos Genéticos, para otimizar o projeto conceitual de uma aeronave de baixo número de Reynolds. Nesse processo de otimização, busca-se, simultaneamente, maximizar o peso máximo de decolagem e minimizar o peso vazio da aeronave. As rotinas de avaliação adotadas integram as disciplinas de aerodinâmica e de desempenho na decolagem, consideradas fundamentais em um primeiro esforço para a construção de um otimizador de projeto aeronáutico multidisciplinar.

1.1 Objetivos

Desenvolver um modelo de otimização bi-objetivo que aplique o método dos Algoritmos Genéticos para otimizar o projeto conceitual de uma aeronave de baixo número de Reynolds, maximizando o seu peso máximo de decolagem e minimizando o seu peso vazio.

1.1.1 Objetivos Específicos

- Desenvolver uma rotina de avaliação aerodinâmica e de performance de decolagem compatível à adição de novas variáveis de projeto;
- Desenvolver um otimizador de projeto bi-objetivo que produza resultados confiáveis com o menor custo computacional possível;
- Elaborar uma aplicação de otimização autoral utilizando Algoritmos Genéticos.

1.2 Estrutura do Trabalho

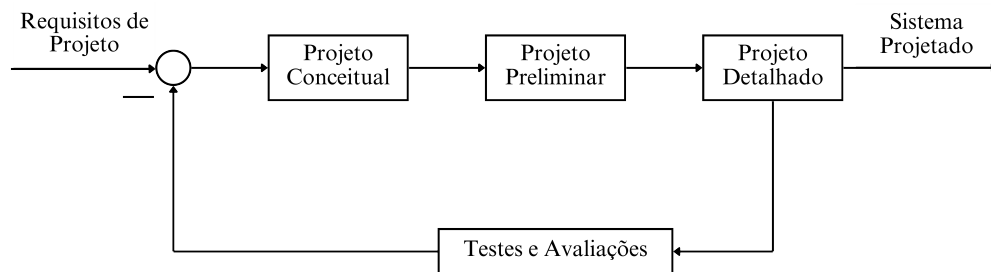
Esta monografia está organizada em cinco capítulos. O [Capítulo 1](#) apresenta o contexto e as motivações do projeto. O [Capítulo 2](#) expõe o referencial teórico utilizado no estudo. Em seguida, o [Capítulo 3](#) descreve a metodologia aplicada no projeto. O [Capítulo 4](#) apresenta e discute os resultados obtidos. Por fim, o [Capítulo 5](#) traz as conclusões e as sugestões de trabalhos futuros.

2 Referencial Teórico

2.1 Metodologia de um Projeto Aeronáutico

Segundo a metodologia apresentada por [Raymer \(2024\)](#), o projeto aeronáutico pode ser dividido em três fases, conforme ilustrado na [Figura 2](#). Com base nos requisitos de projeto previamente estabelecidos, inicia-se a etapa de projeto conceitual, seguida pelas etapas de projeto preliminar e detalhado, culminando na etapa de fabricação, testes e avaliações da aeronave.

Figura 2 – Fluxograma de um projeto aeronáutico.



Fonte: Adaptado de [Sadraey \(2024\)](#).

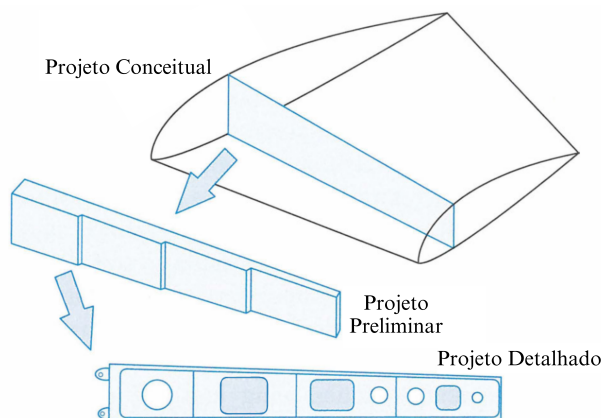
A etapa de projeto conceitual dá início ao desenvolvimento do projeto aeronáutico, com o objetivo de selecionar um único conceito de aeronave que atenda satisfatoriamente às exigências determinadas. Essa fase constitui o ponto de partida para o desenvolvimento de uma aeronave, fornecendo as dimensões iniciais, estimativas de peso e uma análise preliminar de seu desempenho, aferindo a sua capacidade de cumprir com a missão a qual foi projetada. Para alcançar o conceito ideal, desenvolve-se e analisa-se uma ampla variedade de configurações, nas quais são realizados *trade-offs* para identificar a configuração inicial que melhor se adapta aos requisitos definidos.

Após isso, segue-se para o projeto preliminar, fase em que o conceito melhor avaliado será refinado o suficiente a fim de possibilitar a análise de sua viabilidade em atender aos requisitos estabelecidos. A partir dessa etapa, não são realizadas grandes mudanças no projeto, sendo esperadas apenas pequenas otimizações ou revisões. Com o avanço dessa fase, aumenta-se gradativamente a confiança no projeto, até que ele esteja suficientemente consolidado para prosseguir à etapa de projeto detalhado.

Nas fases de projeto conceitual e preliminar, não são desenvolvidos projetos específicos dos componentes. Considerando que o conceito ainda está sujeito a alterações, apenas decisões gerais são tomadas ([Raymer, 2024](#)). Por outro lado, na etapa de projeto detalhado, os componentes passam a ser projetados em maior nível de detalhamento, tornando-se

aptos à fabricação e montagem. Ao final dessa parte do processo, a aeronave está pronta para os testes e validações. A [Figura 3](#) ilustra de forma clara as diferenças no nível de detalhamento de uma peça em cada uma das etapas do projeto.

Figura 3 – Peça de uma aeronave nas diferentes etapas de projeto.



Fonte: Adaptado de [Raymer \(2024\)](#).

2.2 Projeto Aeronáutico Conceitual

Aeronaves de baixo números de Reynolds, como as desenvolvidas para competições de AeroDesign, seguem as mesmas etapas do projeto aeronáutico tradicional. Entretanto, nesse contexto, os projetos precisam ser concluídos no menor tempo possível. Em menos de um ano, os requisitos estipulados pelo regulamento da competição devem ser transformados em uma aeronave confiável, competitiva e segura. Por esse motivo, busca-se acelerar ao máximo o processo de desenvolvimento, especialmente a etapa de projeto conceitual, que pode demandar bastante tempo se não for realizada com técnicas refinadas. Nessa fase, diversas configurações de aeronaves precisam ser testadas para encontrar um conceito que atenda da melhor forma possível às exigências estabelecidas.

Uma das técnicas mais eficazes para a realização dos *trade-offs* no projeto conceitual é a aplicação de modelos de Otimização de Projeto Multidisciplinar (MDO). Segundo [Sadraey \(2024\)](#), essa técnica permite que projetistas incorporem todas as disciplinas do projeto simultaneamente, resultando em uma solução ótima que considera todas as disciplinas envolvidas no problema de maneira integrada.

A aplicação de modelos de MDO requer um elevado número de avaliações dos objetivos e das restrições estabelecidas. Essas técnicas são guiadas por uma função objetivo sujeita a um conjunto de restrições, que passa por processos matemáticos voltados para a otimização dessa função. Quanto mais simples forem os objetivos de missão da aeronave, mais simples é a formulação da função objetivo. Da mesma forma, quanto menor o número de disciplinas envolvidas, mais rápido será o processo de busca pela solução ótima.

Em equipes que desenvolvem aeronaves para a competição SAE BRASIL AeroDesign, os modelos de MDO são elaborados e aprimorados ao longo de vários anos de execução projeto, evoluindo juntamente com a experiência da equipe. Entretanto, duas disciplinas essenciais para o desenvolvimento inicial desses modelos são o projeto aerodinâmico e a análise de performance de decolagem. Isso se deve ao fato de a decolagem ser o ponto mais crítico em projetos voltados para a competição, considerando os requisitos estabelecidos pelo regulamento.

2.2.1 Projeto Aerodinâmico na Etapa Conceitual

O projeto aerodinâmico conceitual de uma aeronave para competições de AeroDesign tem como objetivo determinar as dimensões iniciais da asa, bem como definir o aerofólio a ser utilizado no projeto. Nessa fase, não se considera a implementação de dispositivos hypersustentadores, dispositivos de ponta de asa, e superfícies de comando. Além disso, determinadas simplificações são aplicadas, visando reduzir o esforço computacional das análises de *trade-offs* iniciais. Entre essas simplificações, destaca-se a exclusão do efeito solo e a ausência de análises mais detalhadas quanto ao arrasto parasita, as quais devem ser realizadas em etapas posteriores do projeto.

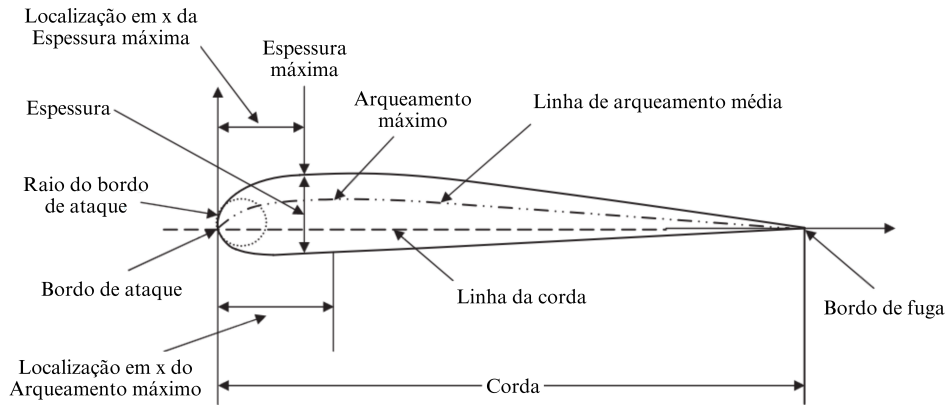
De acordo com [Anderson \(2024\)](#), um grande avanço na análise de aeronaves foi a percepção de que as considerações aerodinâmicas de uma asa podem ser divididas em duas partes: o estudo da seção bidimensional de uma asa, um aerofólio; e a modificação das propriedades bidimensionais para a representação de uma asa finita. Essa abordagem, proposta por Ludwig Prandtl e outros pesquisadores de Göttingen na década de 1910, é usada até os dias atuais.

2.2.1.1 Aerodinâmica de Aerofólios

Segundo [Raymer \(2024\)](#), o aerofólio pode ser considerado o “coração” da aeronave, uma vez que influencia diretamente a velocidade de cruzeiro, as distâncias de decolagem e pouso, a velocidade de estol, a manobrabilidade e a eficiência aerodinâmica em todas as fases do voo, sendo uma peça chave no projeto aerodinâmico de uma aeronave. A [Figura 4](#) apresenta os principais parâmetros geométricos de um aerofólio.

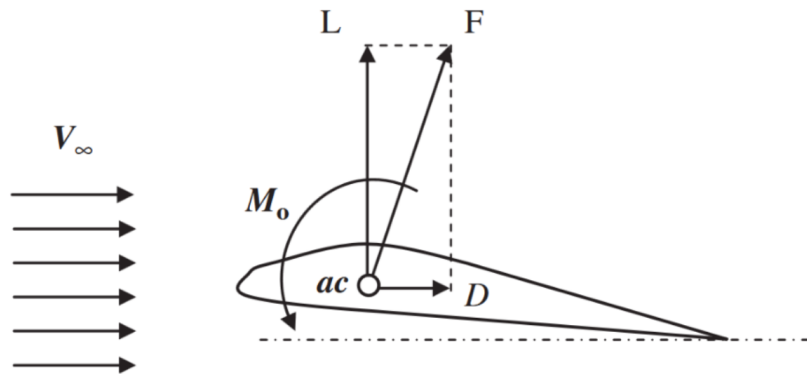
O principal objetivo do aerofólio é gerar sustentação, o que ocorre pela alteração na velocidade do ar que percorre as superfícies superior e inferior do perfil. Essa alteração é gerada pela combinação entre o ângulo de ataque (α) – definido entre a linha de corda do aerofólio e a direção do componente velocidade do escoamento (V_∞) – e o arqueamento do aerofólio, resultando em uma diferença de pressão entre as faces. Essa diferença de pressão gera uma força resultante (F) no centro aerodinâmico do perfil (ac), que é decomposta em duas forças: sustentação (L) e arrasto (D), e no momento de arfagem (M_o), conforme ilustrado na [Figura 5](#).

Figura 4 – Parâmetros geométricos de um aerofólio.



Fonte: Adaptado de [Sadraey \(2024\)](#).

Figura 5 – Forças e momento aerodinâmicos atuantes em um aerofólio.



Fonte: ([Sadraey, 2024](#)).

No projeto de aerofólios, as forças aerodinâmicas e o momento são analisados por meio de coeficientes adimensionais. A [Equação 2.1](#) apresenta o coeficiente de sustentação (C_l), a [Equação 2.2](#) o coeficiente de arrasto (C_d) e a [Equação 2.3](#) o coeficiente de momento (C_m), conforme apresentado em [Raymer \(2024\)](#). Essa abordagem facilita a comparação entre diferentes aerofólios.

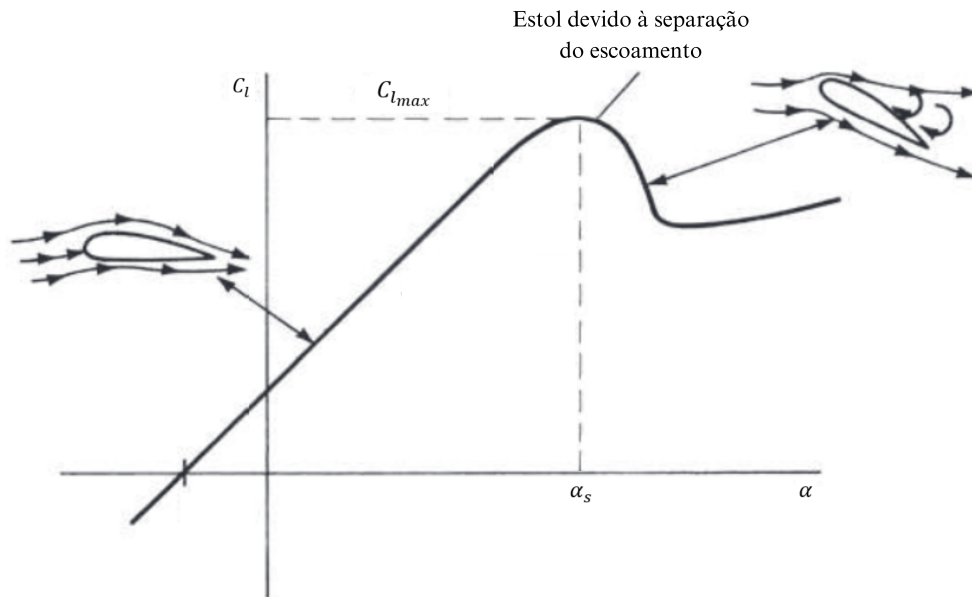
$$C_l = \frac{L}{\frac{1}{2}\rho_\infty V_\infty^2 c} \quad (2.1)$$

$$C_d = \frac{D}{\frac{1}{2}\rho_\infty V_\infty^2 c} \quad (2.2)$$

$$C_M = \frac{M_o}{\frac{1}{2}\rho_\infty V_\infty^2 c^2} \quad (2.3)$$

Outra consideração importante no projeto de aerofólios é o comportamento da curva de C_l em função do α . Conforme ilustrado na [Figura 6](#), para baixos valores de α , o C_l cresce linearmente. Contudo, ao atingir o ângulo de estol ¹ (α_s), a capacidade do aerofólio de gerar sustentação atinge seu valor máximo ($C_{l_{max}}$) e, em seguida, é abruptamente reduzida devido à separação do escoamento na superfície superior.

Figura 6 – Coeficiente de sustentação em função do ângulo de ataque de um aerofólio.



Fonte: Adaptado de [Anderson \(2024\)](#).

Essas análises das forças e propriedades aerodinâmicas devem ser realizadas para as diferentes condições de voo às quais a aeronave é submetida. Uma maneira de representar essas condições é por meio do coeficiente adimensional chamado número de Reynolds (Re), que estabelece a razão entre as forças inerciais e viscosas presentes em um escoamento, conforme apresentado na [Equação 2.4](#). ([Anderson, 2024](#)).

