



UNIVERSIDADE FEDERAL DE SANTA CATARINA
DEPARTAMENTO DE INFORMÁTICA E ESTATÍSTICA
BACHARELADO EM CIÊNCIAS DA COMPUTAÇÃO

Paulo Barbato Fogaça de Almeida

**Otimização Paramétrica de Hélices de Drone Híbrido para Desempenho na Água
por Meio de Algoritmos Evolutivos**

Florianópolis
2023

Paulo Barbato Fogaça de Almeida

**Otimização Paramétrica de Hélices de Drone Híbrido para Desempenho na Água
por Meio de Algoritmos Evolutivos**

Trabalho de Conclusão de Curso apresentado à Universidade Federal de Santa Catarina – UFSC como requisito parcial para obtenção do título de bacharel em Ciências da Computação

Orientador: Prof. Jônata Tyska Carvalho, Dr.

Florianópolis
2023

Ficha de identificação da obra elaborada pelo autor,
através do Programa de Geração Automática da Biblioteca Universitária da UFSC.

Almeida, Paulo Barbato Fogaça de
Otimização Paramétrica de Hélices de Drone Híbrido para
Desempenho na Água por Meio de Algoritmos Evolutivos /
Paulo Barbato Fogaça de Almeida ; orientador, Jônata Tyska
Carvalho, 2023.
92 p.

Trabalho de Conclusão de Curso (graduação) -
Universidade Federal de Santa Catarina, Centro Tecnológico,
Graduação em Ciências da Computação, Florianópolis, 2023.

Inclui referências.

1. Ciências da Computação. 2. Drones híbridos. 3.
Algoritmos evolutivos. 4. Otimização de hélices de drone.
5. Estratégias de Evolução. I. Carvalho, Jônata Tyska. II.
Universidade Federal de Santa Catarina. Graduação em
Ciências da Computação. III. Título.

Paulo Barbato Fogaça de Almeida

Otimização Paramétrica de Hélices de Drone Híbrido para Desempenho na Água por Meio de Algoritmos Evolutivos

O presente trabalho em nível de Graduação foi avaliado e aprovado por banca examinadora composta pelos seguintes membros:

Prof. Jônata Tyska Carvalho, Dr.
Universidade Federal de Santa Catarina

Prof. Rafael de Santiago, Dr.
Universidade Federal de Santa Catarina

Prof. Álvaro Junio Pereira Franco, Dr.
Universidade Federal de Santa Catarina

Certificamos que esta é a **versão original e final** do trabalho de conclusão que foi julgado adequado para obtenção do título de bacharel em Ciências da Computação.

Florianópolis, 27 de Novembro de 2023.

Coordenação do Curso de Graduação em
Ciências da Computação

Prof. Jônata Tyska Carvalho, Dr.
Orientador

Florianópolis, 2023.

Dedico este trabalho ao meu querido amigo Vitor Della.
Nas horas de incerteza, sua presença constante e apoio
incansável iluminou meu caminho. Cada palavra sua foi
um raio de luz nos dias mais desafiadores.

AGRADECIMENTOS

Agradeço a minha família, especialmente meus pais que me deram todo o amparo e suporte com almoços no Giraffas, purezinhas, chocolates e computadores.

Aos professores que presentes na minha formação, principalmente o professor Jônata Tyska por estar sempre disponível para me orientar.

Ao CNPq (Conselho Nacional de Desenvolvimento Científico e Tecnológico), pela bolsa de estudos que possibilitou a dedicação a este trabalho.

A todos os meus amigos, que estiveram presentes no meu dia a dia e me acompanharam em conversas, passeios, festas, praias e jogos.

Ao Robson e Wesley que viraram incontáveis noites comigo fazendo trabalhos, e durante o dia acataram minhas ideias mais malucas.

Ao Pedro por compartilhar séries, aniversários, aulas, festas, caronas, reclamações, risadas e tudo mais.

Ao Cross, por me permitir fazer incontáveis projetos com sua imagem.

Ao Teo pelos eventos juntos, almoços, conversas de séries e jogos.

Ao Klaus, Nicolas, Favarin e Zinho pela presença do grupo do Churros.

A Laura por me acompanhar em quase todas as festas.

Ao Tiger (Leo), Maru e Teo novamente que foram em eventos de cosplay comigo.

A Rebeca, John, Manel, Thomas, Samuel, Lourenço e Gibran que me carregaram nas vitórias e derrotas nos jogos e me fizeram companhia mesmo que a distância.

Ao Ohanez e Helena que sempre estiveram preocupados com a minha sanidade mental e me acompanharam em tudo que conseguiram.

Ao Tuca, Bruno e Nicolas por estarem presentes.

A Andrielly, por esperar os ônibus comigo de noite.

A Nati minha primeira amiga de química.

A Ana, Nati e Carol por todo o suporte e eventos não tradicionais.

A todos meus amigos do grupo da Positividade, mesmo que nossas amizades só tenham se fortalecido no final da faculdade eu sempre apreciei a presença de vocês.

Aos meus amigos de química, por me fazerem gostar de esperar para ter aula.

Aos membros da Apex Rocketry, que participaram do lançamento do foguete gravatá e me proporcionaram momentos inesquecíveis.

Por fim, agradeço a todos que me acompanharam durante minha jornada, saibam que vocês fizeram parte de quem eu sou hoje e portanto parte da criação deste trabalho.

RESUMO

Um dos componentes cruciais no desempenho do drone é a presença de hélices eficientes, uma vez que estas são responsáveis por converter energia em movimento. O processo de melhora do *design* da hélice, em busca de uma maior eficiência energética, envolve a geração e teste de diferentes conjuntos de parâmetros da morfologia para encontrar a configuração que diminua a energia necessária do motor para mover o drone. Esse procedimento é igualmente aplicado para melhorar hélices de drones híbridos, capazes de operar tanto no ar quanto na água, utilizando as mesmas hélices para propulsão em ambos os meios. Este estudo concentra-se na descrição da morfologia através de parâmetros contínuos e discretos, buscando aprimorar esses elementos para melhorar a configuração da hélice, proporcionando um menor consumo de energia do motor. Assim, este trabalho compara a utilização de Algoritmos Evolutivos de Estratégias de Evolução, CMA-ES e OpenAI-ES, comparando-os com algoritmos de busca local e busca aleatória por meio de experimentos, a fim de avaliar a eficiência energética das hélices em diferentes velocidades de serviço. Os resultados obtidos demonstram que os Algoritmos Evolutivos encontram resultados válidos para velocidades onde abordagens anteriores não haviam alcançado, e nos casos onde já existiam resultados, as soluções obtidas apresentaram desempenho superior.

Palavras-chave: Drones híbridos. Otimização de hélices de drone. Algoritmos evolutivos. CMA-ES. OpenAI-ES.

ABSTRACT

One of the crucial components in the drone's performance is the presence of efficient propellers, as they are responsible for converting energy into motion. The process of enhancing the propeller design, aiming for greater energy efficiency, involves generating and testing different sets of morphology parameters to find the configuration that reduces the motor's energy consumption to move the drone. This procedure is equally applied to improve propellers for hybrid drones, capable of operating in both air and water, using the same propellers for propulsion in both mediums. This study focuses on describing morphology through continuous and discrete parameters, aiming to enhance these elements to improve the propeller configuration, providing lower motor energy consumption. Thus, this work compares the use of Evolutionary Algorithms, specifically Evolution Strategies, CMA-ES and OpenAI-ES, with local search algorithms and random search through experiments to evaluate the energy efficiency of the propellers at different service speeds. The results obtained demonstrate that Evolutionary Algorithms find valid results for speeds where previous approaches had not reached, and in cases where results already existed, the solutions obtained showed superior performance.

Keywords: Hybrid drones. Drone Propeller optimization. Evolutionary algorithms. CMA-ES. OpenAI-ES.

LISTA DE FIGURAS

| | |
|--|----|
| Figura 1 – Influência dos Parâmetros na Geometria da hélice | 18 |
| Figura 2 – Visão geral do funcionamento de um Algoritmo Evolutivo. | 22 |
| Figura 3 – Representação do ajuste da CMA. | 25 |
| Figura 4 – Representação do funcionamento do CMA-ES. | 26 |
| Figura 5 – Diferença entre mínimo local e global. | 28 |
| Figura 6 – Diagrama de Fluxo de Execução das Heurísticas. | 37 |
| Figura 7 – Estudo do Tamanho da População e Máximo de Gerações | 42 |
| Figura 8 – P_B encontrados para os conjuntos de parâmetros testados. | 45 |
| Figura 9 – P_B encontrados para os conjuntos de parâmetros testados. | 46 |
| Figura 10 – Valores encontrados pelas execuções. | 46 |
| Figura 11 – Valores encontrados pelas execuções. | 48 |
| Figura 12 – Melhor Fitness encontrada por geração. | 50 |
| Figura 13 – Número de configurações válidas diferentes por geração. | 51 |
| Figura 14 – Linhas contínuas representam número de configurações válidas diferentes por geração, enquanto que as linhas tracejadas representam as válidas podendo ser repetidas por geração. | 52 |
| Figura 15 – Representações da Geometria 3D da Hélice. | 53 |

LISTA DE TABELAS

| | |
|--|----|
| Tabela 1 – Parâmetros para a geração da hélice. | 17 |
| Tabela 2 – Conteúdo dos Trabalhos Relacionados. | 32 |
| Tabela 3 – Parâmetros de entrada constantes. | 34 |
| Tabela 4 – Parâmetros de entrada no DE | 40 |
| Tabela 5 – Parâmetros de descrição dos algoritmos OpenAI-ES e CMA-ES. . . | 40 |
| Tabela 6 – Parâmetros de descrição <i>Coordinate Descent</i> | 41 |
| Tabela 7 – Quantidade de avaliações por Tamanho da População e Gerações. | 42 |
| Tabela 8 – Parâmetros modificados de descrição dos algoritmos OpenAI-ES e CMA-ES. | 44 |
| Tabela 9 – Parâmetros dos melhores resultados encontrados pelas heurísticas para valores de v_S de 7,0 e 7,5. | 47 |
| Tabela 10 – Parâmetros dos melhores resultados encontrados pelas heurísticas para todos os valores de v_S | 49 |

LISTA DE ABREVIATURAS E SIGLAS

| | |
|-----------|---|
| AE | Algoritmo Evolutivo |
| CMA | Covariance Matrix Adaptation (Adaptação da Matriz de Covariância) |
| CMA-ES | Covariance Matrix Adaptation Evolution Strategies (Estratégias Evolutivas com Adaptação da Matriz de Covariância) |
| DE | Differential Evolution |
| EE | Estratégias Evolutivas |
| OpenAI-ES | OpenAI's Evolution Strategies (Estratégias Evolutivas da OpenAI) |

LISTA DE SÍMBOLOS

| | |
|----------------------|---|
| P_B | Brake Power (Potência de Frenagem) |
| v_S | Service Speed (Velocidade de Serviço) |
| D | Diâmetro externo da hélice |
| A_E/A_O | Razão da área da pá |
| P/D | Divisão do passo da hélice pelo Diâmetro |
| Z | Número de pás na hélice |
| $t_{0.75R}$ | Espessura da lâmina a 75% do raio da hélice |
| $t_{0.75R,min}$ | Espessura mínima da lâmina da hélice exigida |
| $\sigma_{0.7R}$ | Coefficiente de carga de empuxo na Cavitação |
| $\sigma_{c0.7R,max}$ | Coefficiente de carga de empuxo máximo permitido na Cavitação |
| v_{tip} | Velocidade periférica da ponta da lâmina |
| λ | Tamanho da população |
| μ | Quantidade de melhores resultados a serem selecionados |
| σ | Tamanho do passo |
| N | Distribuição Normal Multivariada |
| g | Geração g |
| I | Matriz Identidade |
| f | Função fitness |
| x | Estado dos indivíduos |
| y_w | Variável auxiliar que representa o centro dos melhores indivíduos |
| m | Valor da Média da distribuição |
| D | População |
| x_j | j -ésimo indivíduo |
| θ | Parâmetros do indivíduo |
| Δ | Média |
| φ | Desvio padrão |
| F | Função objetivo |

SUMÁRIO

| | | |
|--------------|---|-----------|
| 1 | INTRODUÇÃO | 14 |
| 1.1 | ESTRUTURA DO TRABALHO | 16 |
| 2 | FUNDAMENTAÇÃO TEÓRICA | 17 |
| 2.1 | CONCEITOS BÁSICOS | 17 |
| 2.1.1 | Geração da Morfologia da Hélice | 17 |
| 2.1.2 | Restrições da Hélice | 18 |
| 2.1.3 | Cálculo de Propulsão da Hélice | 19 |
| 2.1.4 | Heurísticas e Algoritmos de Otimização | 20 |
| 2.1.5 | Algoritmos Evolutivos (AE) | 20 |
| 2.2 | FUNCIONAMENTO DOS ALGORITMOS | 22 |
| 2.2.1 | Função de Avaliação | 22 |
| 2.2.2 | Differential Evolution (DE) | 23 |
| 2.2.3 | Estratégias Evolutivas (EE) | 23 |
| 2.2.3.1 | CMA-ES | 24 |
| 2.2.3.2 | OpenAI-ES | 26 |
| 2.2.4 | Busca Local | 27 |
| 2.2.4.1 | Coordinate Descent | 28 |
| 2.2.5 | Busca Aleatória | 29 |
| 2.3 | VARREDURA SISTEMÁTICA DO ESPAÇO DE BUSCA | 29 |
| 3 | TRABALHOS RELACIONADOS | 31 |
| 4 | METODOLOGIA | 33 |
| 4.1 | ABORDAGEM PROPOSTA | 33 |
| 4.1.1 | Protocolo Experimental para Água | 33 |
| 4.2 | VARREDURA DO ESPAÇO DE BUSCA | 35 |
| 4.3 | EXPERIMENTOS INICIAIS COM BASE NA LITERATURA | 36 |
| 4.3.1 | Função Objetivo | 38 |
| 4.3.2 | Implementação dos Algoritmos | 39 |
| 4.3.2.1 | Implementação do DE Base | 39 |
| 4.3.2.2 | DE Modificado | 39 |
| 4.3.2.3 | Implementação do OpenAI-ES e CMA-ES | 40 |
| 4.3.2.4 | Implementação do <i>Coordinate Descent</i> | 40 |
| 4.3.2.5 | Implementação da busca aleatória | 41 |
| 4.4 | EXPERIMENTOS EXCLUSIVOS COM ESTRATEGIAS EVOLUTIVAS | 41 |
| 4.4.1 | Estudo Sobre Tamanho da População e Número de Gerações | 41 |
| 4.4.2 | Descrição dos Experimentos | 43 |
| 4.4.2.1 | OpenAI-ES e CMA-ES Modificados | 43 |
| 5 | ANÁLISE DE RESULTADOS | 45 |