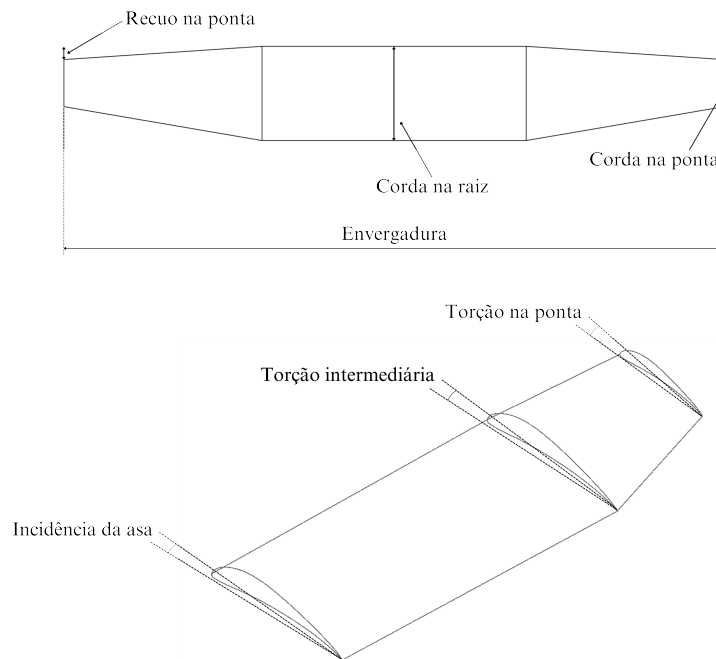
$$Re = \frac{\rho_{\infty} V_{\infty} c}{\mu_{\infty}} \quad (2.4)$$

2.2.1.2 Aerodinâmica de Asas Finitas

Ao extrapolar as propriedades de um aerofólio para o domínio tridimensional, considera-se a junção de diversas seções bidimensionais, formando uma asa finita. O estudo da asa finita baseia-se em como as propriedades dos aerofólios individuais são modificadas pelos efeitos tridimensionais. A [Figura 7](#) apresenta os principais parâmetros geométricos da asa relevantes na etapa de projeto conceitual.

¹ O termo estol refere-se a esta perda drástica de sustentação.

Figura 7 – Parâmetros geométricos de uma asa.

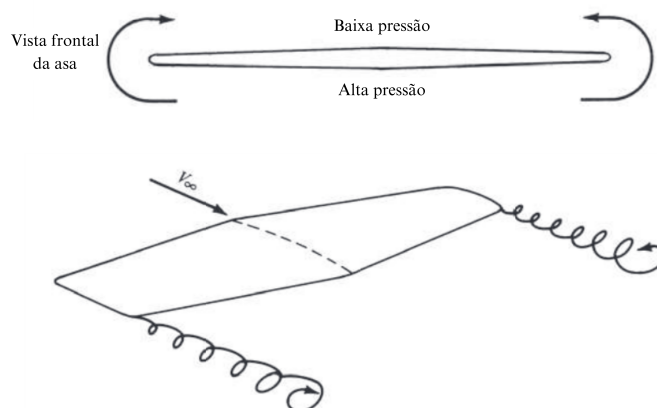


Fonte: Elaborado pelo autor.

Assim como no caso dos aerofólios, as forças e momentos atuantes em uma asa são expressos por coeficientes adimensionais. No entanto, em suas equações, o termo referente à corda do aerofólio é substituído pela área da asa. Além disso, em uma asa tridimensional, o coeficiente de arrasto associado ao aerofólio é tratado como um componente do coeficiente de arrasto da asa, denominado coeficiente de arrasto parasita (C_{D_0}) (Anderson, 2024).

Diferentemente de uma asa bidimensional, quando uma asa finita gera sustentação, a diferença de pressão entre as superfícies superior e inferior faz com que o ar na região do intradorso tenda a escapar para o extradorso, gerando vórtices na ponta da asa, conforme ilustrado na Figura 8.

Figura 8 – Geração de vórtices na ponta de uma asa finita.



Fonte: Adaptado de Anderson (2024).

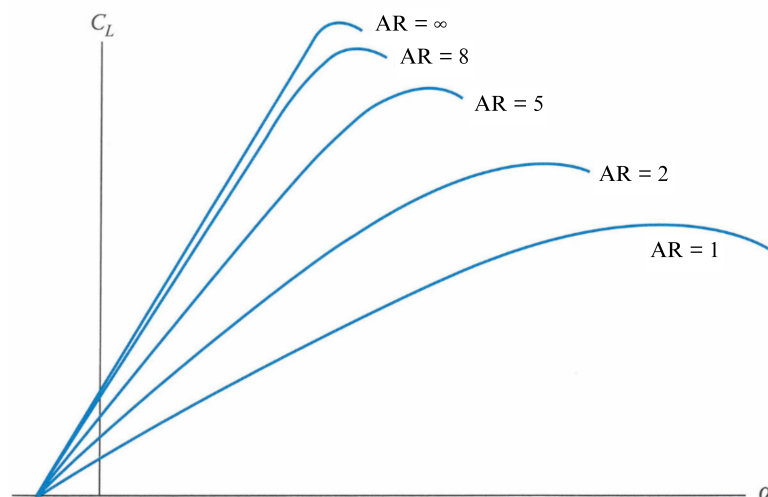
Esses vórtices reduzem a sustentação atuante nas pontas da asa e geram uma componente de arrasto induzido pela sustentação, mensurado utilizando-se o coeficiente de arrasto induzido ² (C_{D_i}) (Raymer, 2024). Conseqüentemente, para duas asas de mesma área, aquela com maior envergadura é menos afetada pelo vórtice da ponta, uma vez que a área comprometida pela formação dos vórtices representa uma fração menor da área total.

Dessa comparação entre asas de áreas iguais, mas com diferentes envergaduras, surge a propriedade geométrica denominada alongamento (AR), calculada por meio da Equação 2.5, conforme exposto por Anderson (2024), em que b representa a envergadura de uma asa (m), e S a sua área (m^2).

$$AR = \frac{b^2}{S} \quad (2.5)$$

O comportamento da sustentação em função do alongamento está ilustrado na Figura 9, onde pode-se observar que a curva de sustentação em função de α para asas mais alongadas tende a se aproximar do comportamento de seu aerofólio. Por outro lado, quanto menor o valor de AR de uma asa, maiores são os seus valores de α_s , porém menores os seus valores de $C_{l_{max}}$. Embora asas mais alongadas apresentem maior capacidade de geração de sustentação, elas também são estruturalmente mais complexas, configurando, assim, um importante *trade-off* no projeto aerodinâmico.

Figura 9 – Efeito do alongamento na sustentação de uma asa.



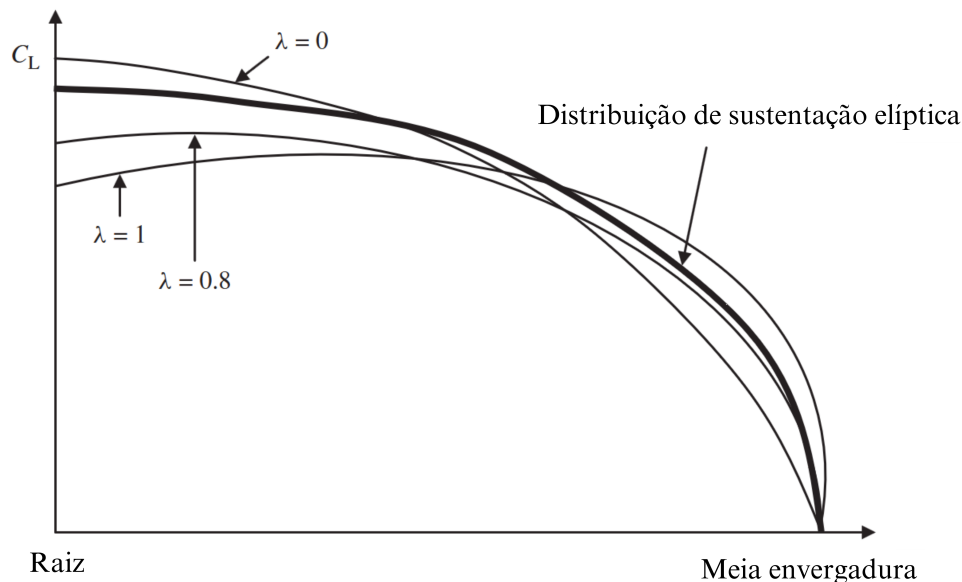
Fonte: Adaptado de Raymer (2024).

Dessa forma, diferentes classes de aeronaves apresentam, em geral, intervalos característicos de AR . Por exemplo, caças militares costumam exibir valores de AR entre 2 e 4; aeronaves cargueiras apresentam tipicamente alongamento entre 6 e 12; já os planadores e *sailplanes* costumam ter valores de AR na faixa de 20 a 40 (Sadraey, 2024).

² O coeficiente de arrasto de uma asa finita (C_D) é obtido através da soma entre o C_{D_0} e o C_{D_i} (Anderson, 2024).

Ao analisar uma asa, é importante considerar o comportamento da sustentação em toda sua área. Através da contribuição na sustentação de cada seção ao longo da envergadura, obtém-se a chamada distribuição de sustentação. Determinadas propriedades geométricas da asa influenciam diretamente nessa distribuição, uma delas é a razão de afilamento (λ), definida como a razão entre a corda na ponta e a corda na raiz. Em um projeto aerodinâmico, buscam-se asas que apresentem a sua distribuição de sustentação o mais próxima possível da distribuição elíptica de sustentação. Isso porque esta apresenta a maior eficiência aerodinâmica, e o controle da razão de afilamento de uma asa pode auxiliar nesta busca. O comportamento típico da λ na distribuição de sustentação está representado na [Figura 10](#).

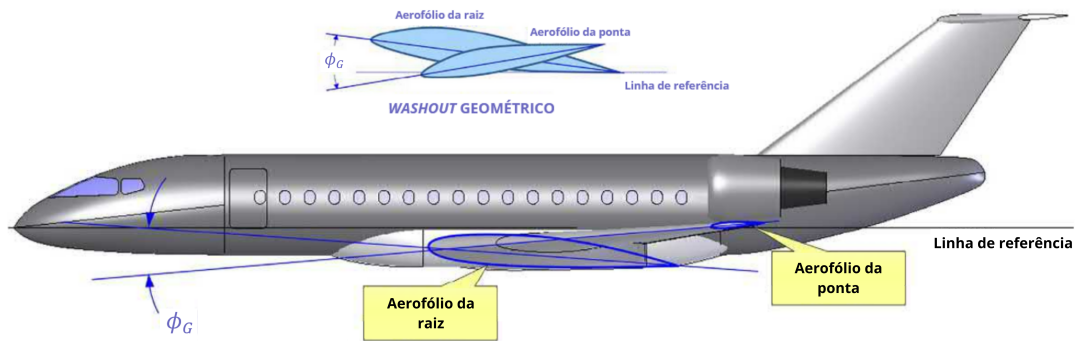
Figura 10 – Efeito da razão de afilamento na distribuição de sustentação de uma asa.



Fonte: Adaptado de [Sadraey \(2024\)](#).

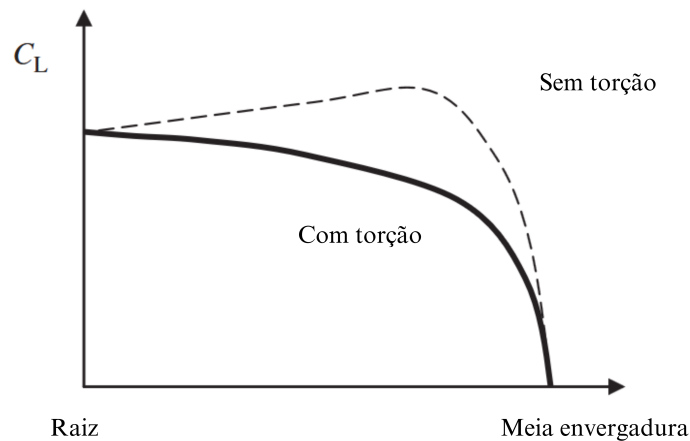
Asas com valores muito elevados de afilamento, que se aproximam de asas retangulares, são estruturalmente ineficientes, pois apresentam uma elevada área próxima às pontas com baixa capacidade de geração de sustentação e alto arrasto induzido. Entretanto, elas são mais econômicas de serem manufaturadas. Por outro lado, asas com valores baixos de afilamento tendem a apresentar o início de estol na ponta da asa, o que é indesejável para as características de estabilidade e controle lateral da aeronave. Porém, elas tendem a apresentar boa geração de sustentação.

Uma forma de contornar este problema é por meio da combinação de asas de baixos valores de λ com a introdução de uma torção geométrica nas seções da asa, aplicando ângulos negativos nos aerofólios em direção à ponta da asa (*washout*). A [Figura 11](#) representa a geometria de uma asa com *washout*. Desta forma, desloca-se o ponto de início de estol para próximo da raiz, corrigindo a distribuição de sustentação. O comportamento típico da torção negativa na distribuição de sustentação está representado na [Figura 12](#).

Figura 11 – Aplicação de *washout* geométrico em uma asa.

Fonte: Adaptado de Gudmundsson (2022).

Figura 12 – Efeito da torção negativa na distribuição de sustentação de uma asa.



Fonte: Adaptado de Sadraey (2024).

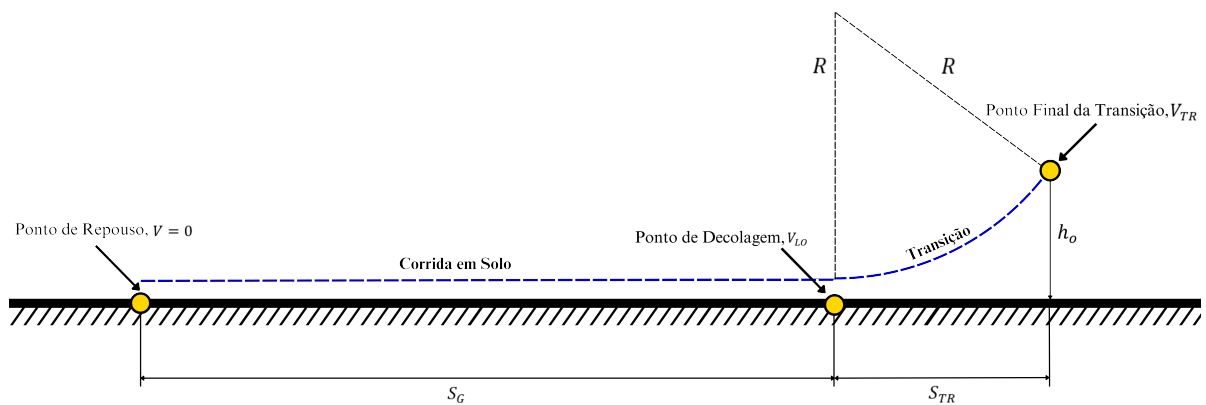
Após a execução dos devidos *trade-offs* de projeto aerodinâmico, é possível definir uma configuração inicial de asa e aerofólio. Essas variáveis podem ser avaliadas para a obtenção de seus coeficientes aerodinâmicos, para, em seguida, ter a sua performance de decolagem analisada.

2.2.2 Performance de Decolagem na Etapa Conceitual

A análise de performance de decolagem tem como objetivo determinar o comprimento de pista necessário para que uma aeronave acelere do repouso até a decolagem, bem como definir a distância necessária para realizar uma subida segura (Gudmundsson, 2022). No contexto do projeto de aeronaves para a Competição SAE BRASIL AeroDesign, essa etapa é de grande importância devido às restrições impostas pelo regulamento da competição, como o peso máximo de decolagem (*MTOW*), a distância total de decolagem (*S_{TO}*) e a altura do obstáculo (*h_o*).

O processo de decolagem de uma aeronave é, geralmente, subdividido em quatro etapas: corrida em solo, rotação, transição e subida. No entanto, no projeto conceitual de aeronaves para competições de AeroDesign, esse modelo pode ser simplificado. Nele, é possível considerar apenas as etapas de corrida em solo e transição, visto que a análise apenas destas etapas fornece resultados próximos do processo de decolagem completo de uma aeronave, com um menor custo computacional, conforme ilustrado na [Figura 13](#). Essa simplificação reduz o tempo necessário para as simulações, mantendo resultados suficientemente precisos.

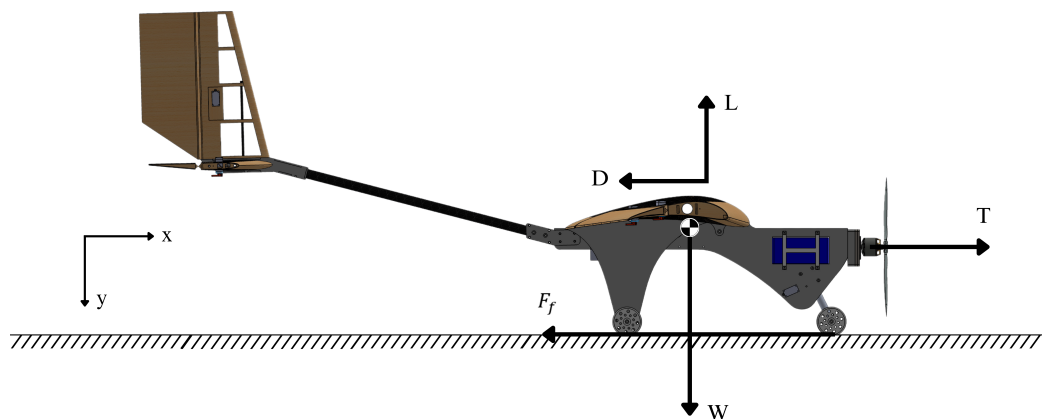
Figura 13 – Representação esquemática da corrida de decolagem.



Fonte: Adaptado de [Dias Filho *et al.* \(2024\)](#).

Durante a etapa de decolagem em uma pista reta, cinco forças atuam simultaneamente na aeronave, assim como ilustrado na [Figura 14](#): o empuxo gerado pelo sistema propulsivo (T), o arrasto (D), a sustentação (L), o peso da aeronave (W) e a força de atrito entre as rodas e o solo (F_f).

Figura 14 – Forças atuantes na aeronave durante a decolagem.



Fonte: Elaborado pelo autor.

A análise dessas forças é baseada na Segunda Lei de Newton, permitindo determinar a aceleração instantânea da aeronave ($\frac{dV}{dt}$), conforme representado pela [Equação 2.6](#):

$$\sum F = m \frac{dV}{dt} \quad (2.6)$$

Para a corrida em solo, considera-se apenas as forças paralelas ao solo, resultando na [Equação 2.7](#):

$$T - D - F_f = m \frac{dV}{dt} \quad (2.7)$$

A força de atrito (F_f) é calculada pelo [Equação 2.8](#), considerando o coeficiente de atrito com o solo (μ), que varia conforme o tipo de superfície. Segundo [Sadraey \(2023\)](#), para pistas de asfalto seco, como as usadas no AeroDesign, o valor do varia entre 0,03 e 0,05. A força normal (N) depende da velocidade da aeronave e é calculada como a diferença entre W e L . Substituindo-se esses valores, obtém-se a [Equação 2.9](#).

$$F_f = \mu N \quad (2.8)$$

$$T - D - \mu(W - L) = m \frac{dV}{dt} \quad (2.9)$$

O empuxo (T) é aferido experimentalmente em função da velocidade. Os dados obtidos através de ensaios de empuxo dinâmico são ajustados a uma curva de tendência, modelada por um polinômio quadrático, conforme descrito na [Equação 2.10](#).

$$T(V) = aV^2 + bV + c \quad (2.10)$$

As forças aerodinâmicas são calculadas por meio da [Equação 2.11](#) para a sustentação (L) e da [Equação 2.12](#) para o arrasto (D), conforme apresentado por [Raymer \(2024\)](#).

$$L(V) = \frac{1}{2} \rho_{\infty} V_{\infty}^2 S C_L \quad (2.11)$$

$$D(V) = \frac{1}{2} \rho_{\infty} V_{\infty}^2 S (C_{D_0} + C_{D_i}) \quad (2.12)$$

Para determinar o comprimento de pista necessário para o fim da corrida em solo (quando L supera W), utiliza-se a técnica de integração numérica, conforme proposto por [Gudmundsson \(2022\)](#). Essa abordagem permite simular a variação das forças a cada instante através da [Equação 2.13](#) e da [Equação 2.14](#), garantindo assim uma maior precisão na simulação da corrida de decolagem.

$$V_i = \int_0^t \frac{dV}{dt} dt = \sum_{i=0}^N \left(V_{i-1} + \frac{dV}{dt}_i \Delta t_i \right) \quad (2.13)$$

$$S_G = \int_0^t \left(V_0 + \frac{dV}{dt} t \right) dt = \sum_{i=0}^N \left(S_{i-1} + V_{i-1} \Delta t_i + \frac{1}{2} \frac{dV}{dt}_i (\Delta t_i)^2 \right) \quad (2.14)$$

Após a corrida de decolagem, a aeronave realiza a etapa de transição, superando o obstáculo ao final da pista. De acordo com [Raymer \(2024\)](#), nessa etapa, a aeronave segue um trajeto aproximado de um arco circular, acelerando da velocidade de decolagem (V_{LO}) até a velocidade de subida ($V_{CL} = 1,09V_{LO}$), dessa forma a velocidade média para o trecho de transição (V_{TR}) pode ser assumida como $1,045 V_{LO}$ e o raio do trajeto de transição (R) é calculado por meio da [Equação 2.15](#).

$$R = \frac{V_{TR}^2}{0,2g} \quad (2.15)$$

A distância necessária para a transição (S_{TR}), considerando a ultrapassagem do obstáculo ao seu fim, é calculada pela [Equação 2.16](#).

$$S_{TR} = \sqrt{R^2 - (R - h_o)^2} \quad (2.16)$$

Finalmente, a S_{TO} é a soma das distâncias de corrida em solo e de transição, conforme apresentado na [Equação 2.17](#).

$$S_{TO} = S_G + S_{TR} \quad (2.17)$$

2.3 Athena Vortex Lattice (AVL)

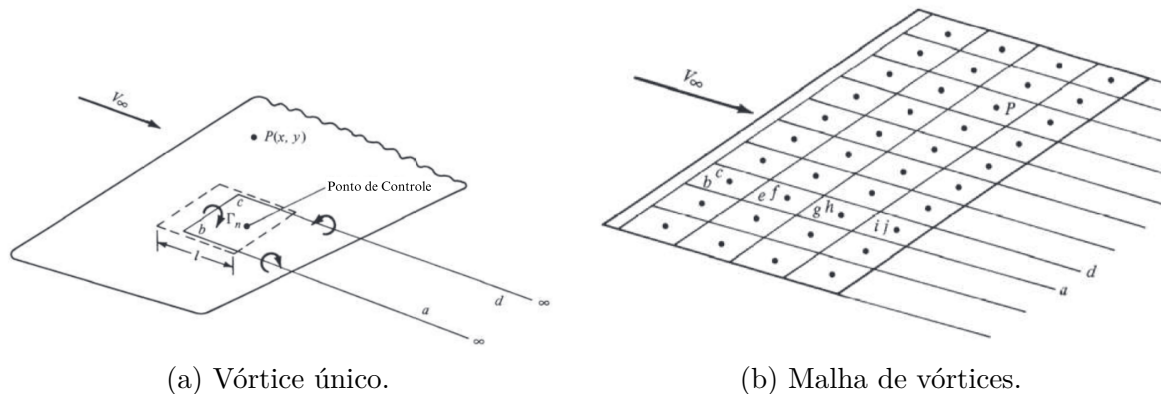
O *AVL* é um *software* desenvolvido por Mark Drela e Harol Youngren para a análise aerodinâmica e da dinâmica de voo de aeronaves rígidas com configurações arbitrárias. O programa utiliza o Método *Vortex Lattice* (*VLM*) para resolver o escoamento ao redor de superfícies sustentadoras.

O *VLM* é uma técnica numérica que trabalha com superfícies sustentadoras tri-dimensionais, sendo a solução mais simples para escoamentos potenciais em asas finitas. Devido à sua simplicidade e alta velocidade de execução, é amplamente utilizado no projeto inicial de aeronaves, permitindo a análise de um grande número de configurações em um curto período de tempo ([Drela, 2014](#)).

Nesse método, as superfícies sustentadoras são discretizadas em uma malha de vórtices em formato de ferradura, como mostrado na [Figura 15](#). Cada vórtice possui uma

intensidade (Γ_n), calculada para todos os pontos de controle através da solução de um sistema de equações algébricas, considerando as devidas condições de contorno (Anderson, 2024). A partir da distribuição de Γ_n , é possível determinar as forças aerodinâmicas atuantes sobre a superfície.

Figura 15 – Representação esquemática dos vórtices em asas finitas.



Fonte: Adaptado de Anderson (2024).

2.4 Otimização de Projeto

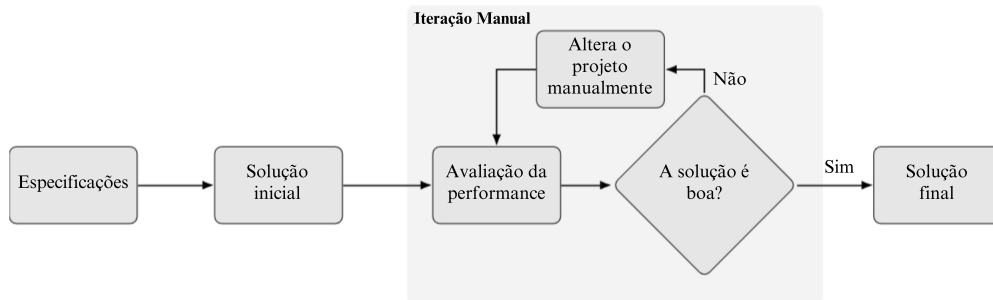
A otimização³ de projeto é uma ferramenta aplicada em substituição ao processo de projeto iterativo, com o objetivo de acelerar o ciclo de desenvolvimento e obter resultados mais eficientes (Martins; Ning, 2021). Em um projeto de engenharia tradicional, conduzido pelo método iterativo, cada etapa é baseada em escolhas humanas, nas quais especificações e requisitos são estabelecidos, e uma configuração inicial é considerada. Assim, inicia-se um ciclo no qual, a partir do desempenho da solução inicial, o projetista realiza ajustes embasados em sua experiência, buscando aprimorar a performance do projeto a cada iteração. Dessa forma, o projeto é finalizado quando o projetista considera que a performance da configuração final é satisfatória.

Ao aplicar uma ferramenta de otimização de projeto, a definição das especificações e requisitos, bem como a determinação da solução inicial, continuam sendo uma escolha humana. Nesta abordagem, o projetista deve realizar a modelagem do problema, sendo necessário definir variáveis de projeto a serem ajustadas, objetivos a serem maximizados ou minimizados, e restrições a serem satisfeitas. As decisões tomadas nessa etapa de modelagem exercem grande influência sobre a solução final, tornando a expertise do projetista fundamental. Entretanto, em uma abordagem de otimização, a avaliação do objetivo é realizada com base em valores numéricos tanto do objetivo quanto das restrições

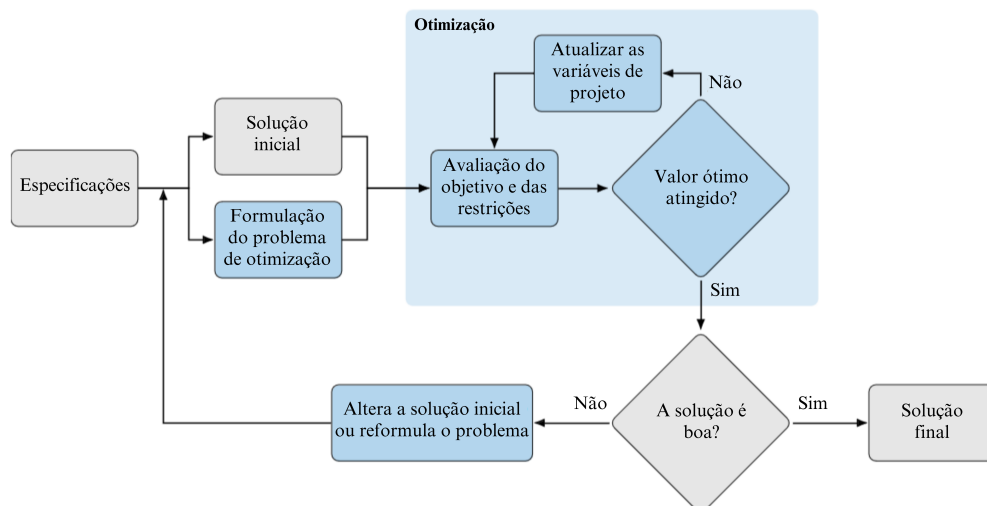
³ O conceito de otimização difere-se do de melhoria, pois o processo de otimização envolve uma abordagem matemática que busca o máximo ou o mínimo de uma função representativa do modelo matemático de um problema, considerando as restrições impostas.

estabelecidas (Martins; Ning, 2021). A Figura 16 apresenta um fluxograma comparando o ciclo de um projeto tradicional e de um projeto no qual se aplica uma ferramenta de otimização.

Figura 16 – Fluxograma do ciclo de um projeto realizado de duas maneiras distintas.



(a) Tradicionalmente.



(b) Com a aplicação de uma ferramenta de otimização de projeto.

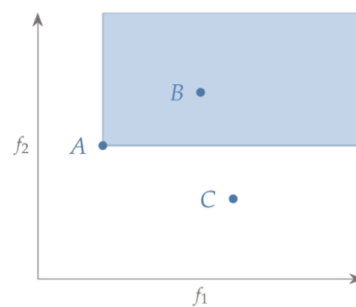
Fonte: Adaptado de Martins e Ning (2021).

2.4.1 Otimização Multiobjetivo

Muitos problemas de engenharia apresentam uma natureza mais complexa, não envolvendo a maximização ou minimização de apenas um único objetivo. Por exemplo, uma aeronave projetada para competições de AeroDesign pode ser modelada considerando dois objetivos principais: maximizar a quantidade de carga e minimizar o peso vazio da aeronave. Em casos como este de dois objetivos conflitantes, empregam-se as técnicas de otimização multiobjetivo. Esse tipo de abordagem auxilia na exploração do *trade-off* entre os objetivos, compreendendo como a melhoria de um pode impactar negativamente o outro.

Diante de múltiplos objetivos, é necessário reconsiderar o significado de um ponto de ótimo, tendo em vista a relação conflitante entre eles. Por esse motivo, trabalha-se com o conceito de Ótimo de Pareto (Martins; Ning, 2021). Esse conceito pode ser compreendido a partir da Figura 17, na qual se deseja minimizar ambos os objetivos f_1 e f_2 . Ao comparar a solução A com a solução B, observa-se que A é superior a B em ambos os objetivos. Assim, diz-se que B é dominada por A e que A é a melhor solução possível dentro da região em azul. Entretanto, ao comparar A com C, verifica-se que cada uma é melhor em um objetivo específico. Logo, nenhuma solução domina a outra.

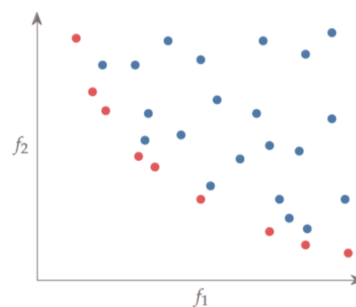
Figura 17 – Três soluções (A, B e C) representadas em função dos objetivos f_1 e f_2 .



Fonte: (Martins; Ning, 2021).

Quando uma solução não é dominada por nenhuma outra, como ocorre com A, ela é denominada como solução não dominada. Caso essa condição seja verdadeira em relação a todas as outras soluções possíveis, o ponto é chamado de Ótimo de Pareto. O conjunto de todos os Ótimos de Pareto, por sua vez, é denominado Fronteira de Pareto (Kochenderfer; Wheeler, 2019). A Figura 18 ilustra essa fronteira, composta por todas as soluções não dominadas destacadas em vermelho.

Figura 18 – Formação da Fronteira de Pareto pelas soluções não dominadas.



Fonte: (Martins; Ning, 2021).

A Fronteira de Pareto é uma ferramenta valiosa para compreender a relação entre dois objetivos. O ponto mais à direita da fronteira representa a maximização de f_1 , enquanto o ponto mais à esquerda representa a maximização de f_2 . Os pontos intermediários indicam quanto é necessário sacrificar de um dos objetivos para melhorar o valor do outro, tornando o *trade-off* visível e auxiliando no processo de decisão.

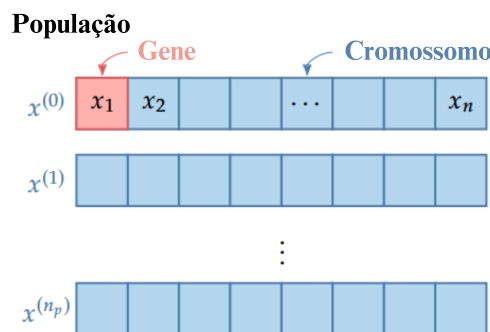
2.4.2 Algoritmos Genéticos

A resolução de problemas de otimização pode ser realizada por diferentes métodos. Dentre eles, destaca-se a família dos Algoritmos Evolutivos, que se baseia na geração de uma população inicial de soluções aleatórias, avaliadas por uma função objetivo. Essas soluções são selecionadas e combinadas por meio de operadores estocásticos, resultando em melhorias sucessivas a cada iteração. Graças à aleatoriedade desses operadores, raramente as soluções subsequentes serão semelhantes às anteriores (Mirjalili; Dong; Lewis, 2020).

Dentre os Algoritmos Evolutivos, o chamado Algoritmo Genético (AG) é especialmente conhecido e prestigiado. Inspirado na teoria da evolução proposta por Charles Darwin, o AG simula a sobrevivência do mais apto (Holland, 1975). Assim como na natureza, os indivíduos melhor adaptados têm maior probabilidade de sobreviver e transmitir seus genes às próximas gerações. Ao longo das gerações, genes mais adequados tendem a tornar as espécies mais adaptadas ao ambiente, tornando-se dominantes nas populações subsequentes (Mirjalili; Dong; Lewis, 2020).

Em um AG, cada iteração é denominada geração, composta por um conjunto de soluções chamado população. Cada solução é representada por um cromossomo, formado por genes que correspondem às variáveis de projeto, conforme ilustrado na Figura 19. Na rotina de otimização, inicia-se com uma população, e, a cada geração, as soluções passam por quatro operadores voltados a aprimorar os cromossomos: seleção, cruzamento, mutação e elitismo.