| | | |
|-------|---|-----------|
| 5.1 | RESULTADO DA VARREDURA DO ESPAÇO DE BUSCA | 45 |
| 5.2 | RESULTADOS DAS HEURÍSTICAS | 46 |
| 5.3 | RESULTADOS DOS EXPERIMENTOS EXCLUSIVOS DE ESTRATÉ- GIAS EVOLUTIVAS | 48 |
| 5.3.1 | Convergência na Fitness | 50 |
| 5.3.2 | Quantidade de indivíduos válidos por geração | 51 |
| 5.3.3 | Visualização do resultado | 53 |
| 6 | CONCLUSÃO | 55 |
| 6.0.1 | Código Fonte | 56 |
| | REFERÊNCIAS | 57 |
| | APÊNDICE A – ARTIGO DO TCC | 65 |

1 INTRODUÇÃO

A utilização de drones torna-se mais presente no cotidiano em diferentes áreas com uma maior variedade de aplicações, segundo (AYAMGA; AKABA; NYAABA, 2021) essas aeronaves não tripuladas revolucionaram as operações como monitoramento do ecossistema e plantas na agricultura (HAFEEZ *et al.*, 2023), nas inspeções de segurança alcançando áreas que poderiam ser perigosas (IRIZARRY; GHEISARI; WALKER, 2012), através do mapeamento aéreo e exploração (MOHD DAUD *et al.*, 2022), proporcionando benefícios significativos, como maior praticidade, redução custos e tempo de operações.

Uma das limitações dos drones convencionais é sua restrição de locomoção exclusivamente pelo ar, conforme descrito em (DREWS-JR; NETO; CAMPOS, 2009) sendo necessário o uso de equipamentos extra ou ainda impossibilitando operações feitas na água. Enquanto que agentes autônomos submarinos feitos exclusivamente para locomoção em ambientes aquáticos ficam limitados a somente operar dentro desse ambiente (CUI, 2019), necessitando de equipamentos extras para se deslocar fora da água. Nesse contexto, a utilização de drones híbridos capazes de operar tanto no ar quanto na água torna-se favorável pela simplicidade, diminuição de custo e tempo de operação como expressado em (KASNO *et al.*, 2017), permitindo a execução de tarefas diversas e complexas em ambientes variados, em numerosas possibilidades de aplicações desde pesquisas ambientais, inspeção e manutenção de infraestruturas até operações de busca e salvamento como exposto em (DINELLI *et al.*, 2023).

A otimização da morfologia em drones tem relação direta com o aprimoramento de seu desempenho e eficiência, pois a melhora no *design* das hélices e formato do drone tem influência na capacidade de voo e a eficiência energética do motor (PEREZ; LIU; BEHDINAN, 2004). Em veículos não tripulados um dos componentes cruciais no desempenho é a presença de hélices eficientes, em razão desta ser o elemento responsável por converter energia em movimento, como explicado em (WOUD, H. K.; STAPERSMA, D., 2013).

Encontrar o formato da hélice que habilite um drone a se locomover tanto no ar quanto na água representa um problema significativo para os modelos atuais. Devido a esta falta de pesquisas específicas na área, os modelos empregados por este trabalho são inspirados em literatura de otimização de hélices para drones convencionais, como a utilização de Algoritmos Evolutivos (AEs) na escolha dos parâmetros de descrição da morfologia da hélice em (KLIMCZYK, 2022). Que como enfatizado em (SALIMANS *et al.*, 2017), AEs podem rivalizar com abordagens modernas de aprendizado por reforço, com resultados de qualidade semelhante, porém com uma abordagem mais simples do problema.

Devido aos diversos métodos para a geração do formato das hélices de drone

presentes na literatura, como apresentados em (ELMENSHAWY; ALSHWAILY, 2022). É importante selecionar um método que possa ter seu desempenho avaliado de forma objetiva, segundo (PATEL *et al.*, 2022). Desta forma este trabalho irá utilizar os parâmetros da série "Wageningen B-screw" (LAMMEREN; MANEN; OOSTERVELD, M., 1969) para descrever o formato das hélice, de forma semelhante a (MALAPUR *et al.*, 2022), devido a ampla utilização na literatura assim como a quantidade reduzida de parâmetros para a descrição da morfologia da hélice. Enquanto os testes de desempenho serão realizados com o cálculo da energia mínima necessária da hélice para mover o corpo do veículo, métrica nomeada como "Brake Power" (P_B), ou "Potência de Frenagem", de forma análoga a realizada em (MARQUES, Crístofer H.; BELCHIOR; CAPRACE, 2019).

Devido a falta de métodos de otimização na literatura que sejam focados em hélices que atuem tanto no ar e água para drones, esse trabalho se propõe a estudar a utilização de heurísticas de otimização estado da arte como algoritmos evolutivos (BENINI, 2003), especificamente OpenAI-ES (SALIMANS *et al.*, 2017) e CMA-ES (HANSEN, 2014), juntamente com outras heurísticas da literatura como algoritmos de busca local em (LEE, 2020) e busca aleatória, com a finalidade de encontrar a configuração de parâmetros de "Wageningen B-screw" para a geração de uma hélice energeticamente mais eficiente, que apresente menor P_B durante a operação em água.

Visto que a utilização de ambos parâmetros contínuos e discretos apresenta uma complexidade extra na otimização, e algoritmos de Estratégia Evolutiva trabalham apenas com parâmetros contínuos, este trabalho optou por realizar a otimização de parâmetros contínuos para cada valor do intervalo da variável discreta, número de pás de hélice. Enquanto que a utilização de heurísticas que trabalham com parâmetros contínuos e discretos como feito em (MARQUES, Crístofer H.; BELCHIOR; CAPRACE, 2019), se parece ideal para este trabalho, essa abordagem se mostrou apenas possível para velocidades limitadas de funcionamento na água.

Assim este trabalho utiliza dados de uma balsa projetada para operação em um lago no sudoeste do Brasil, assemelhando-se ao desafio abordado em (MARQUES, Crístofer H.; BELCHIOR; CAPRACE, 2019). Sendo conduzidos experimentos para cada heurística investigada em diferentes "service speed" (v_S), com o intuito de identificar configurações de hélice válidas que gerassem menores valores de P_B , incluindo análises específicas para os AEs como desempenho e configuração dos hiperparâmetros. Desta forma os AEs encontraram resultados válidos para valores de v_S os quais as abordagens de referência não haviam alcançado. Nos casos em que resultados já eram conhecidos, as soluções obtidas mostraram desempenho comparável ou superior ao cenário de referência, sendo o melhor resultado com 0,26% na redução do valor de P_B .

1.1 ESTRUTURA DO TRABALHO

O trabalho está organizado da seguinte maneira:

- Capítulo 1 expõe o problema atual que será o foco deste trabalho, assim como o os objetivos tratados nesta pesquisa.
- Capítulo 2 contém Conceitos Básicos do cálculo de propulsão da hélice e de algoritmos de otimização, necessários para compreender o restante do trabalho.
- Capítulo 3 apresenta os trabalhos relacionados na literatura que foram utilizados como base de estudo para este trabalho.
- Capítulo 4 descreve a abordagem proposta para os experimentos e análises feitas neste trabalho.
- Capítulo 5 apresenta os resultados obtidos dos experimentos e as análises sobre os dados encontrados.
- Capítulo 6 expõe um breve resumo sobre o que foi realizado neste trabalho, juntamente com as conclusões do autor e possíveis continuações deste trabalho.

2 FUNDAMENTAÇÃO TEÓRICA

Este capítulo apresenta conceitos básicos que são necessários para o entendimento do trabalho. A seção 2.1 apresenta informações sobre geração da morfologia e cálculo de propulsão de hélices. Seguida pela seção 2.2 contendo uma revisão teórica sobre métodos de otimização explicando o funcionamento dos algoritmos utilizados neste trabalho.

2.1 CONCEITOS BÁSICOS

2.1.1 Geração da Morfologia da Hélice

Na área de veículos aquáticos, a literatura oferece uma variedade de maneiras para a geração da morfologia de hélices, como abordado em trabalhos como (HORN *et al.*, 2019). A abordagem adotada para este trabalho é a série de "Wageningen B-Screw" (LAMMEREN; MANEN; OOSTERVELD, M., 1969) para a geração da morfologia da hélice, por apresentar variados estudos na literatura e ser amplamente utilizada.

A série de "Wageningen B-Screw" faz uso de equações polinomiais para determinar os atributos da hélice e realizar cálculos de desempenho hidrodinâmicos. Uma grande vantagem desta abordagem é o uso de quantidade reduzida de parâmetros necessários para descrever a morfologia de uma hélice, contando com apenas quatro, conforme detalhados na Tabela 1.

Como descrito em (LAMMEREN; MANEN; OOSTERVELD, M., 1969) os parâmetros da Tabela 1 possuem os seguintes significados:

Diâmetro da hélice (D) é a medida do círculo circunscrito pelas pontas das pás da hélice, determinando o tamanho da hélice.

Área expandida dividida pela área original (A_E/A_O), representa a razão entre a área total da lâmina da hélice expandida, área que a lâmina ocupa quando totalmente estendida, e a área original da lâmina.

Relação passo pelo diâmetro (P/D), sendo a relação entre o passo da hélice, ou também chamada de *pitch*, e o diâmetro da hélice.

Número de lâminas (Z) é parâmetro discreto referente ao número de pás em uma hélice.

Tabela 1 – Parâmetros para a geração da hélice.

| Parâmetro | Significado | Intervalo | Tipo |
|-----------|--|-----------|----------|
| D | Diâmetro externo da hélice (metros) | 0,5–0,8 | Contínuo |
| A_E/A_O | Razão da área da pá | 0,3–1,05 | Contínuo |
| P/D | Razão do passo da hélice pelo diâmetro | 0,5–1,4 | Contínuo |
| Z | Número de pás na hélice | 2–7 | Discreto |

Fonte: Elaborada pelo autor.

Sendo esses parâmetros melhorados pelas heurísticas de otimização, respeitando seus respectivos intervalos, de forma similar ao realizado em (MARQUES, Crístopher H.; BELCHIOR; CAPRACE, 2019).

A Figura 1 representa a influência dos parâmetros da Tabela 1 na geração da morfologia da hélice, o software utilizado para a geração das imagens da geometria foi CAESSES® do Friendship Systems AG.

As Figuras 1a e 1b apresentam as configurações de diâmetro externo mínimo e máximo, respectivamente. Nota-se que essas variações têm um impacto direto no tamanho das pás da hélice.

A Figura 1c representa o valor mínimo da razão da área da pá, enquanto a Figura 1d representa a configuração com a razão com o valor máximo.

As Figuras 1e e 1f apresentam o valor da razão do passo da hélice pelo diâmetro, mínima e máxima respectivamente, evidenciando como esse valor afeta o ângulo das pás da hélice.

As Figuras 1g e 1h demonstram os valores de 3 e 5 para o número de pás, que não são os extremos do intervalo para esse parâmetro, entretanto são os limites estabelecidos no software utilizado.

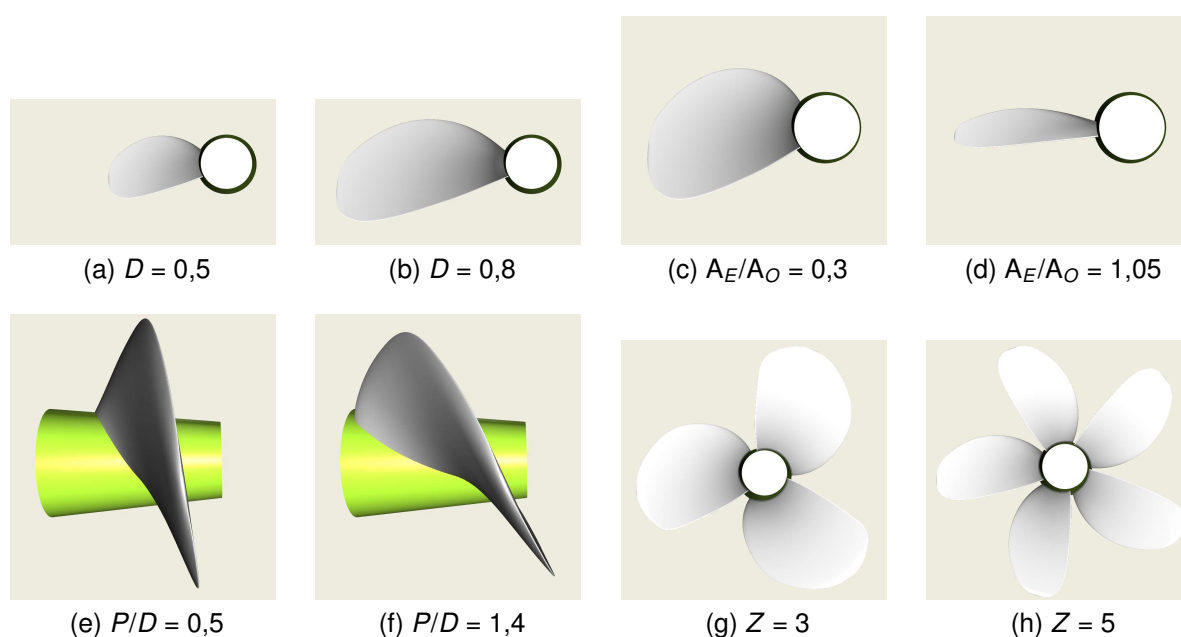


Figura 1 – Influência dos Parâmetros na Geometria da hélice

Fonte: Elaborada pelo autor.

2.1.2 Restrições da Hélice

As hélices são afetadas por restrições físicas, o que impossibilita a viabilidade de algumas configurações de morfologia, como discutido em (WEBB; EAMES; TUFFREY, 1975). Portanto, na avaliação das hélices, foram consideradas as seguinte

restrições, como descrito em detalhes em (MARQUES, Crístofer H.; BELCHIOR; CAPRACE, 2019) e (MARQUES, Crístofer Hood *et al.*, 2023).