Figura 19 – Representação de uma população em um Algoritmo Genético.



Fonte: Adaptado de Martins e Ning (2021).

De acordo com Martins e Ning (2021), embora a maioria dos AGs siga aplicando os quatro operadores tradicionais, há grande flexibilidade em como esses passos são executados, resultando em diversas variações. Outro exemplo dessa flexibilidade está na representação das variáveis de projeto, que pode ser feita por meio de elementos binários ou de valores reais. A escolha de um método em detrimento do outro não altera os princípios fundamentais dos operadores, apenas detalhes de sua aplicação.

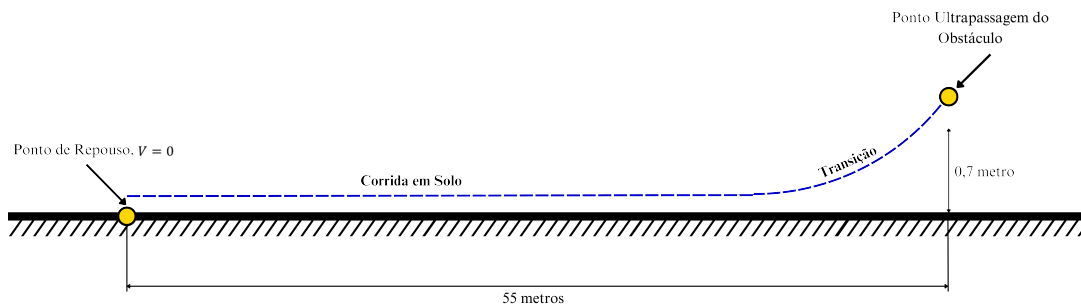
3 Materiais e Métodos

Durante o desenvolvimento do projeto, foi adotada uma metodologia estruturada em seis etapas, seguidas na ordem em que são apresentadas neste capítulo. Inicialmente, definiram-se os requisitos de projeto, passo fundamental para a elaboração da função objetivo do problema de otimização. Para compor a rotina de avaliação, estabeleceu-se a metodologia das análises aerodinâmicas, cujos resultados integram as avaliações de performance de decolagem da aeronave. Por fim, a integração de todas essas etapas permitiu o desenvolvimento do AG voltado à maximização da função objetivo.

3.1 Requisitos do Projeto

No contexto da Competição SAE BRASIL AeroDesign, considerando as edições recentes, os requisitos de missão comuns à Classe *Advanced* (excluindo-se a missão de sistemas embarcados, que varia anualmente) envolvem a realização de uma decolagem na qual a aeronave deve ultrapassar com segurança um obstáculo de 0,7 m de altura posicionado a 55 m do ponto de partida, conforme ilustrado na [Figura 20](#).

Figura 20 – Requisitos de decolagem para a Classe Advanced.



Fonte: Elaborado pelo autor.

Um dos critérios de pontuação da competição é a quantidade de carga paga que a aeronave consegue transportar durante a missão. Quanto maior a carga paga, maior a pontuação, de forma exponencial. Contudo, o regulamento impõe um valor de *MTOW* igual a 20 kg. Assim, um projeto competitivo deve buscar uma aeronave capaz de atingir ou superar este valor de *MTOW*, sendo ao mesmo tempo estruturalmente leve, a fim de maximizar a fração desse peso destinada à carga paga.

Pode-se concluir que o regulamento geral da competição incentiva o desenvolvimento de uma aeronave cargueira, cujo objetivo principal é transportar uma grande quantidade de carga atendendo aos requisitos de performance de decolagem estabelecidos. Dessa forma, os requisitos de missão adotados para o projeto a ser otimizado são:

- A aeronave deve possuir *MTOW* mínimo de 20 kg;
- A aeronave deve apresentar o menor peso vazio (*EW*) possível, para maximizar a fração de carga paga;
- Durante a decolagem, a aeronave deve ultrapassar o obstáculo de 0,7 m, posicionado a 55 m do ponto inicial de decolagem, com uma margem de segurança mínima de 0,1 m.

3.2 Elaboração da Função Objetivo

Com base nos requisitos estabelecidos para o projeto, identifica-se um problema de otimização com múltiplos objetivos, mais especificamente um problema bi-objetivo, no qual se deseja maximizar o *MTOW* e, simultaneamente, minimizar o *EW* da aeronave, respeitando as restrições estabelecidas para o problema.

O valor de *MTOW* de cada indivíduo é calculado por meio das rotinas de avaliações aerodinâmicas e de desempenho de decolagem. Já o valor de *EW* é estimado a partir da Equação 3.1, através da multiplicação da área da asa de um indivíduo por uma densidade por área alar (ρ_{asa}). Esse valor de densidade é igual a $1,5 \frac{kg}{m^2}$, e foi obtido pela média das razões entre o peso e a área de cinco asas previamente manufaturadas pela equipe Urutau AeroDesign para edições anteriores da competição SAE BRASIL.

$$EW = S \cdot \rho_{asa} \quad (3.1)$$

Desta forma, a função objetivo a ser maximizada para o problema estabelecido foi desenvolvida e apresentada na Equação 3.2. Nessa expressão, o termo *EW* é acompanhado do coeficiente 9 com o intuito de eliminar a diferença de ordem de grandeza entre o *MTOW* e o *EW*, garantindo maior relevância à minimização de *EW* no processo de maximização da função objetivo. Caso tal coeficiente não fosse aplicado, o *MTOW* (cujo valor costuma ser de nove a doze vezes maior que o *EW*) passaria a influenciar desproporcionalmente no resultado da função.

$$f(obj) = \frac{MTOW}{9 \cdot EW} \quad (3.2)$$

Segundo Eiben e Smith (2015), em problemas de otimização com restrições, penalidades são frequentemente utilizadas para aprimorar o desempenho do algoritmo de otimização. A penalização altera o valor original da função objetivo de cada indivíduo, de modo a bonificá-lo ou penalizá-lo conforme a sua localização no espaço de busca, ou conforme determinadas características de projeto.

No presente trabalho, a função objetivo está sujeita a três penalidades:

1. Penalidade estática relacionada à distribuição de sustentação: indivíduos que apresentam elevada distribuição de sustentação na ponta da asa recebem penalização, pois tendem a manifestar comportamento de estol indesejável. Essa condição é implementada por meio de uma função condicional, disposta na [Equação 3.3](#):

$$f(obj) = \begin{cases} f(obj) & \text{se o ponto máximo de sustentação está na raiz,} \\ f(obj) \cdot 0,2 & \text{se o ponto máximo de sustentação está na ponta.} \end{cases} \quad (3.3)$$

2. Bonificação por atingir *MTOW* próximo a 20 kg: como o valor idealizado pela competição é 20 kg, estabeleceu-se uma função de segundo grau, disposta na [Equação 3.4](#). Esta função aumenta o valor da função objetivo das soluções cujo *MTOW* está contido entre [18, 22] kg. O ápice da bonificação ocorre no valor de 20 kg.

$$f(obj) = \begin{cases} f(obj) + (-\frac{5}{2}MTOW^2 + 100MTOW - 990) & \text{se } 18 \leq MTOW \leq 22. \end{cases} \quad (3.4)$$

3. Penalização linear para *MTOW* acima de 22 kg: valores de *MTOW* superiores a 22 kg são penalizados de forma linear, conforme a função disposta na [Equação 3.5](#), buscando que evitar o aumento desta variável se sobressaia sobre a diminuição do *EW*.

$$f(obj) = \begin{cases} f(obj) + (-MTOW + 22) & \text{se } MTOW \geq 22. \end{cases} \quad (3.5)$$

3.3 Escolha das Variáveis de Projeto

Com o problema de otimização já modelado em uma função objetivo com as suas devidas bonificações e penalidades, a etapa seguinte consiste em definir quais variáveis de projeto são relevantes, com o intuito de maximizar esta função.

O *MTOW* de uma aeronave, componente principal da função objetivo, é determinado por cinco fatores: a geometria da asa, o aerofólio utilizado, o empuxo gerado pelo seu sistema propulsivo, o comprimento de pista disponível para a decolagem, e a altura do obstáculo que deve ser ultrapassado ao final da pista. Considerando que os dois últimos fatores são requisitos estabelecidos pelos regulamentos das competições SAE BRASIL AeroDesign, os demais fatores mencionados devem ser considerados como as variáveis de projeto a serem otimizadas de modo a atender às exigências determinadas.

3.3.1 Parâmetros Geométricos da Asa

A geometria da asa foi descrita no algoritmo de otimização por meio de sete variáveis de projeto representadas por números reais, cujos intervalos de variação estão apresentados na [Tabela 1](#). Esses limites foram estabelecidos com base nas limitações de fabricação apresentadas pela equipe Urutau AeroDesign.

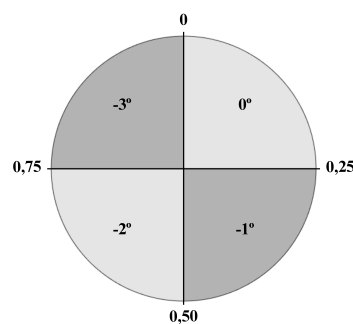
Tabela 1 – Limites dos parâmetros geométricos da asa.

Variável de Projeto	Limite Inferior	Limite Superior
Corda na raiz (m)	0,20	0,50
Razão de afilamento - λ	0,20	1,00
Envergadura (m)	2,00	3,60
Posição do afilamento	0,20	1,00
Recuo na ponta (m)	0,00	0,08
Torção intermediária	0	1
Torção na ponta	0	1

Fonte: Elaborado pelo autor.

Embora as duas variáveis relacionadas à torção sejam representadas por valores reais, sua natureza é essencialmente categórica. No caso da torção intermediária, o intervalo delimitado $[0, 1]$ é dividido em quatro subintervalos de igual tamanho. De tal forma que o valor real que representa a variável de projeto obrigatoriamente faz parte de um destes subintervalos, onde cada um define uma angulação específica dentre 0° , -1° , -2° ou -3° , de forma semelhante ao funcionamento de uma roleta, conforme ilustrado na [Figura 21](#).

Figura 21 – Roleta com a torção intermediária correspondente a cada intervalo.



Fonte: Elaborado pelo autor.

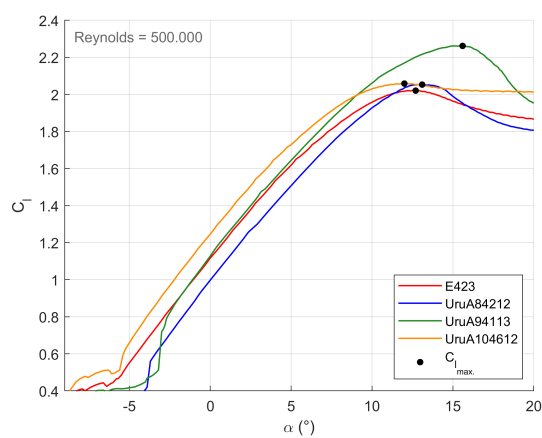
3.3.2 Aerofólios

A seleção dos aerofólios a serem considerados como variáveis de projeto buscou perfis capazes de gerar alta sustentação, visando melhorar o desempenho de decolagem. Foram escolhidos dois aerofólios tradicionais em aeronaves de AeroDesign, o Selig 1223 e o Eppler 423, como base para gerarem novos aerofólios através de uma rotina de

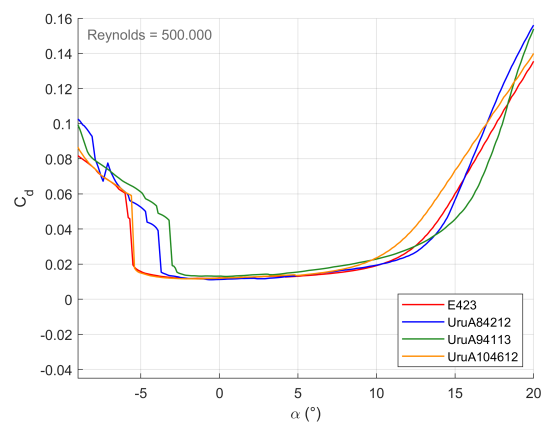
otimização para maximizar seu coeficiente de sustentação e minimizar seu coeficiente de arrasto, respeitando uma restrição relacionada ao coeficiente de momento. A ferramenta de otimização de aerofólios utilizada foi desenvolvida pela equipe Urutau AeroDesign e integra a parametrização de um perfil do solver *XFOIL* com uma aplicação de AG desenvolvida na linguagem *Python*[®] (Teixeira *et al.*, 2024).

Assim, foram selecionados quatro aerofólios como opções para o modelo de otimização. A Figura 22 apresenta as curvas aerodinâmicas desses aerofólios para um valor de Re compatível com a condição de decolagem.

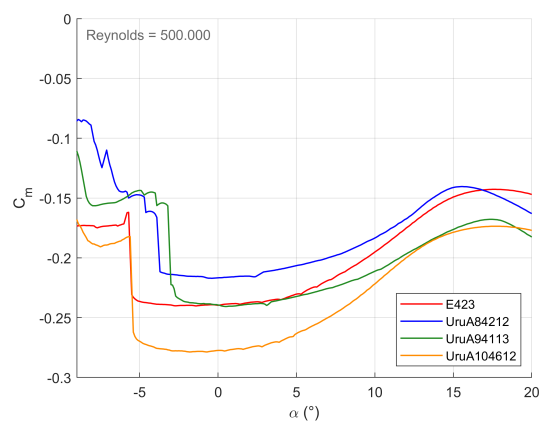
Figura 22 – Curvas aerodinâmicas dos aerofólios.



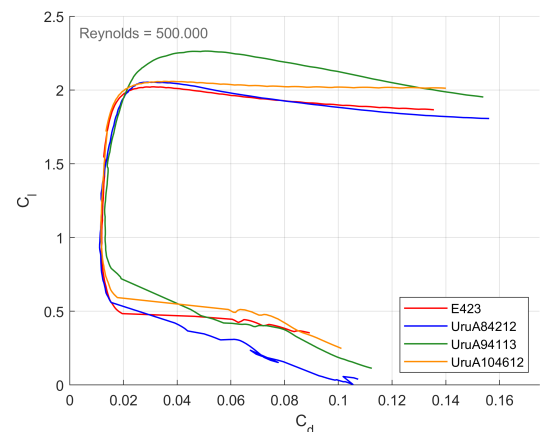
(a) Coeficiente de sustentação.



(b) Coeficiente de arrasto.



(c) Coeficiente de momento.



(d) Polar de arrasto.

Fonte: Elaborado pelo autor.

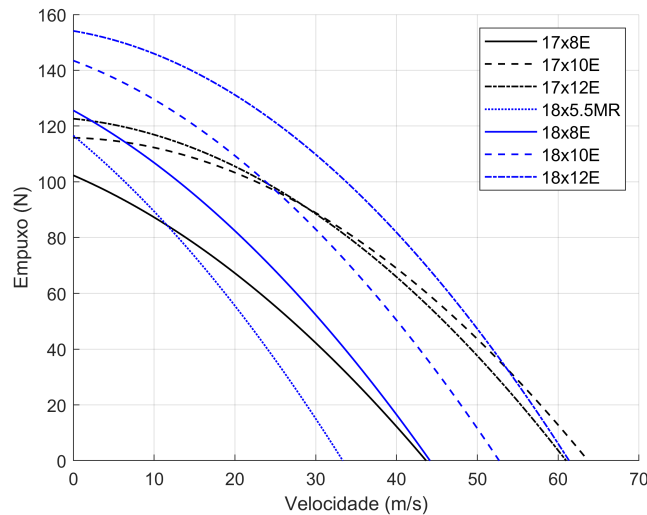
Para o algoritmo de otimização, a variável de projeto associada ao aerofólio, embora categórica, também foi representada por um número real no intervalo $[0, 1]$. Essa faixa de valores foi dividida em quatro subintervalos de igual tamanho, de modo que cada um corresponde a um aerofólio específico a ser selecionado.

3.3.3 Sistemas Propulsivos

A escolha dos sistemas propulsivos considerados como opção na rotina de otimização baseou-se no motor e nas hélices disponíveis no estoque da equipe Urutau AeroDesign. Adotou-se, portanto, o modelo de motor *Scorpion SII-4035-450kv*, em conjunto com sete diferentes modelos de hélices: 17x8E, 17x10E, 17x12E, 18x5.5MR, 18x8E, 18x10E e 18x12E. Cada opção foi representada através da sua curva de empuxo em função da velocidade.

Essas curvas foram obtidas a partir dos dados de desempenho fornecidos pelo fabricante das hélices (APC PROPELLERS, 2022). Para se utilizar dos resultados apresentados no banco de dados, assumiu-se como hipótese de aproximação que, independentemente da hélice, a rotação do motor permanece em um valor aproximado de 10.000 rpm. Assim, a partir das medições de empuxo e velocidade, obteve-se um polinômio quadrático que descreve o comportamento de cada sistema propulsivo, conforme ilustrado na Figura 23. As funções para cada sistema propulsivo são apresentadas na Tabela 2.

Figura 23 – Empuxo em função da velocidade dos sistemas propulsivos.



Fonte: Elaborado pelo autor.