Resistência da lâmina da hélice: Segundo (CARLTON, JS, 2007) caso a lâmina da hélice não seja capaz de suportar as forças envolvidas durante a operação, pode ocorrer desta entortar, se deformar ou até mesmo quebrar, devido às altas tensões e cargas envolvidas, resultando no desbalanceamento, gerando perda de eficiência, ou até total inoperabilidade. Para restringir hélices que não atendem aos requisitos de resistência adequada, é necessário que a espessura da lâmina a 75% do raio da hélice ($t_{0.75R}$) não seja inferior à espessura mínima exigida ($t_{0.75R,min}$). Foram utilizadas as abordagens para $t_{0.75R}$ e $t_{0.75R,min}$ fornecidas em (OOSTERVELD, M. W. C.; OOSSANEN, 1975).

Cavitação: A cavitação é um fenômeno que ocorre quando a diminuição de pressão gerada pela rotação das pás formam bolhas de ar que podem estourar próximas a hélice gerando um impacto, como descrito em (CAUPIN; HERBERT, 2006). Segundo (KUIPER, 1997), a cavitação durante a operação da hélice pode causar problemas como redução do desempenho, danos estruturais e ruído. Portanto, a restrição da cavitação na parte traseira da lâmina foi feita de acordo com o diagrama de *cavitação de Burrill* para fluxo uniforme (CARLTON, J.S., 2019). De forma que o coeficiente de carga de empuxo ($\sigma_{0.7R}$) seja inferior ou igual ao coeficiente de carga de empuxo máximo ($\sigma_{c0.7R,max}$).

Velocidade periférica da hélice: Velocidades periféricas elevadas podem resultar em níveis mais altos de ruído e vibração que implicam na durabilidade e confiabilidade da hélice, além de resultarem em redução da eficiência. A velocidade periférica na ponta da lâmina (v_{tip}) deve ser mantida a mais baixa possível para reduzir ruído e vibração. Com a velocidade máxima, v_{tip} , não ultrapassando 39 m/s, como descrito em (HOLDEN; FAGERJORD; FROSTAD, 1980).

2.1.3 Cálculo de Propulsão da Hélice

A Hélice que trabalha em um ambiente aquático deve considerar a velocidade da corrente de água na qual ela estará imersa. Essa métrica é nomeada "service speed" (v_S), ou "velocidade de serviço", que denota a velocidade em nós na qual uma embarcação normalmente opera durante seu serviço, como descrito em (CARLTON, J.S., 2019). Quando projetada a lâmina, deve-se escolher o valor de v_S de acordo com a aplicação, pois o propósito de operação da embarcação e as condições do ambiente em que a hélice será utilizada devem ser levados em conta. Dessa maneira diferentes valores de v_S podem ser estudados no *design* da hélice, de acordo com as necessidades do projeto. Porém, como essa métrica é utilizada no cálculo de propulsão, obriga que comparações entre diferentes designs de hélices devam ser feitos somente entre valores iguais de v_S .

O desempenho da hélice em água pode ser medido pela métrica de "Brake Power" (P_B), ou "Potência de Frenagem" em português, que refere-se à potência do motor de sistemas de propulsão necessária para mover a embarcação, sendo uma medida da potência de saída de um motor (STAPERSMA; WOULD, H., 2005).

A métrica de P_B é particularmente relevante em veículos aquáticos, pois representa a potência do motor necessária para que a hélice consiga fazer a embarcação superar a velocidade da água (v_S) na qual está operando. Logo uma hélice que requer menos potência do motor, i. e. o menor valor de P_B , para uma determinada v_S apresenta uma maior eficiência. Assim os métodos de otimização devem buscar essa configuração de parâmetros de hélice que minimizem o valor de P_B para uma v_S escolhida, como explicado em (MARQUES, Crístofer H.; BELCHIOR; CAPRACE, 2019).

Entretanto a otimização deve considerar as restrições operacionais estabelecidas, visto que ultrapassar qualquer desses limites torna a hélice não viável para operação, como abordado em (BENINI, 2003). Assim não deve ser considerada somente o menor valor de P_B , mas sim o menor que gere uma configuração válida de hélice, na qual as restrições estejam dentro dos seus limites.

2.1.4 Heurísticas e Algoritmos de Otimização

Os algoritmos de otimização do tipo métodos exatos são utilizados para encontrar uma solução ótima, ou seja, a melhor possível para o determinado problema. Segundo (ROY, 1971) algoritmos de otimização utilizam de uma função objetivo que faz a avaliação de uma solução candidata, medindo seu desempenho ou qualidade em relação ao problema. Assim os algoritmos de otimização tem como objetivo encontrar os valores dos parâmetros que tenham o melhor valor na função objetivo, ou seja, tenham o melhor desempenho no problema, como explicado em (ADBY, 2013).

Heurísticas são procedimentos computacionais que melhoram uma solução candidata iterativamente com relação a uma medida de qualidade, como a dada pela função objetivo. As heurísticas fazem poucas suposições sobre o problema e exploram diversas de soluções candidatas para encontrar soluções ótimas ou muito próximas de ótimas locais e globais, mas com foco em terem custo computacional razoável, procurando por soluções vizinhas das que encontrarem os melhores valores. Assim são muito indicadas para problemas em que garantir a solução ótima é muito difícil, já que não podem poder garantir que a solução encontrada seja a ótima global, como explicado em (WANG; CHEN, 2013).

2.1.5 Algoritmos Evolutivos (AE)

Segundo (DOERR; NEUMANN, 2019) algoritmos evolutivos são uma classe de algoritmos de otimização inspirados na evolução biológica, que operam de forma

similar ao processo de seleção natural.

Geralmente essas heurísticas utilizam populações de soluções candidatas, representadas por **indivíduos**, ou também chamados de **cromossomos**, onde cada indivíduo é composto por **genes** que são os parâmetros a serem melhorados pelo algoritmo. Ou seja, cada indivíduo possui um conjunto de valores para que servem como solução candidata a resolução do problema.

Os indivíduos da população são avaliados por uma **função de avaliação**, que mede seu desempenho em um determinado objetivo. Com base nessa avaliação, seleciona-se os melhores indivíduos para gerar uma nova população, através de técnicas como mutação e recombinação entre os indivíduos, como discutido em (BARTZ-BEIELSTEIN *et al.*, 2014).

A **mutação** introduz pequenas alterações aleatórias nos indivíduos, permitindo uma diversificação nos atributos e criação de características não presentes na população. Pequenos valores para taxa de mutação não criam uma grande variedade entre os indivíduos, levando o algoritmo a explorar um pequeno espaço de busca. Entretanto grandes taxas de mutação fazem os valores serem muito aleatórios e dificulta na convergência.

Enquanto que o **cruzamento**, também chamado de recombinação ou *crossover*, por sua vez, combina características dos indivíduos existentes, gerando novos indivíduos com atributos diferentes dos indivíduos pais, mas mantendo características já presentes na população (DOERR; NEUMANN, 2019). Em alguns modelos de AE, são definidas elites de soluções, as que apresentarem os melhores valores de fitness, utilizando apenas a mutação sobre esses indivíduos, e assim pode ser pulada a etapa de cruzamento, levando a uma convergência em menos iterações.