Tabela 2 – Polinômios do empuxo em função da velocidade dos sistemas propulsivos.

Hélices	Polinômios Quadráticos
17x8E	$T(V) = -0,0249V^2 - 1,2568V + 102,3320$
17x10E	$T(V) = -0,0273V^2 - 0,07934V + 115,8218$
17x12E	$T(V) = -0,0282V^2 - 0,02911V + 122,6452$
18x5.5MR	$T(V) = -0,0332V^2 - 2,3898V + 116,6083$
18x8E	$T(V) = -0,0284V^2 - 1,5923V + 125,6174$
18x10E	$T(V) = -0,0311V^2 - 1,0840V + 143,4866$
18x12E	$T(V) = -0,0330V^2 - 0,4877V + 154,1342$

Fonte: Elaborado pelo autor.

A representação dos sistemas propulsivos no algoritmo seguiu a mesma lógica aplicada aos aerofólios. Entretanto, nesse caso, o intervalo $[0, 1]$ foi dividido em sete subintervalos de igual tamanho, cada um correspondendo a um grupo motopropulsor.

3.4 Avaliações Aerodinâmicas

As avaliações aerodinâmicas foram conduzidas no *software AVL* através de sua integração com *scripts* em *MATLAB*[®]. O objetivo central dessa análise foi determinar os coeficientes de sustentação e de arrasto induzido da aeronave, bem como a sua distribuição de sustentação na condição de decolagem. Enquanto os coeficientes aerodinâmicos obtidos viabilizam as avaliações de desempenho durante a decolagem, a distribuição de sustentação permite estimar a localização do início do estol da asa, servindo, assim, como parâmetro em uma das penalidades estabelecidas no projeto.

3.4.1 Criação do Modelo da Asa

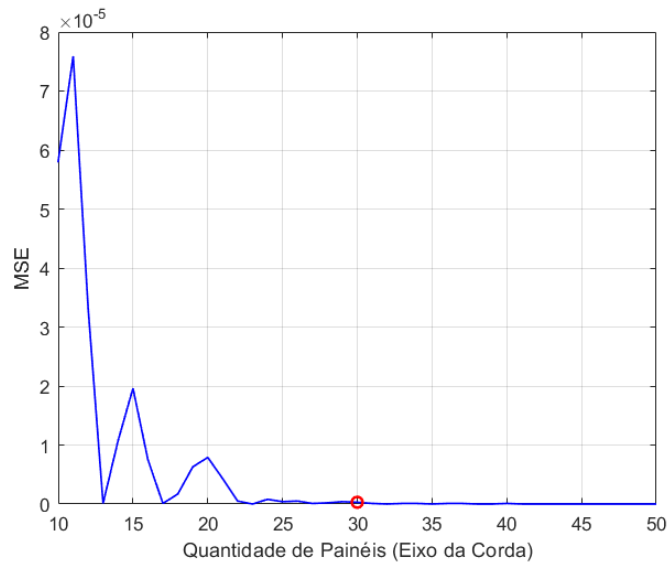
O arquivo de entrada da asa no *software AVL*, que determina a modelagem geométrica da asa, foi gerado por meio do algoritmo apresentado no [Apêndice A](#). Essa função recebe como entradas todas as variáveis de projeto e retorna o modelo da asa.

Buscando-se resultados acurados sem elevar excessivamente o custo computacional das avaliações aerodinâmicas, a quantidade e a disposição dos vórtices utilizados pelo *VLM* foram selecionadas com base em uma análise de convergência de malha. Nessa análise, avaliou-se o erro quadrático da aferição do coeficiente de sustentação de uma asa à medida que se aumentava a quantidade de elementos de malha, até o ponto em que a redução do erro não justificasse o incremento no custo computacional.

Para tal estudo, a quantidade de elementos no sentido da envergadura foi mantida em 20 vórtices por meia asa. Isso se justifica pelo fato de o interesse principal da análise recair sobre o efeito do arqueamento do perfil, sendo este o fator mais relevante para a obtenção de resultados fidedignos. Considerando este valor fixo, variou-se a quantidade de vórtices no eixo da corda e calculou-se o erro quadrático de cada observação, conforme representado na [Figura 24](#). Concluiu-se, então, que a utilização de 30 vórtices nessa direção propiciava um bom equilíbrio entre acurácia e custo computacional, resultando em um total de 1200 painéis para cada asa analisada.

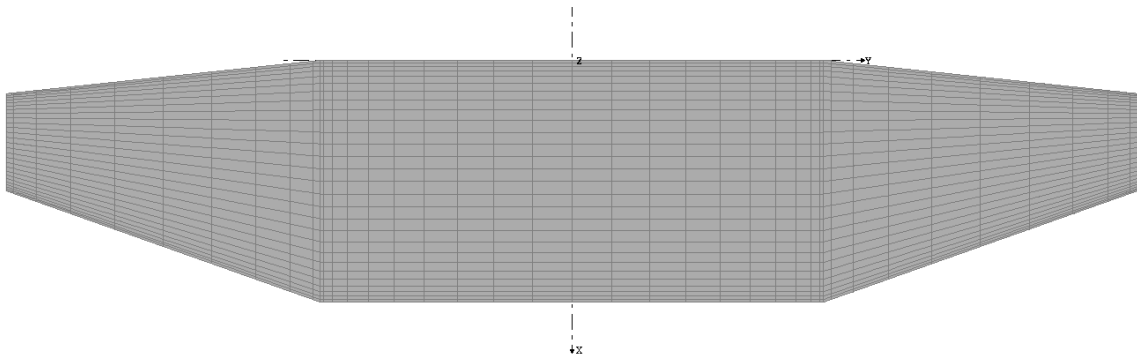
No que diz respeito à distribuição dos elementos, buscou-se uma maior concentração nas proximidades dos bordos de ataque e de fuga na direção da corda, utilizando a distribuição do tipo “*cosine*”. Já no eixo da envergadura, almejou-se uma concentração maior na interseção entre as seções e nas pontas da asa. Essa configuração foi obtida por meio da distribuição do tipo “*-sine*” na primeira seção, e do tipo “*cosine*” na segunda seção. A [Figura 25](#) ilustra um exemplo de uma asa que aplica essa disposição de painéis.

Figura 24 – Análise de convergência de malha.



Fonte: Elaborado pelo autor.

Figura 25 – Análise de convergência de malha.



Fonte: Elaborado pelo autor.

3.4.2 Execução das Avaliações Aerodinâmicas

A execução das análises aerodinâmicas é realizada por meio do algoritmo apresentado no [Apêndice B](#), o qual recebe todas as variáveis de projeto como entrada. Dentro desta função, são realizados dois passos. O primeiro consiste na geração do arquivo de entrada para o *software AVL*, e o segundo gera um arquivo que é utilizado para de fato executar as análises para obter os valores dos coeficientes aerodinâmicos e a distribuição de sustentação na asa. Dentro dessas análises, não é realizada nenhuma alteração no ângulo de ataque da aeronave, pois a simulação da performance de decolagem considera apenas a asa, desconsiderando o seu movimento de arfagem. Por fim, a rotina de avaliações aerodinâmicas apresenta como resultado o coeficiente de sustentação, o coeficiente de arrasto induzido e uma variável categórica que representa a posição inicial de estol esperada para determinada asa.

3.5 Avaliações de Performance de Decolagem

A simulação da performance de decolagem foi realizada por meio do algoritmo apresentado no [Apêndice C](#). Essa função recebe como entrada a massa total da aeronave, as variáveis de projeto e os coeficientes aerodinâmicos de sustentação e arrasto induzido obtidos para a configuração da asa analisada. Visando manter a rotina de cálculos enxuta e devido à dificuldade em determinar com precisão o coeficiente de arrasto parasita da aeronave, adotou-se um valor padrão de 0,016, obtido a partir da média de valores aferidos em aeronaves desenvolvidas para a competição SAE AeroDesign. Com base nesses parâmetros, a função simula tanto a corrida de decolagem quanto a transição necessária para ultrapassagem do obstáculo, determinando, assim, a distância total de decolagem.

Por fim, um segundo algoritmo, disposto no [Apêndice D](#), realiza a avaliação final da performance de decolagem, testando valores crescentes de massa para a mesma aeronave e verificando se os requisitos de comprimento de pista e ultrapassagem de obstáculo são atendidos. Esse procedimento permite determinar o *MTOW* da aeronave para que esta seja capaz de cumprir todos os requisitos de projeto.

3.6 Desenvolvimento do Algoritmo Genético

A aplicação de AGs na rotina de otimização do projeto foi desenvolvida em *MATLAB*[®], sem o uso de bibliotecas ou softwares de otimização externos. Essa abordagem foi adotada em virtude da maior flexibilidade do algoritmo autoral e do melhor aprofundamento no conhecimento do processo de otimização, o que possibilita um melhor entendimento em casos de eventuais divergências.

Neste trabalho, a população inicial do AG foi estabelecida com 30 indivíduos (cromossomos). Essa quantidade foi escolhida para não ampliar excessivamente a competição na etapa de seleção, preservando a participação de indivíduos com avaliações inferiores. Cada cromossomo é composto por nove genes, correspondentes às variáveis de projeto, todos representados por números reais e contínuos, de acordo com os intervalos definidos anteriormente. Uma vez gerada, a população inicial é avaliada por meio da função objetivo estabelecida.

Cada cromossomo é composto por 9 genes, correspondentes às variáveis de projeto. Todos os genes foram representados por números reais e contínuos, pertencentes aos intervalos apresentados anteriormente. Os cromossomos dessa população inicial são então avaliados por meio da função objetivo estabelecida.

O número de indivíduos na população permanece constante ao longo de todas as gerações. Seguindo a rotina de otimização, dentro de uma geração, para cada dois indivíduos selecionados, são gerados dois novos indivíduos através do operador de cruzamento, até

preencher completamente uma nova geração. Todos os novos indivíduos passam pelo operador de mutação e, em seguida, uma porção dos piores novos indivíduos são substituídos por uma porção dos melhores indivíduos da geração anterior através do operador elitismo. Assim, a nova geração é estabelecida e o ciclo se repete. O critério de parada do AG foi estabelecido pelo número máximo de 100 gerações, sendo este valor considerado suficiente para a avaliação do modelo.

3.6.1 Definição da População Inicial

O processo de otimização por AGs tem início com um conjunto de cromossomos gerados aleatoriamente. De acordo com [Mirjalili, Dong e Lewis \(2020\)](#), o principal objetivo na geração da população inicial é obter soluções aleatórias uniformemente distribuídas entre todas as variáveis. Esse objetivo foi alcançado por meio da aplicação da [Equação 3.6](#), na qual X_n é o n-ésimo gene de um cromossomo, r_n é um número aleatório no intervalo $[0, 1]$, L_{S_n} é o limite superior e L_{I_n} é o limite inferior para o n-ésimo gene.

$$X_n = (L_{S_n} - L_{I_n})r_n + L_{I_n} \quad (3.6)$$

3.6.2 Avaliação dos Indivíduos

A avaliação de cada indivíduo foi realizada por meio da função de *fitness*, apresentada no [Apêndice E](#). Essa função integra as rotinas de avaliação aerodinâmica e de performance de decolagem à função objetivo do problema, permitindo que cada indivíduo seja avaliado com base em seu objetivo, bonificações e penalidades, retornando, assim, o valor de *fitness* correspondente a cada indivíduo.

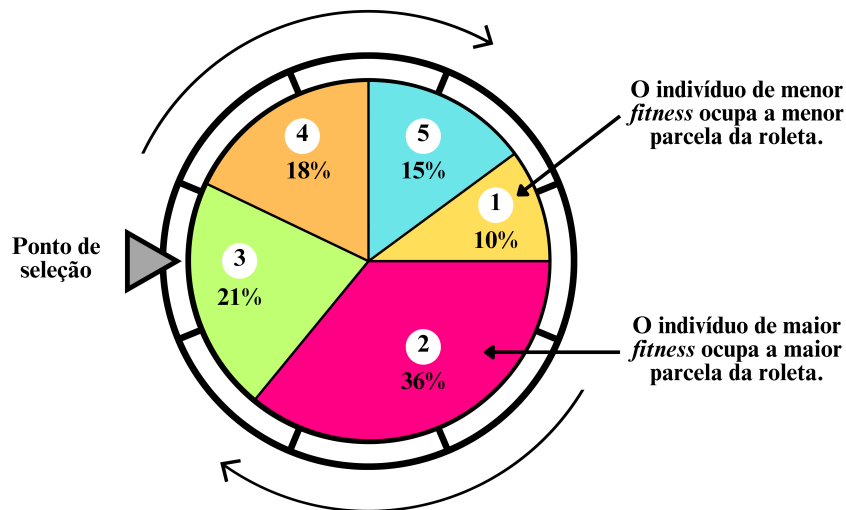
3.6.3 Operador de Seleção

O operador de seleção escolhido foi a Roda da Fortuna, que simula a seleção natural com base no princípio de que a probabilidade de sobrevivência e reprodução de um indivíduo aumenta proporcionalmente ao seu *fitness*.

A Roda da Fortuna normaliza e transforma os valores de *fitness* da população em probabilidades. Assim, ao gerar um número aleatório no intervalo $[0, 1]$, um dos indivíduos é selecionado ([Mirjalili; Dong; Lewis, 2020](#)). Desta forma, indivíduos com melhor avaliação ocupam uma porção maior da roda de seleção, sendo mais prováveis de serem escolhidos, conforme representado na [Figura 26](#). Contudo, indivíduos menos adaptados ainda podem ser escolhidos, mantendo-se a possibilidade de que genes menos favoráveis, quando combinados aos de indivíduos melhores, resultem em soluções de maior *fitness*. Essa abordagem preserva a variabilidade genética da população.

A implementação da seleção pela Roda da Fortuna está disposta no [Apêndice F](#).

Figura 26 – Representação do funcionamento da seleção pela Roda da Fortuna.



Fonte: Elaborado pelo autor.

3.6.4 Operador de Cruzamento

No operador de cruzamento, foi incluído o parâmetro denominado Probabilidade de Cruzamento (P_C) igual a 0,8. Esse valor indica a probabilidade de que dois pais selecionados efetivamente cruzem seus genes, sendo que este evento tem a probabilidade de 80% de ocorrer, enquanto nos outros 20% os pais selecionados são simplesmente copiados, isto é, clonados, para a próxima geração.

Em uma aplicação de AG com variáveis contínuas, o cruzamento combina os valores dos genes de dois pais, resultando em um gene filho. Um método simples para essa combinação é expresso pela Equação 3.7, apresentada em (Haupt; Haupt, 2004). Na qual X_n é n-ésimo gene de um novo indivíduo, β é um valor aleatório no intervalo $[0,1]$, P_{1_n} é o n-ésimo gene do primeiro pai, e P_{2_n} é o n-ésimo gene do segundo pai.

$$X_n = \beta P_{1_n} + (1 - \beta) P_{2_n} \quad (3.7)$$

Este método foi adaptado para a geração de dois filhos a partir de dois pais, simplesmente trocando a posição dos pais na equação para o segundo novo indivíduo. O procedimento completo do operador de cruzamento está exposto no Apêndice G.

3.6.5 Operador de Mutação

O operador de mutação visa ampliar o espaço de busca pela solução ótima, introduzindo alterações aleatórias nos genes dos indivíduos recém-gerados. Após o cruzamento, cada gene tem a probabilidade de 25% de sofrer uma mutação, valor este determinado a partir do parâmetro de Probabilidade de Mutação (P_M). Essa escolha busca evitar que as

soluções fiquem presas em ótimos locais. Além disso, embora seja relativamente elevada em comparação a outros estudos, é compensada pela aplicação do operador de elitismo. Quando a mutação ocorre, o valor do gene é substituído por um novo valor calculado da mesma forma que na geração da população inicial.

3.6.6 Operador de Elitismo

Após o operador de mutação, os indivíduos são novamente avaliados e partem para a última etapa da geração, o operador de elitismo. Segundo [Mirjalili, Dong e Lewis \(2020\)](#), os operadores de cruzamento e mutação podem, dependendo de seus parâmetros, levar à substituição completa da população, o que acarretaria a perda de boas soluções. Para prevenir isso, o elitismo garante que os 2 melhores indivíduos da geração anterior sejam mantidos na nova geração, substituindo os 2 piores indivíduos recém-gerados. Este número de indivíduos é definido a partir do parâmetro Preservação por Elitismo (P_E).

Após a aplicação do elitismo, a nova geração é formada e o ciclo se reinicia com a seleção de novos indivíduos para o cruzamento. A implementação final do AG, integrando todas as etapas, encontra-se no [Apêndice H](#), e os parâmetros utilizados no modelo estão dispostos na [Tabela 3](#).

Tabela 3 – Parâmetros do modelo de Algoritmos Genéticos.

Parâmetro	Valor
Número de genes	9
Número máximo de gerações	100
Tamanho da população	30
Probabilidade de cruzamento	0,80
Probabilidade de mutação	0,25
Preservação por elitismo	2

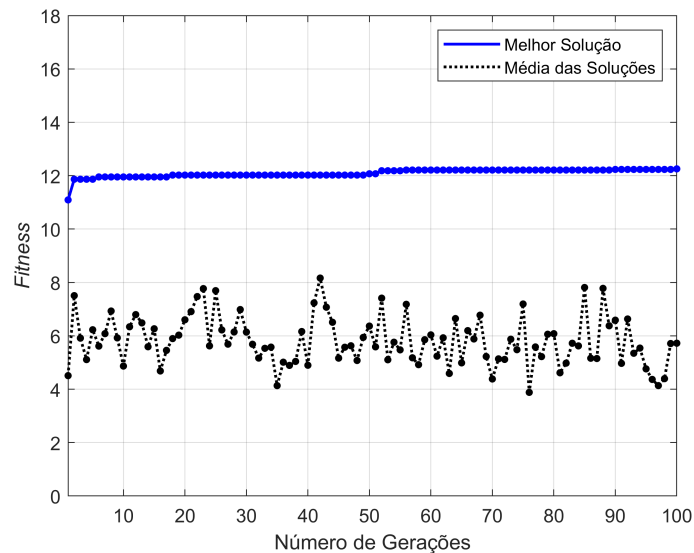
Fonte: Elaborado pelo autor.

4 Resultados e Discussão

4.1 Comportamento da Otimização Bi-objetivo

Os resultados deste trabalho foram obtidos após a análise de 100 gerações, com 30 indivíduos cada, totalizando 1 hora e 55 minutos de processamento¹. O processo de otimização ao longo das gerações está ilustrado na [Figura 27](#)², na qual a curva azul contínua representa o valor de *fitness* do melhor indivíduo por iteração, e a curva preta pontilhada indica a média do *fitness* em cada iteração.

Figura 27 – Comportamento do *fitness* dos indivíduos ao longo das gerações.



Fonte: Elaborado pelo autor.

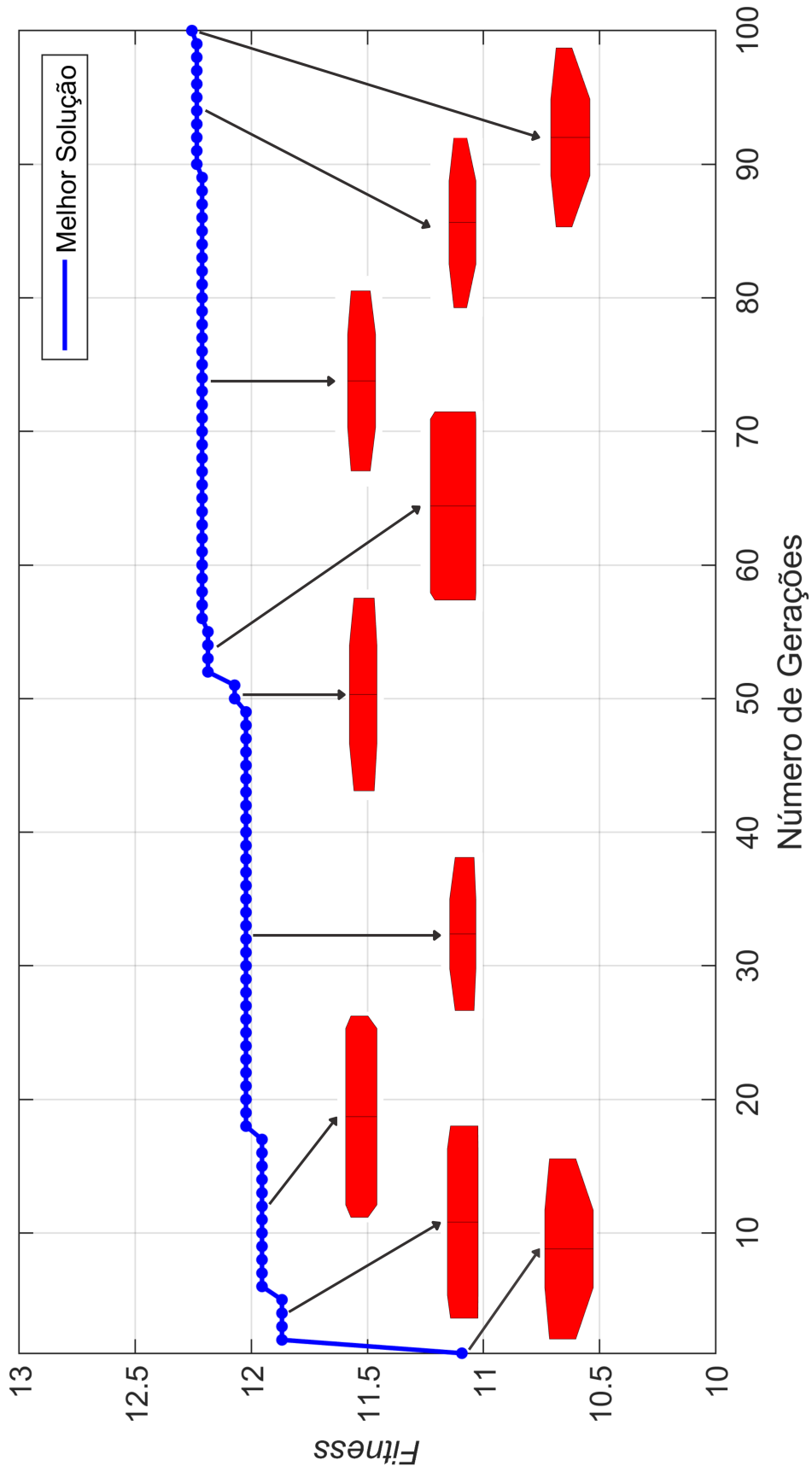
A partir do comportamento da curva azul, nota-se que a melhor solução mantém ou melhora seu valor de *fitness* ao longo das gerações. Esse fenômeno ocorre devido à aplicação do operador de elitismo, que, ao clonar os melhores indivíduos da geração anterior na nova geração, garante que as melhores soluções apenas evoluam. A curva dos valores médios de *fitness* por sua vez, apresenta um comportamento irregular ao longo das gerações, refletindo a alta variabilidade genética em cada geração. Esse resultado está associado ao elevado valor de P_M adotado, o que reduz a probabilidade de perda de diversidade.

Na [Figura 28](#), apresenta-se uma visualização ampliada da geometria de asa das melhores soluções conforme o algoritmo evolui.

¹ O processamento foi realizado em um *notebook* equipado com um processador Intel® Core™ i7-8550U, 16 GB de RAM e sistema operacional Windows 10.

² A evolução das soluções também foi disposta através de um GIF que apresenta a variação de todas as variáveis de projeto ao longo das gerações, disponível através do *QR Code* disposto no [Apêndice I](#).

Figura 28 – Geometria da asa dos melhores indivíduos ao longo das gerações.

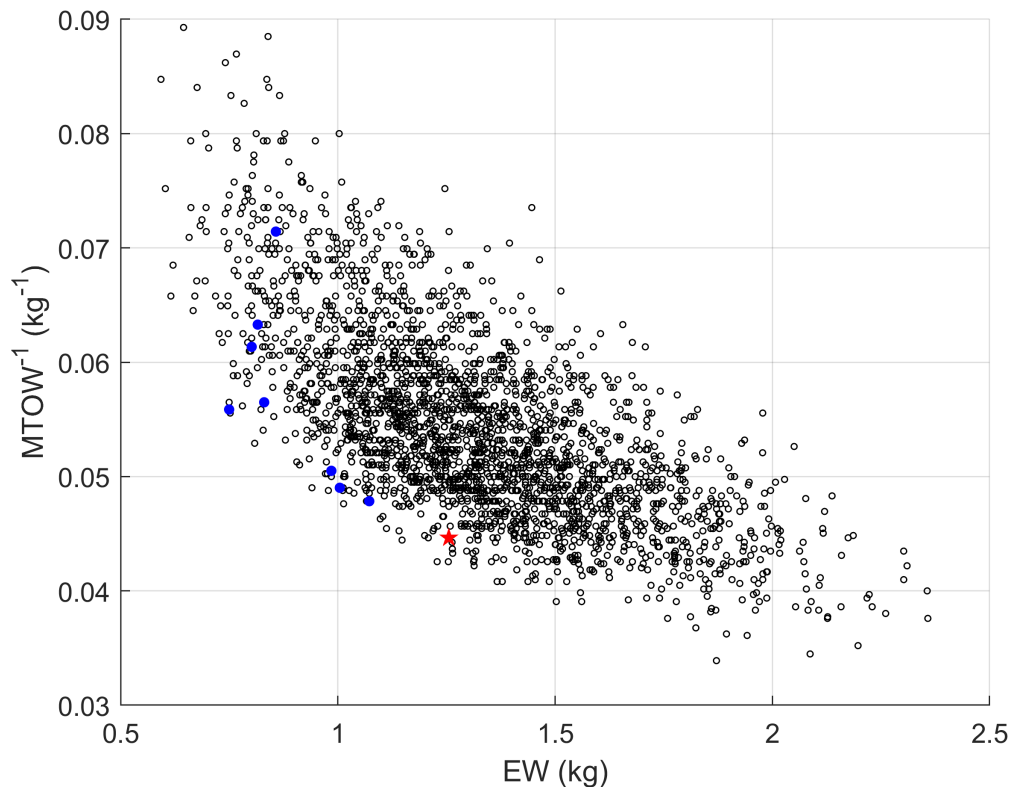


Fonte: Elaborado pelo autor.

Analisando a figura, observa-se uma grande variação nas soluções encontradas ao longo das gerações. Embora apresentem geometrias de asa bem distintas, essas soluções possuem elevados valores de *fitness*, demonstrando que diferentes abordagens de *trade-offs* durante o projeto aeronáutico podem convergir para o cumprimento de um mesmo objetivo.

Na Figura 29, encontram-se todas as soluções geradas. Para os gráficos apresentados a seguir, cada círculo preto vazado representa uma solução distinta, já os círculos azuis preenchidos diferenciam as 10 melhores soluções, enquanto a estrela vermelha preenchida indica a solução de maior *fitness* entre todas. A disposição dos indivíduos no gráfico de $MTOW^{-1}$ em função de EW tem como objetivo evidenciar o comportamento bi-objetivo desses valores que compõem a função objetivo, sendo possível a visualização da formação da Fronteira de Pareto.

Figura 29 – Comportamento bi-objetivo do problema de otimização.

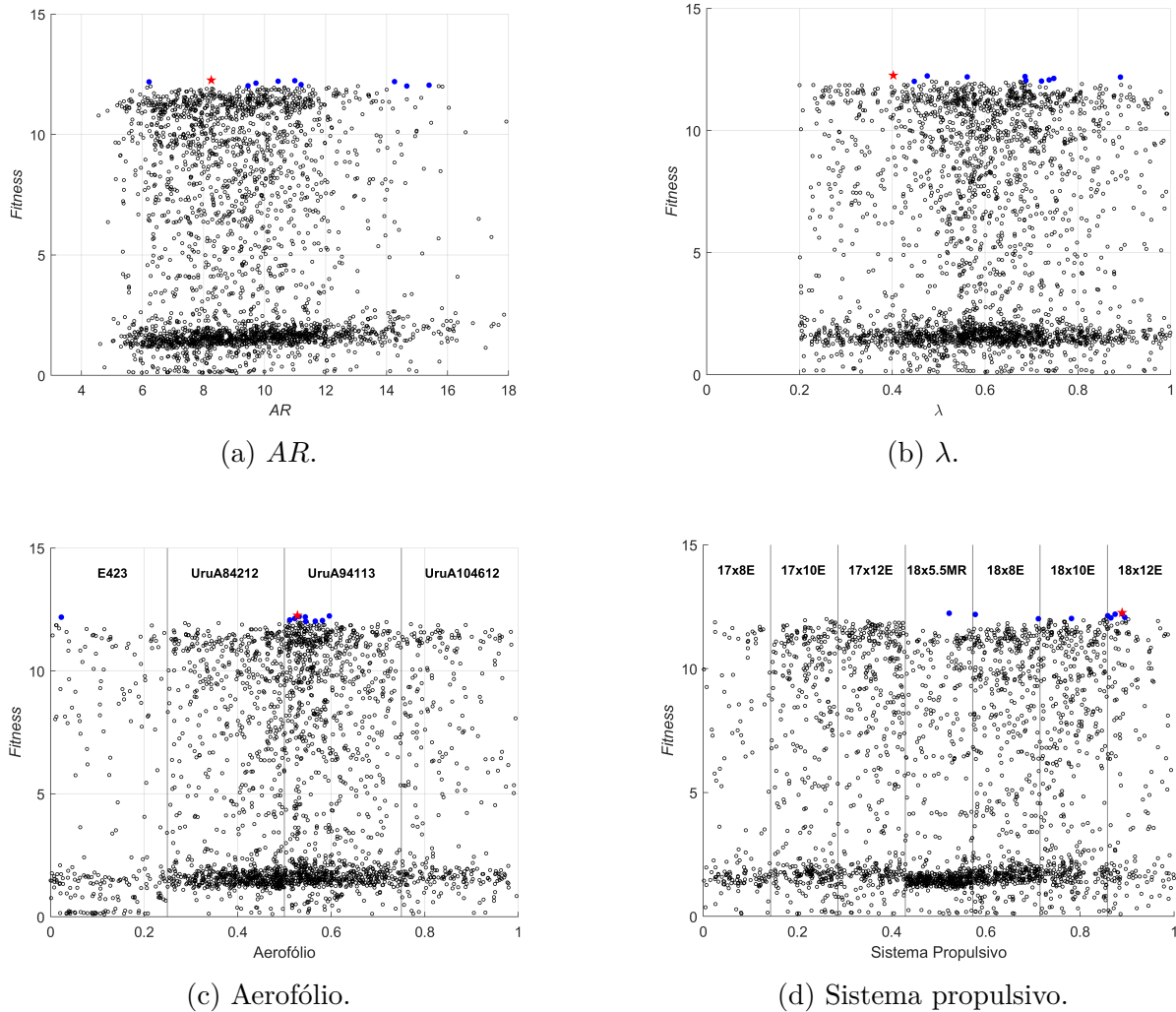


Fonte: Elaborado pelo autor.

Segundo Eiben e Smith (2015), ao se aplicar um Algoritmo Evolutivo em um problema de otimização multiobjetivo, é esperado que os indivíduos se distribuam uniformemente conforme se aproximam da Fronteira de Pareto, comportamento este confirmado nos resultados obtidos. No entanto, a avaliação das soluções não depende apenas dos valores de $MTOW$ e EW , pois as bonificações e penalidades impostas ao modelo reduzem o número de indivíduos efetivamente viáveis como solução final. Esse fato justifica a existência de pontos encontrados fora da região delimitada pelos melhores indivíduos.

Com o intuito de avaliar em maiores detalhes como as variáveis de projeto influenciam no *fitness* das soluções, elaborou-se a Figura 30. Em (a) e (b), os indivíduos são dispostos em função de seus valores de AR e λ , propriedades que refletem grande parte dos *trade-offs* essenciais em um projeto conceitual aerodinâmico. Em (c) e (d), as soluções são classificadas conforme o aerofólio e o sistema propulsivo adotados.

Figura 30 – Distribuição dos indivíduos em função de suas variáveis de projeto.



Elaborado pelo autor.

Analisando cada gráfico, em (a) destaca-se a concentração dos melhores indivíduos no intervalo $[6, 12]$ de AR , valores típicos de aeronaves cargueiras (Sadraey, 2024), assim como esperado para aeronaves do AeroDesign. Já em (b) percebe-se a tendência de obtenção de melhores indivíduos para valores de λ no intervalo $[0,4; 0,8]$. Essa faixa é coerente com os requisitos de missão e a penalidade quanto à distribuição de sustentação estabelecida, pois caracteriza asas que equilibram um baixo valor de arrasto induzido com boas características de estol. Normalmente, asas com esse valor de λ apresentam *washout* moderado (Gudmundsson, 2022), fato constatado diretamente no melhor indivíduo.

Em (c), constata-se que o aerofólio mais frequente entre as soluções de maior *fitness* é o UruA94113, que, conforme a [Figura 22](#), possui a maior capacidade de geração de sustentação dentre as opções, além de apresentar arrasto intermediário. Tal escolha contribui para maximizar a performance de decolagem e reduzir a área de asa. Por fim, em (d), nota-se uma distribuição equilibrada entre as soluções, pois todos os sistemas propulsivos geram empuxo suficiente para atender aos requisitos de projeto. Ainda assim, o sistema dos indivíduos de maior *fitness* é o conjunto do motor *Scorpion SII-4035-450kv* com a hélice 18x12E, sendo esse o que apresenta os maiores valores de empuxo na faixa de velocidade da corrida de decolagem, conforme apresentado na [Figura 23](#).