Após a mutação e a recombinação, ocorre a **seleção**, onde depois de avaliar os indivíduos da população, os melhores são selecionados através de critérios determinados, onde algoritmos diferentes aplicam critérios diferentes, de forma a criar uma nova população para a geração seguinte. Caso o AE aplique uma maior probabilidade de selecionar os melhores indivíduos, com base no seu desempenho na função objetivo, se tem elitismo, onde os indivíduos mais adaptados tem maior chance de passar seus genes para as próximas gerações, como explicado em (DUAN, 2022).

Através da iteração pelas gerações, visando o aprimoramento das soluções, o algoritmo tende a **convergir** para uma solução que melhore o valor da função objetivo.

Assim a população vai evoluindo ao longo das gerações, de modo que, ao fim da execução, sejam encontradas soluções boas e desta forma se obtenha os parâmetros melhorados para o problema, como explicado em (EIBEN; SMITH, 2015). A Figura 2 apresenta uma visão geral do funcionamento de um Algoritmo Evolutivo.

Segundo (SLOWIK; KWASNICKA, 2020), AEs são flexíveis e podem lidar com problemas de otimização multiobjetivo, problemas com restrições complexas, ou de

alta dimensionalidade, e por isso vem sendo cada vez mais utilizados em áreas como na engenharia.

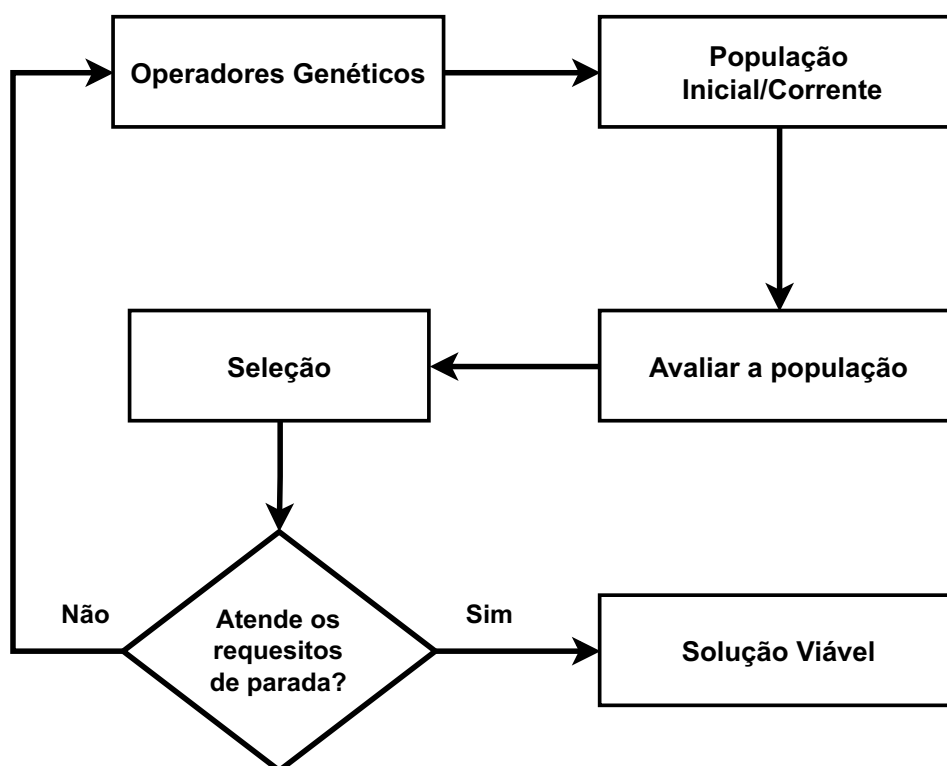


Figura 2 – Visão geral do funcionamento de um Algoritmo Evolutivo.

Fonte: Adaptado de (PACHECO *et al.*, 1999).

2.2 FUNCIONAMENTO DOS ALGORITMOS

Esta seção contém uma revisão teórica sobre as heurísticas de otimização utilizadas neste trabalho, descrevendo o funcionamento e algumas vantagens e possíveis desvantagens da utilização de cada um.

E a seção 2.3 descreve o estudo no espaço de busca dos parâmetros afim de explorar melhor as suas influências nos resultados dos experimentos.

2.2.1 Função de Avaliação

Como explicado em (BARTZ-BEIELSTEIN *et al.*, 2014), a função de avaliação ou função fitness, é uma maneira de avaliar a qualidade ou o desempenho dos parâmetros, através da atribuição de um valor numérico, uma forma de pontuação, representando o quão adequada a solução candidata é em relação ao objetivo do problema.

2.2.2 Differential Evolution (DE)

O *Differential Evolution* (DE) é um algoritmo evolutivo utilizado em problemas com espaços de busca complexos, proposto em (STORN; PRICE, K., 1997b). O algoritmo não utiliza o gradiente do problema, permitindo ser usado em problemas de otimização que são ruidosos ou que não são contínuos.

Como explicado em (DAS; SUGANTHAN, 2011), o DE encontra uma boa solução para o problema mantendo uma população de soluções candidatas e criando novas soluções candidatas através de combinação das pré existentes, por meio de mutação e recombinação. Desta forma criando novas soluções candidatas e explorando o espaço de busca. Durante as gerações, mantém-se a solução candidata que apresente melhor desempenho no problema de otimização.

O algoritmo tem início com a população de indivíduos com valores aleatórios distintos para os genes. Cada indivíduo é composto por um vetor de números reais, equivalente aos parâmetros a serem otimizados, sendo estes aplicados a função de fitness para ser feita a avaliação do indivíduo, composta um número real que indica a qualidade da solução candidata no problema. São criados os novos indivíduos da população utilizando os operadores genéticos para o melhor indivíduo encontrado até o momento.

São feitas iterações até se alcançar o critério de parada, podendo ser a quantidade de gerações, ou uma não mudança na fitness por determinadas gerações. Desta forma, se espera alcançar uma solução melhor para o problema, conforme descrito em (STORN; PRICE, K., 1997a).

2.2.3 Estratégias Evolutivas (EE)

Os algoritmos de Estratégias Evolutivas (EE) são AE que realizam o processo de busca e aprimoramento gradual de soluções, de maneira a imitar o que acontece na natureza com a seleção natural (HANSEN; ARNOLD; AUGER, 2015). Os algoritmos utilizam populações de soluções candidatas que são avaliadas por uma função de avaliação. Com base nessa avaliação, os melhores indivíduos são selecionados para gerar uma nova população, utilizando técnicas como mutação e recombinação. Como é explicado por (HA, 2017a) com esse processo iterativo, espera-se que a cada iteração as soluções evoluam apresentando melhores valores da função de avaliação em comparação às iterações anteriores. Dessa forma, é possível encontrar soluções cada vez mais aprimoradas para o problema.

Uma das vantagens das EE é a capacidade de testar a população de soluções candidatas de forma simultânea, visto que para cada iteração a função de avaliação de cada indivíduo é independente dos demais da população (SALIMANS *et al.*, 2017). Podendo assim acelerar significativamente o processo de busca, reduzindo o tempo

de cada iteração.

Diferentes algoritmos de EE fazem uso de distintas estratégias de busca e atualização das soluções candidatas durante o processo evolutivo, isso inclui a geração de novas soluções candidatas, a introdução de variações e a diversificação das soluções presentes na população. O algoritmo OpenAI-ES (SALIMANS *et al.*, 2017) utiliza uma abordagem de amostragem estocástica, onde as soluções candidatas são geradas com base em uma distribuição probabilística. Enquanto o *Covariance Matrix Adaptation Evolution Strategy* (CMA-ES) (HANSEN; OSTERMEIER; GAWELCZYK, 1997) utiliza uma adaptação da matriz de covariância (CMA) dos resultados da função de avaliação da população para se orientar na direção onde as melhores soluções foram encontradas.

2.2.3.1 CMA-ES

O algoritmo do CMA-ES (HANSEN, 2014) pertence à classe das EE, utilizando uma adaptação da matriz de covariância (CMA) (HANSEN; OSTERMEIER, 1996) podendo levar em consideração os resultados de cada geração para criar de forma adaptativa o espaço de busca para a próxima geração, através do cálculo da matriz de covariância completa do espaço de parâmetros (HANSEN, 2023). A matriz de covariância é uma matriz quadrada que fornece a covariância entre cada par de elementos de um determinado conjunto de parâmetros, nesse caso, os indivíduos da população.

Como detalhado em (HANSEN, 2014) para cada geração, o CMA-ES usa uma distribuição normal multivariada dos parâmetros para criar as amostras das soluções. Isso acontece através da avaliação de cada indivíduo da população, daí escolhe-se um grupo com os melhores indivíduos, e utiliza apenas as melhores soluções, juntamente com a média das soluções da geração atual, para calcular a matriz de covariância da próxima geração. E assim criar a próxima geração usando a nova matriz de covariância gerada (MELO, 2019). Desta forma o algoritmo consegue expandir ou reduzir o espaço de busca de forma iterativa.

O algoritmo possui população de tamanho λ , μ quantidade de melhores soluções candidatas a serem selecionadas em cada geração, e σ como o tamanho do passo. A distribuição multivariada normal (N) pode ser representada como na Equação (1).

Na geração inicial, $g = 0$, a CMA está como na Equação (2), produzindo uma distribuição em forma esférica, devido a matriz de identidade I , como na imagem mais a esquerda da Figura 3, e de forma similar a imagem (1) da Figura 4.

Então é feita a distribuição das amostras, com os λ indivíduos, é feito o ranking (classificação) através da função de fitness $f(x)$, e então selecionados os melhores μ indivíduos. Como representado na imagem (2) da Figura 4.

É calculado a variável auxiliar y_w para a geração g , como na Equação (3), sendo o centro dos melhores μ indivíduos. Como representado na imagem (3) da Figura 4.

Assim pode ser atualizada o centro da CMA para a nova distribuição com a média \mathbf{m} como na Equação (4), com tamanho do passo de σ . Como representado na imagem (4) da Figura 4.

Então é feita a atualização da matriz de covariância, para melhorar a "fitness esperada" da próxima geração, como na Equação (5). Isso altera o formato da matriz de covariância para um elipsoide com a excentricidade proporcional ao valor de covariância dos parâmetros. E representado pela imagem do meio na Figura 3 e imagem (3) da Figura 4.

Este processo continua de forma iterativa, através das gerações, como na Equação (6), até se alcançar o critério de parada, como o limite máximo de gerações.

$$\mathcal{N}(\mathbf{m}, \mathbf{C}) \sim \mathbf{m} + \mathcal{N}(\mathbf{0}, \mathbf{C}) \quad (1)$$

$$\mathcal{N}^{(0)} \sim \mathcal{N}(\mathbf{0}, \mathbf{I}) \quad (2)$$

$$\mathbf{y}_w^{(g)} = \sum_{i=1}^{\mu} w_i \mathbf{x}_{i:\lambda}^{(g)} \quad (3)$$

$$\mathbf{m}^{(g+1)} = \mathbf{m}^{(g)} + \sigma^{(g)} \mathbf{y}_w \quad (4)$$

$$\mathbf{C}^{(g+1)} = 0,8\mathbf{C}^{(g)} + 0,2\mathbf{y}_w^{(g)} (\mathbf{y}_w^{(g)})^\top \quad (5)$$

$$\mathbf{x}^{(g+1)} = \mathbf{m}^{(g)} + \sigma^{(g)} \mathcal{N}(\mathbf{0}, \mathbf{C}^{(g)}) \lambda \quad (6)$$

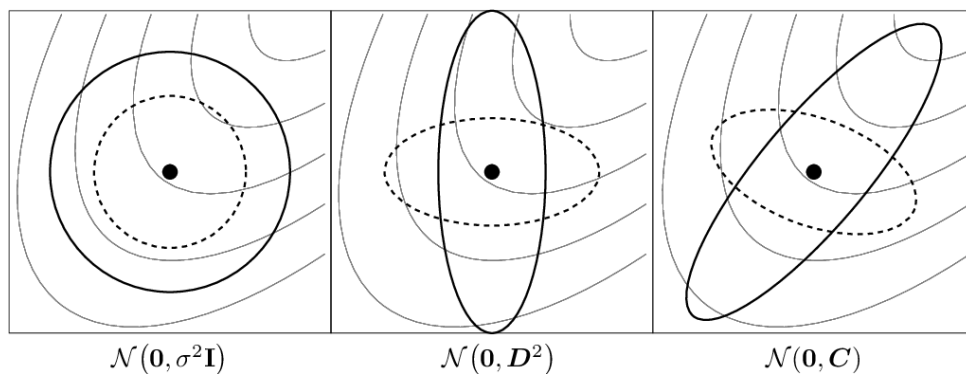


Figura 3 – Representação do ajuste da CMA.

Fonte: Retirado de (HANSEN, 2023).

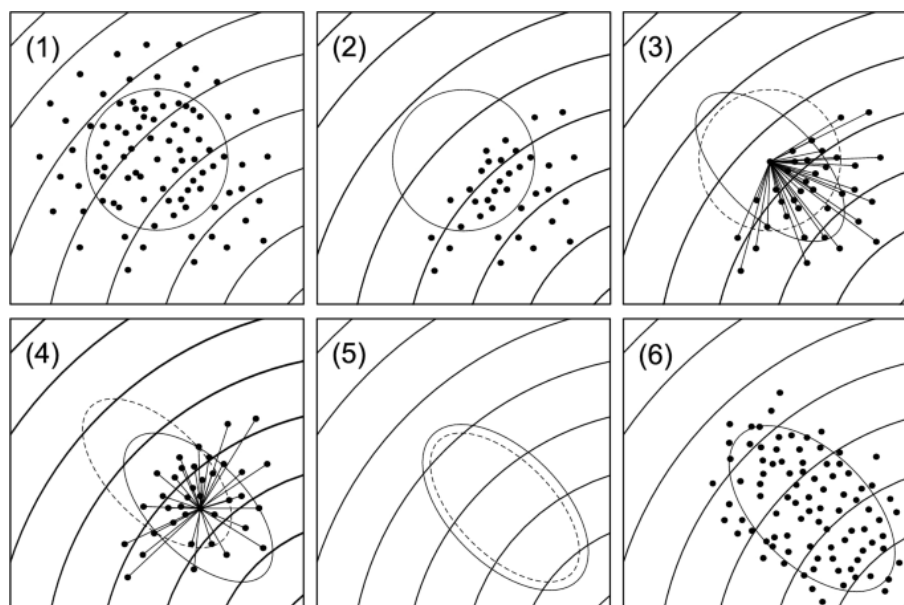


Figura 4 – Representação do funcionamento do CMA-ES.