4.2 Parâmetros da Solução Ótima

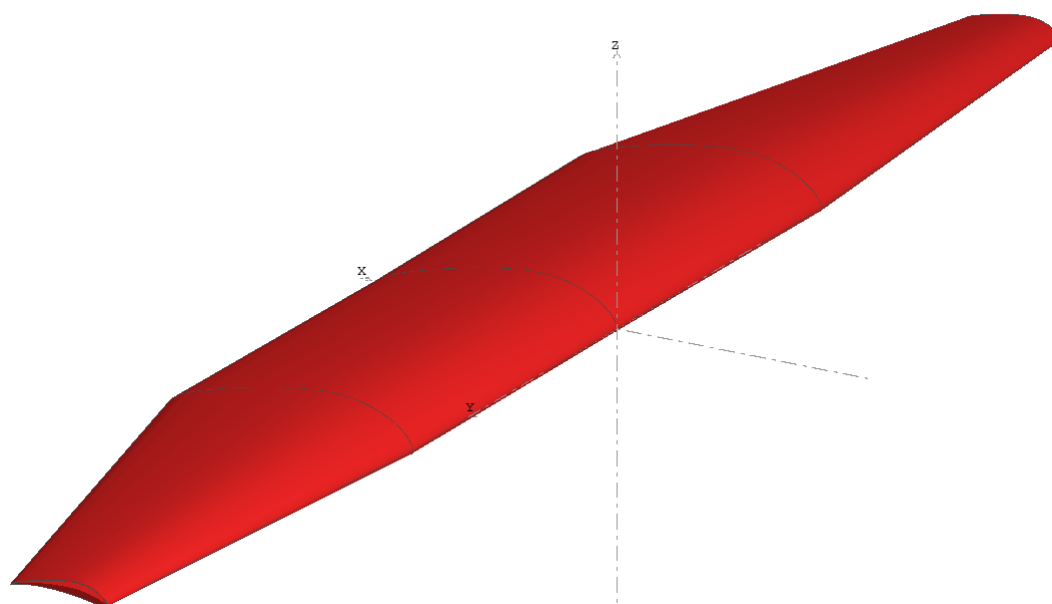
Os parâmetros do indivíduo com maior *fitness*, obtido no processo de otimização, estão apresentados na [Tabela 4](#), e sua modelagem tridimensional na [Figura 31](#). A [Figura 32](#) ilustra a simulação da decolagem desse indivíduo, comprovando sua capacidade de decolar e ultrapassar o obstáculo no seu *MTOW* de 22,4 kg.

Tabela 4 – Parâmetros do indivíduo de maior avaliação.

Parâmetro	Valor
Aerofólio	UruA94113
Motor	4035-450kv
Hélice	18x12E
Corda na raiz (<i>m</i>)	0,384
Razão de afilamento	0,402
Posição do afilamento (%)	0,429
Envergadura (<i>m</i>)	2,628
Meia-envergadura constante (<i>m</i>)	0,564
Meia-envergadura trapezoidal (<i>m</i>)	0,750
Recuo na ponta da asa (<i>m</i>)	0,053
Torção na seção intermediária (°)	-1,0
Torção na ponta da asa (°)	-1,0
Alongamento	8,26
Área da asa (<i>m</i> ²)	0,836
Peso vazio (<i>kg</i>)	1,255

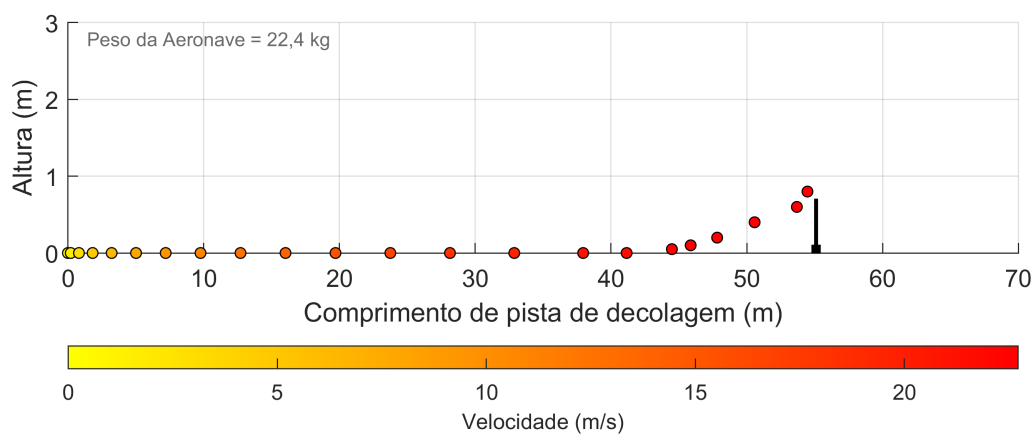
Fonte: Elaborado pelo autor.

Figura 31 – Modelagem tridimensional da solução ótima.



Fonte: Elaborado pelo autor.

Figura 32 – Simulação da decolagem da solução ótima em seu *MTOW*.



Fonte: Elaborado pelo autor.

5 Conclusão

O principal objetivo deste trabalho foi desenvolver um modelo de otimização bi-objetivo que aplique o método dos Algoritmos Genéticos para otimizar o projeto conceitual de uma aeronave de baixo Número de Reynolds, destinada à Competição SAE BRASIL AeroDesign, maximizando o seu peso máximo de decolagem e minimizando o seu peso vazio. Para isso, elaborou-se uma rotina de avaliação aerodinâmica por meio do *software Athena Vortex Lattice*, integrada a uma rotina de desempenho de decolagem capaz de simular a corrida de decolagem e a ultrapassagem de um obstáculo. Essas rotinas foram, então, acopladas a uma aplicação autoral de Algoritmos Genéticos desenvolvida em *MATLAB*[®].

A aplicação do modelo de otimização desenvolvido permitiu a avaliação de mais de 1500 soluções por hora. Esse desempenho eficiente deve-se, principalmente, ao refinamento da malha na análise aerodinâmica de cada indivíduo e à abordagem adotada para a simulação da corrida de decolagem. A análise das soluções, em função de seus valores de *MTOW* e *EW*, revelou a formação da Fronteira de Pareto, evidenciando a existência de múltiplas configurações que atendem aos requisitos de projeto estabelecidos e o caráter bi-objetivo do problema entre a maximização de um objetivo e a minimização do outro.

O indivíduo com a melhor avaliação em relação à função objetivo elaborada não se encontra na Fronteira de Pareto, devido às bonificações e penalidades aplicadas. Essas funções foram fundamentais para identificar a solução mais adequada aos objetivos estabelecidos pela competição SAE Brasil AeroDesign, maximizando a competitividade da aeronave projetada.

A solução ótima identificada neste trabalho apresenta uma geometria de asa comumente empregada em competições de AeroDesign. No entanto, suas dimensões foram refinadas pelo modelo para aprimorar a performance e aumentar a competitividade da aeronave. Essa configuração atende aos requisitos estabelecidos pela competição, destacando-se por seus parâmetros geométricos, como alongamento e razão de afilamento, típicos de aeronaves cargueiras.

Os algoritmos desenvolvidos neste trabalho foram estruturados de forma a possibilitar a integração de novas rotinas de avaliação e variáveis de projeto. Essas expansões têm o potencial de ampliar a capacidade do modelo e de aperfeiçoar ainda mais a etapa de projeto conceitual da equipe Urutau AeroDesign em aplicações futuras. Assim, diante dos objetivos definidos e com base nos resultados obtidos, conclui-se que o propósito do trabalho foi atingido.

5.1 Sugestões para Trabalhos Futuros

A seguir estão dispostas algumas sugestões para trabalhos futuros, visando a continuidade do algoritmo desenvolvido:

- Realização de ensaios de empuxo dinâmico dos sistemas propulsivos viáveis, permitindo determinar as curvas de empuxo em função da velocidade com maior acurácia, o que aumentaria a confiabilidade dos resultados de desempenho;
- Desenvolvimento de uma rotina de avaliação de estabilidade longitudinal estática que permita incluir a otimização da geometria do estabilizador horizontal, tornando as soluções geradas mais próximas do que se espera ao fim do projeto conceitual;
- Aprimoramento do módulo de performance ao incluir análises de outras fases de voo (subida, cruzeiro, descida e pouso), permitindo melhor controle quanto às características esperadas da aeronave.

A implementação dessas melhorias resultaria em um projeto conceitual mais preciso, contemplando uma parcela maior da configuração da aeronave e acelerando o processo de desenvolvimento.

Referências

- ANDERSON, J. D. **Fundamentals of Aerodynamics**. 7. ed. New York (NY): McGraw-Hill International, 2024. 1168 p. Citado 6 vezes nas páginas 12, 17, 19, 20, 21 e 27.
- APC PROPELLERS. **Performance Data Files**. 2022. Disponível em: <<https://www.apcprop.com/technical-information/performance-data/>>. Acesso em: 10 dez. 2024. Citado na página 36.
- BORTOLETE, F. A. **Algoritmo de Otimização Aerodinâmica de Asas Voltadas a uma Aeronave de Baixo Número de Reynolds**. Manaus (AM): Universidade do Estado do Amazonas, 2017. Citado na página 13.
- DIAS FILHO, C. A. M. *et al.* Preliminary low reynolds number wing sizing optimization using genetic algorithm. In: XII CONGRESSO NACIONAL DE ENGENHARIA MECANICA, 2024, Natal (RN). **Proceedings of the XII Congresso Nacional de Engenharia Mecânica - CONEM 2024**. Rio de Janeiro (RJ): ABCM, 2024. Citado 2 vezes nas páginas 12 e 24.
- DRELA, M. **Flight Vehicle Aerodynamics**. 1. ed. Cambridge (MA): MIT Press, 2014. 304 p. Citado na página 26.
- EIBEN, A. E.; SMITH, J. E. **Introduction to Evolutionary Computing**. 2. ed. Berlin (BE): Springer Berlin, 2015. 287 p. Citado 2 vezes nas páginas 32 e 45.
- FUJIKI, G. [**Aeronave da equipe Urutau AeroDesign em 2024**]. 2024. 1 fotografia, color., 5760 p x 3840 p. Citado na página 12.
- GUDMUNDSSON, S. **General Aviation Aircraft Design: Applied methods and procedures**. 2. ed. Oxford (OX): Butterworth-Heinemann, 2022. 934 p. Citado 3 vezes nas páginas 23, 25 e 46.
- HAUPT, R. L.; HAUPT, S. E. **Practical Genetic Algorithms**. 2. ed. New York (NY): Wiley-Interscience, 2004. 192 p. Citado na página 41.
- HOLLAND, J. H. **Adaptation in Natural and Artificial Systems: An introductory analysis with applications to biology control and artificial intelligence**. 1. ed. Ann Arbor (MI): The University of Michigan Press, 1975. 232 p. Citado na página 30.
- KOCHENDERFER, M. J.; WHEELER, T. A. **Algorithms for Optimization**. 1. ed. Cambridge (MA): MIT Press, 2019. 520 p. Citado na página 29.
- MARTINS, J. R. R. A.; NING, A. **Engineering Design Optimization**. 1. ed. Cambridge (MA): Cambridge University Press, 2021. 650 p. Citado 4 vezes nas páginas 27, 28, 29 e 30.
- MIRJALILI, S.; DONG, J. S.; LEWIS, A. **Nature-Inspired Optimizers: Theories, literature reviews and applications**. 1. ed. New York (NY): Springer, 2020. 238 p. Citado 3 vezes nas páginas 30, 40 e 42.

RAYMER, D. P. **Aircraft Design: A conceptual approach**. 7. ed. Washington D. C. (DC): American Institute of Aeronautics and Astronautics, 2024. 1110 p. Citado 7 vezes nas páginas 15, 16, 17, 18, 21, 25 e 26.

RIBEIRO, J. M. B. **Desenvolvimento de um Algoritmo de Otimização Multidisciplinar Baseado em Algoritmo Genético para uma Aeronave do Projeto SAE AeroDesign**. Teresina (PI): Instituto Federal de Educação, Ciência e Tecnologia do Piauí, 2023. Citado 2 vezes nas páginas 12 e 13.

SADRAEY, M. H. **Aircraft Performance: An engineering approach**. 2. ed. Boca Raton (FL): CRC Press, 2023. 692 p. Citado na página 25.

SADRAEY, M. H. **Aircraft Design: A systems engineering approach**. 2. ed. New York, (NY): Wiley, 2024. 864 p. Citado 7 vezes nas páginas 15, 16, 18, 21, 22, 23 e 46.

TEIXEIRA, S. S. *et al.* Genetic algorithm implementation for airfoil geometry optimization. In: XII CONGRESSO NACIONAL DE ENGENHARIA MECANICA, 2024, Natal (RN). **Proceedings of the XII Congresso Nacional de Engenharia Mecânica - CONEM 2024**. Rio de Janeiro (RJ): ABCM, 2024. Citado na página 35.

VILELA, R. L. S. M.; SILVA, E. C. Aerodesign aircraft wing optimization using genetic algorithm. In: 2019 IEEE SYMPOSIUM SERIES ON COMPUTATIONAL INTELLIGENCE (SSCI), 2019, Xiamen (CN). **Proceedings of the 2019 IEEE Symposium Series on Computational Intelligence (SSCI)**. Piscataway (NJ): IEEE, 2019. Citado na página 13.

Apêndices

APÊNDICE A – Criação do Arquivo da Asa

```

1  %% ----- CRIACAO DO ARQUIVO DA ASA ----- %%
2  function AVL_FILE(INPUT)
3
4  % DEFINICAO DOS INPUTS:
5  c1 = INPUT.design.chord_r;
6  c2 = INPUT.design.chord_r * INPUT.design.taper;
7  bt = INPUT.design.span;
8  b = INPUT.design.span/2;
9  b1 = b * INPUT.design.y_taper;
10 o = INPUT.design.offset_tip;
11 t1 = INPUT.design.twist_mid;
12 t2 = INPUT.design.twist_tip;
13 a = INPUT.design.airfoil;
14
15 % CALCULOS NECESSARIOS:
16 Sref = 2*((c1 * b1) + (((c1 + c2)*(b - b1))/2));
17 Cref = Sref / bt;
18
19 % CRIACAO DO ARQUIVO:
20 avl_file = fopen('Astra.avl', 'w');
21
22 fprintf(avl_file, 'Astra\n');
23 fprintf(avl_file, '0.0                | Mach\n');
24 fprintf(avl_file, '0      0      0 | iYsym iZsym Zsym\n');
25 fprintf(avl_file, '%.4f %.4f %.4f | Sref Cref Bref\n', Sref,
    Cref, bt);
26 fprintf(avl_file, '0.00  0.00  0.00 | Xref Yref Zref\n\n');
27
28 fprintf(avl_file, 'SURFACE\n');
29 fprintf(avl_file, 'Main Wing\n');
30 fprintf(avl_file, '30  1.0\n\n');
31
32 fprintf(avl_file, 'YDUPLICATE\n');
33 fprintf(avl_file, '0.0\n\n');
34

```

```
35 fprintf(avl_file, 'ANGLE\n');
36 fprintf(avl_file, '0.0\n\n');
37
38 fprintf(avl_file, 'SECTION\n');
39 fprintf(avl_file, '0.000 0.000 0.000 %.3f 0.0 10 -2.0\n', c1)
   ;
40 fprintf(avl_file, 'AFILE\n');
41 fprintf(avl_file, '%s\n\n', a);
42
43 fprintf(avl_file, 'SECTION\n');
44 fprintf(avl_file, '0.000 %.3f 0.000 %.3f %.1f 10 1.0\n', b1,
   c1, t1);
45 fprintf(avl_file, 'AFILE\n');
46 fprintf(avl_file, '%s\n\n', a);
47
48 fprintf(avl_file, 'SECTION\n');
49 fprintf(avl_file, '=%.3f %.3f 0.000 %.3f %.1f\n', o, b, c2, t2
   );
50 fprintf(avl_file, 'AFILE\n');
51 fprintf(avl_file, '%s', a);
52
53 fclose(avl_file);
54 end
```

APÊNDICE B – Avaliações Aerodinâmicas

```

1  %% ----- AVALIACOES AERODINAMICAS - AVL ----- %%
2  function [OUTPUT] = AVL_RUN (INPUT)
3
4  % ESCRITA DO ARQUIVO ".AVL":
5  AVL_FILE (INPUT);
6
7  % PUXA O ARQUIVO DO AEROFOLIO PARA A PASTA PRINCIPAL:
8  movefile (strcat('Airfoils\'', INPUT.design.airfoil));
9
10 % CRIACAO DO ARQUIVO ".RUN":
11 avl_run = fopen('Astra.run', 'w');
12 fprintf(avl_run, 'LOAD Astra.avl\n');
13 fprintf(avl_run, 'OPER\n');
14 fprintf(avl_run, 'X\n');
15 fprintf(avl_run, 'ST\n');
16 fprintf(avl_run, 'Astra_Coefficients.txt\n');
17 fprintf(avl_run, 'FS\n');
18 fprintf(avl_run, 'Astra_Distribution.txt\n\n');
19 fprintf(avl_run, 'QUIT\n');
20 fclose(avl_run);
21
22 % EXECUCAO DAS ANALISES AERODINAMICAS:
23 dos(strcat('AVL.exe', '<', 'Astra.run'));
24
25 % DEVOLVE O ARQUIVO DO AEROFOLIO PARA A SUA PASTA:
26 movefile(INPUT.design.airfoil, 'Airfoils\');
27
28 % LEITURA DO ARQUIVO ".TXT" E DEFINICAO DOS OUTPUTS:
29 read1 = fopen('Astra_Coefficients.txt');
30 OUTPUT.AVL.CL = cell2mat(textscan(read1, 'CLtot =%f',1, '
    HeaderLines', 23));
31 OUTPUT.AVL.CDi = cell2mat(textscan(read1, 'CDtot =%f',1));
32 fclose(read1);
33
34 % LEITURA DO ARQUIVO ".TXT" E DEFINICAO DOS OUTPUTS:

```

```
35 read2 = fopen('Astra_Distribution.txt');
36 data = textscan(read2, '%s', 'Delimiter', '\n');
37 fclose(read2);
38
39 cl_distribution = zeros(1,20);
40 for i = 21:40
41     local = textscan(data{1}{i}, '%f %f %f %f %f %f %f %f %f
42         %f %f %f %f');
43     cl_distribution(i-20) = local{8};
44 end
45 [M, I] = max(cl_distribution);
46 if I == 1
47     OUTPUT.AVL.Cl_Distribution = 'root';
48 else
49     OUTPUT.AVL.Cl_Distribution = 'tip';
50 end
51
52 % DELETA OS ARQUIVOS ".TXT", ".RUN" e ".AVL":
53 delete('Astra_Coefficients.txt');
54 delete('Astra_Distribution.txt');
55 delete('Astra.run');
56 delete('Astra.avl');
57 end
```

APÊNDICE C – Simulação da Decolagem

```

1  %% ----- SIMULACAO DA DECOLAGEM ----- %%
2  function [s_tot, OUTPUT] = TAKEOFF_SIMULATION (m, INPUT,
      OUTPUT)
3
4  % DECLARACAO DAS VARIAVEIS:
5  c1 = INPUT.design.chord_r;
6  c2 = INPUT.design.chord_r * INPUT.design.taper;
7  b = INPUT.design.span/2;
8  b1 = b * INPUT.design.y_taper;
9  Sref = 2*((c1 * b1) + (((c1 + c2)*(b - b1))/2));
10 OUTPUT.design.Sref = Sref;
11
12 g = INPUT.world.gravity;
13 rho = INPUT.world.density;
14 mu = INPUT.world.friction;
15 Ta = INPUT.design.power_plant(1);
16 Tb = INPUT.design.power_plant(2);
17 Tc = INPUT.design.power_plant(3);
18 CDO = INPUT.decisions.friction_drag;
19 h_o = INPUT.decisions.obstacle;
20
21 CL = OUTPUT.AVL.CL;
22 CDi = OUTPUT.AVL.CDi;
23
24 % DEFINICAO DAS VARIAVEIS IMPORTANTES PARA O LOOP:
25 v_x = 0; a_x = 0; % Velocidade-X em t=0.
26 vx_list = zeros(1,10); t_list = zeros(1,10);
27 i = 1; % Contador de Iteracoes.
28
29 % DEFINICOES DA FORCA PESO E DA SUSTENTACAO:
30 W = m * g; % Peso para determinado m
      .
31 L = 0; % Sustentacao em t=0.
32
33 % SIMULACAO DA CORRIDA EM SOLO:

```

```
34 for t = 0 : 0.05 : 30
35     if L <= W
36         v0_x = v_x;           % Velocidade-X em t-1.
37         v_x = v0_x + a_x * 0.05; % Velocidade-X em t.
38
39         q = 0.5 * rho * (v_x)^2; % Pressao dinamica em t.
40         T = (Ta * (v_x)^2) + (Tb * v_x) + Tc; % Tracao em t.
41         D = q * Sref * (CDO + CDi); % Arrasto em t.
42         L = q * Sref * CL;       % Sustentacao em t.
43         R = mu * (W - L);       % Atrito em t.
44
45         a_x = (T - D - R)/m;    % Aceleracao-X em t.
46
47         vx_list(i) = v_x;      % Armezenamento de V_x.
48         t_list(i) = t;        % Armezenamento de t.
49         i = i + 1;
50     end
51 end
52
53 % CALULO DA DISTANCIA PERCORRIDA NA CORRIDA EM SOLO:
54 s_g = trapz(t_list(1:end), vx_list(1:end));
55
56 % TRANSICAO ATE ULTRAPASSAR O OBSTACULO:
57 tr_r = 0.2156 * ((1.045*v_x)^2); % Raio de transicao.
58 s_t = sqrt(tr_r^2 - (tr_r-h_o)^2); % Distancia -
    ultrapassagem.
59
60 s_tot = s_g + s_t;           % Distancia total.
61 end
```

APÊNDICE D – Performance de Decolagem

```

1  %% ----- PERFORMANCE DE DECOLAGEM ----- %%
2  function [OUTPUT] = PERFORMANCE_ANALYSIS (INPUT)
3
4  % DEFINE OS INPUTS:
5  TO_dist = INPUT.decisions.TO_dist;
6
7  % REALIZA AS AVALIACOES AERODINAMICAS:
8  [OUTPUT] = AVL_RUN (INPUT);
9
10 % REALIZA A SIMULACAO DE DECOLAGEM COM DIFERENTES VALORES DE
    MASSA:
11 for m = 10 : 0.1 : 40
12     [s_tot, OUTPUT] = TAKEOFF_SIMULATION (m, INPUT, OUTPUT);
13     if s_tot > TO_dist
14         [s_tot] = TAKEOFF_SIMULATION ((m - 0.1), INPUT,
15             OUTPUT);
16         MTOW = m - 0.1;
17         break
18     end
19 end
20 % DEFINICAO DOS OUTPUTS RELACIONADOS A PERFORMANCE:
21 OUTPUT.performance.s_tot = s_tot;
22 OUTPUT.performance.MTOW = MTOW;
23 end

```

APÊNDICE E – Função *Fitness*

```

1  %% ----- FUNCAO FITNESS ----- %%
2  function [rank] = GA_FITNESS (INPUT, pop_size, population)
3
4  rank = zeros (pop_size, 1);
5
6  for j = 1 : pop_size
7      individual = population (j,:);
8
9      INPUT.design.chord_r = individual(1);
10     INPUT.design.taper = individual(2);
11     INPUT.design.span = individual(3);
12     INPUT.design.y_taper = individual(4);
13     INPUT.design.offset_tip = individual(5);
14
15     % CALCULO PARA A TORCAO INTERMEDIARIA:
16     twist_mid = [0.0 -1.0 -2.0 -3.0];
17     n = find(individual(6) <= [0.25 0.50 0.75 1.00], 1, '
18         first');
19     INPUT.design.twist_mid = twist_mid(n);
20
21     % CALCULO PARA A TORCAO NA PONTA:
22     twist_tip = [twist_mid(n) twist_mid(n)-1 twist_mid(n)-2
23         twist_mid(n)-3];
24     n = find(individual(7) <= [0.25 0.50 0.75 1.00], 1, '
25         first');
26     INPUT.design.twist_tip = twist_tip(n);
27
28     % CALCULO PARA AEROFOLIO:
29     airfoil = ["E423.dat" "UruA84212.dat" "UruA94113.dat" "
30         UruA104612.dat"];
31     n = find(individual(8) <= [0.25 0.50 0.75 1.0], 1, 'first
32         ');
33     INPUT.design.airfoil = airfoil(n);
34
35     % CALCULO PARA O SISTEMA PROPULSIVO:

```

```
31     GMP = [[-0.0249 -1.2568 102.3320]
32           [-0.0273 -0.07934 115.8218]
33           [-0.0282 -0.2911 122.6452]
34           [-0.0332 -2.3898 116.6083]
35           [-0.0284 -1.5923 125.6174]
36           [-0.0311 -1.0840 143.4866]
37           [-0.0330 -0.4877 154.1342]];
38     n = find(individual(9) <= [0.143 0.286 0.429 0.572 0.715
39           0.858 1.000], 1, 'first');
40     INPUT.design.power_plant = GMP(n,:);
41     % CALCULOS FINAIS:
42     [OUTPUT] = PERFORMANCE_ANALYSIS (INPUT);
43
44     MTOW = OUTPUT.performance.MTOW;
45
46     empty_w = INPUT.decisions.wing_density * OUTPUT.design.
47           Sref;
48
49     rank(j) = OUTPUT.performance.MTOW/(9*empty_w);
50
51     % PENALIDADES E BONIFICACOES:
52     if strcmp(OUTPUT.AVL.Cl_Distribution, 'tip') == 1
53         rank(j) = rank(j) * 0.2;
54     end
55
56     if MTOW >= 18 && MTOW <= 22
57         rank(j) = rank(j) + ((-5/2)*(MTOW^2) + (100)*MTOW -
58           (990));
59     end
60
61     if MTOW > 22
62         rank(j) = rank(j) + ((-1)*MTOW + 22);
63     end
64 end
end
```

APÊNDICE F – Roda da Fortuna

```
1 %% ----- SELECAO DOS PAIS - RODA DA FORTUNA ----- %%
2 function parent = GA_SELECTION (prob)
3
4 % GERA UM VALOR ALEATORIO ENTRE 0 E 1:
5 r = rand;
6
7 % CALCULA A PROBABILIDADE ACUMULADA DE CADA PAI SER
8   SELECIONADO:
9 c = cumsum(prob);
10
11 % SELECIONA O INDIVIDUO COM BASE NA PROBABILIDADE ACUMULADA:
12 parent = find(r <= c, 1, 'first');
```

APÊNDICE G – Operador de Cruzamento

```

1  %% ----- FUNCAO CROSSOVER ----- %%
2  function [c1, c2] = GA_CROSSOVER (prob, n_gens, population,
3     cons_max, ...
4     cons_min, prob_c)
5  % DEFINE SE O CROSSOVER IRA OCORRER OU NAO:
6  cross = rand;
7  % DEFINE A POSICAO DOS DOIS PAIS NA POPULACAO:
8  p1 = GA_SELECTION(prob);
9  p2 = GA_SELECTION(prob);
10
11 % CASO O CROSSOVER OCORRA:
12 if cross <= prob_c
13     % GERA A MATRIZ DE COEFICIENTES BETA ALEATORIOS:
14     beta = rand(1, n_gens);
15
16     % DEFINE OS DOIS FILHOS:
17     c1 = beta .* population(p1,:)+(1 - beta) .* population(p2
18         ,:);
19     c2 = beta .* population(p2,:)+(1 - beta) .* population(p1
20         ,:);
21
22     % VERIFICA SE OS FILHOS ULTRAPASSAM OS LIMITES
23     ESTABELECIDOS:
24     c1 = min(c1, cons_max);
25     c1 = max(c1, cons_min);
26     c2 = min(c2, cons_max);
27     c2 = max(c2, cons_min);
28
29 % CASO O CROSSOVER NAO OCORRA:
30 elseif cross > prob_c
31     % OS PAIS SAO PROPAGADOS PARA A PROXIMA GERACAO:
32     c1 = population(p1,:);
33     c2 = population(p2,:);
34 end

```

APÊNDICE H – Algoritmo Genético

```

1  %% ----- ALGORITMO GENETICO ----- %%
2  function [GENETIC_OUTPUTS] = GA_ELITISM(INPUT,
3      GENETIC_PARAMETERS)
4
5  cons = GENETIC_PARAMETERS.cons;
6  max_generations = GENETIC_PARAMETERS.max_generations;
7  pop_size = GENETIC_PARAMETERS.pop_size;
8  probab_crossover = GENETIC_PARAMETERS.probab_crossover;
9  probab_mutation = GENETIC_PARAMETERS.probab_mutation;
10 elitism_preservation = GENETIC_PARAMETERS.
11     elitism_preservation;
12
13 %% HIPERPARAMETROS DO MODELO
14
15 % DEFINE OS LIMITES PARA CADA VARIÁVEL:
16
17 % DEFINE OS PARAMETROS DO MODELO:
18 n_gens = length(cons);
19
20 % DEFINE LISTAS IMPORTANTES:
21 best_solutions = zeros(pop_size, n_gens);
22 best_fitnesses = zeros(1, max_generations);
23 mean_fitnesses = zeros(1, max_generations);
24 all_solutions = [];
25
26 %% POPULACAO INICIAL
27
28 % CRIA PRIMEIRA POPULACAO:
29 population = (repmat(cons_max, pop_size, 1) - repmat(cons_min
30     , pop_size, 1)) ...
31     .* rand(pop_size, n_gens) + repmat(cons_min, pop_size, 1)

```

```
32 % AVALIA A PRIMEIRA POPULACAO:
33 [fitness] = GA_FITNESS (INPUT, pop_size, population);
34
35 % ARMAZENA A MAIOR AVALIACAO E SUA POSICAO:
36 [best_fitness, i] = max(fitness);
37
38 % ARMAZENA O INDIVIDUO DE MAIOR AVALIACAO:
39 solution_indv = population(i,:);
40 solution_fitness = best_fitness;
41
42
43 for iter = 1 : max_generations
44     %% SELECAO DOS PAIS -> RODA DA FORTUNA
45
46     % DEFINE A PROBABILIDADE ACUMULADA DE CADA INDIVIDUO:
47     prob = zeros(pop_size, 1);
48
49     fitness_pad = fitness ./ (1+fitness);
50
51     for i = 1 : pop_size
52         prob(i) = fitness_pad(i) / sum(fitness_pad);
53     end
54
55
56     %% CRUZAMENTO
57
58     % DEFINE AS MATRIZES QUE IRAO COMPORTAR OS FILHOS:
59     c1 = zeros(pop_size/2, n_gens);
60     c2 = zeros(pop_size/2, n_gens);
61
62     % GERA A PROXIMA GERACAO:
63     for i = 1 : pop_size/2
64         [c1(i,:), c2(i,:)] = GA_CROSSOVER (prob, n_gens,
65             population, cons_max, cons_min, prob_crossover);
66     end
67
68     % ARMAZENA OS INDIVIDUOS GERADOS PELO CROSSOVER:
69     population_crossover = [c1; c2];
```

```
70
71     %% MUTACAO
72
73     % ARMAZENA OS INDIVIDUOS GERADOS APOS AS MUTACOES:
74     population_mutation = GA_MUTATION (pop_size, n_gens,
75         probab_mutation, population_crossover, cons_max,
76         cons_min);
77
78     % ARMAZENA AS AVALIACOES DOS INDIVIDUOS GERADOS APOS AS
79     MUTACOES:
80
81     [fitness_mutation] = GA_FITNESS (INPUT, pop_size,
82         population_crossover);
83
84     %% METODO DO ELITISMO
85
86     % REALIZA O ORDENAMENTO DA POPULACAO DA NOVA GERACAO:
87     [fitness_mutation, sort_order] = sort(fitness_mutation, '
88         ascend');
89     population_mutation = population_mutation(sort_order, :);
90
91     % REALIZA O ORDENAMENTO DA POPULACAO:
92     [fitness, sort_order] = sort(fitness, 'descend');
93     population = population(sort_order, :);
94
95     % TROCA OS PIORES INDIVIDUOS DA NOVA GERACAO PELOS
96     MELHORES DA ANTIGA:
97     for i = 1 : elitism_preservation
98         population_mutation(i,:) = population(i,:);
99         fitness_mutation(i,:) = fitness(i,:);
100     end
101
102     % SELECIONA OS INDIVIDUOS PARA A PROXIMA GERACAO:
103     population = population_mutation;
104     fitness = fitness_mutation;
105
106     % REALIZA O ORDENAMENTO FINAL PARA A NOVA GERACAO:
107     [fitness, sort_order] = sort(fitness, 'descend');
108     population = population(sort_order, :);
109
```

```
103
104     %% RESULTADOS
105
106     % ARMAZENA O MELHOR INDIVIDUO:
107     best_indv = population(1,:);
108     % ARMAZENA O FITNESS DO MELHOR INDIVIDUO:
109     best_fitness = fitness(1,:);
110     % ARMAZENA O FITNESS MEDIO DA GERACAO:
111     mean_fitness = sum(fitness(:,:))/pop_size;
112     % ARMAZENA TODOS OS INDIVIDUOS
113     all_solutions = [all_solutions; population fitness];
114
115     % ARMAZENA ESTAS INFORMACOES EM LISTAS:
116     best_solutions(iter,:) = best_indv;
117     best_fitnesses(iter) = best_fitness;
118     mean_fitnesses(iter) = mean_fitness;
119     std_fitnesses(iter) = std_fitness;
120
121     % SUBSTITUI A SOLUCAO DO PROBLEMA, CASO NECESSARIO:
122     if best_fitness > solution_fitness
123         solution_fitness = best_fitness;
124         solution_indv = best_indv;
125     end
126 end
127
128 % ARMAZENA AS SAIDAS DA FUNCAO EM UM STRUCT:
129 GENETIC_OUTPUTS.best_fitnesses = best_fitnesses;
130 GENETIC_OUTPUTS.mean_fitnesses = mean_fitnesses;
131 GENETIC_OUTPUTS.best_solutions = best_solutions;
132 GENETIC_OUTPUTS.solution_fitness = solution_fitness;
133 GENETIC_OUTPUTS.solution_indv = solution_indv;
134 GENETIC_OUTPUTS.all_solutions = all_solutions;
135 end
```

APÊNDICE I – GIF da Evolução