Fonte: Retirado de (MAKI; AKIMOTO; NAOYA, 2021).

2.2.3.2 OpenAI-ES

O algoritmo de EE da OpenAI (SALIMANS *et al.*, 2017), abreviado nesse trabalho para OpenAI-ES pertence à classe das Estratégias de Evolução Natural (WIERSTRA *et al.*, 2008) e utiliza um estimador da função de pontuação por meio do método de ascensão de gradiente estocástico para maximizar o valor médio da função de avaliação.

A cada iteração, o algoritmo gera uma população a partir de um indivíduo, adicionando uma perturbação aos parâmetros. Isso é feito através da adição de ruído gaussiano nos parâmetros dos indivíduos da população. Após avaliar os indivíduos com a função de avaliação, os valores médios são classificados e normalizados.

Para reduzir a variância, é utilizada a **amostragem antitética** (*antithetic sampling*), que consiste em gerar um par de perturbações, uma com o sinal oposto da outra, para os valores do vetor de perturbação de ruído gaussiano, assim reduzindo a variabilidade das estimativas da função objetivo. Ao calcular a média dos valores da função objetivo em cada par antitético, a variação entre eles é reduzida podendo levar a uma convergência mais rápida, como explicado em (GEWEKE, 1988).

Também é realizado o **ajuste de avaliação** (*fitness shaping*) aplicando uma transformação de classificação (*rank transformation*) aos resultados das avaliações antes de calcular cada atualização de parâmetro (HA, 2017a). Dessa forma, a influência de indivíduos atípicos em cada população é removida e a tendência do algoritmo de cair em ótimos locais no início do treinamento é reduzida. Além disso, também é aplicado o decaimento de peso (*weight decay*) para evitar que os parâmetros cresçam

muito em relação às mutações como explicado em (SALIMANS *et al.*, 2017).

Primeiro é gerada a população D descrito na Equação (7), onde x_i representa o i -ésimo indivíduo da população de tamanho λ , e $f(x_i)$ representa o valor da fitness para aquele indivíduo.

É feito o mapeamento de $p_\theta(x)$ como uma distribuição gaussiana, onde θ mantém a média Δ , e o desvio padrão φ . Assim se gera a população de tamanho λ partindo da distribuição Gaussiana. Se seleciona os μ melhores indivíduos com base na função fitness f , sendo esse subconjunto chamado de **elite**. Então é estimado a nova média e desvio padrão para a próxima geração ($g+1$) utilizando a elite através das Equações (9) e (10).

$$D = \{(x_i, f(x_i))\} \quad (7)$$

$$\theta = (\Delta, \varphi), p_\theta(x) \sim \Delta + \varphi \mathcal{N}(0, \mathbf{I}) \quad (8)$$

$$\Delta^{(g+1)} = \text{média}(D_{elite}^{(g+1)}) = \frac{1}{\Delta} \sum_{i=1}^{\mu} x_i^{(g+1)} \quad (9)$$

$$(\varphi^{(g+1)})^2 = \text{variância}(D_{elite}^{(g+1)}) = \frac{1}{\Delta} \sum_{i=1}^{\mu} (x_i^{(g+1)} - \Delta^{(g)})^2 \quad (10)$$

2.2.4 Busca Local

São heurísticas que escolhem a melhor solução em cada iteração, assim percorrem o espaço de busca a partir de um ponto de partida dado. Ao contrário dos algoritmos de busca global, os métodos de busca local concentram-se em encontrar uma solução que seja localmente ótima dentro de uma vizinhança da solução atual, como explicado em (VOSS *et al.*, 2012). Porém isso não garante que eles cheguem em uma boa solução no final da execução, visto que podem acabar ficando presos em vales locais onde encontraram a melhor solução ao redor, que pode ou não ser a melhor entre todas as soluções possíveis, como descrito em (LAND, 1998).

Como representado na Figura 5 os mínimos locais são os menores valores ao seu redor, porém não os menores de toda a função, assim caso o algoritmo acabe no mínimo local, a solução que ele encontrou não é a melhor para o problema como um todo.

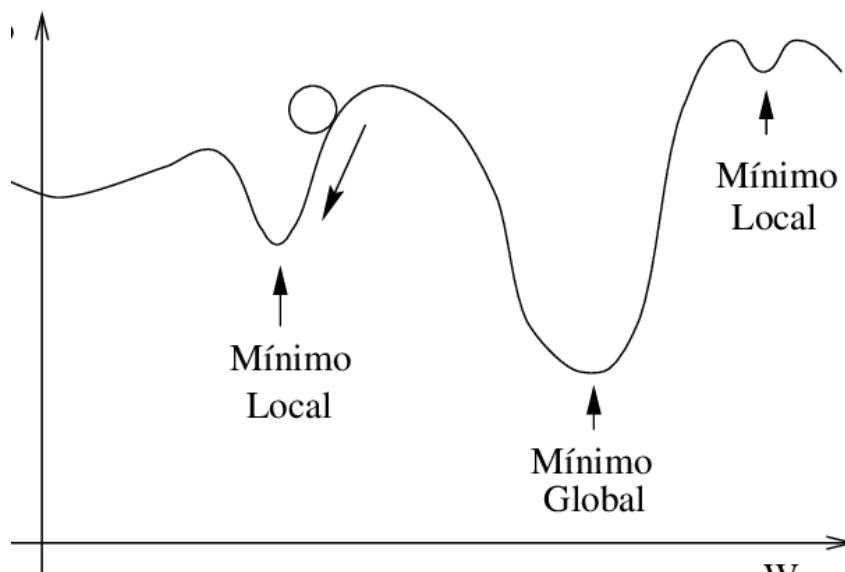


Figura 5 – Diferença entre mínimo local e global.

Fonte: Retirado de (CORTEZ, 1997).

2.2.4.1 Coordinate Descent

Para problemas de minimização com múltiplas variáveis, o algoritmo *Coordinate Descent* simplifica o objetivo alterando apenas uma coordenada por vez, mantendo fixas as demais. O algoritmo começa com um valor aleatório dentro do intervalo de cada coordenada. Em cada iteração, o algoritmo realiza a seleção de uma das coordenadas para ser feita a otimização, mantendo as demais fixas, percorrendo valores dentro dos limites da coordenada selecionada, para encontrar o valor que gere o melhor resultado. Assim se percorre o conjunto de variáveis a serem otimizadas, otimizando uma de cada vez.

Como explicado em (FRIEDMAN *et al.*, 2007), é importante a escolha do método utilizado na melhora da coordenada, no caso deste trabalho foi escolhido a busca linear, que consiste em percorrer um intervalo definido em passos de um tamanho determinado, avaliando através da função de avaliação cada ponto, afim de encontrar o valor que gere melhor resultado da função de avaliação, como explicado em (SLAVÍK, 1997). Em cada iteração é feita uma melhora para cada uma das coordenadas, e este processo é repetido até que um critério de parada seja atingido. Neste trabalho foi utilizado um número máximo de iterações, que será melhor explicado na subseção 4.3.2.4.

Para x sendo um conjunto de n parâmetros na geração g . O algoritmo tem início com valores aleatórios para cada $x_i^{(0)}$, como representado na Equação (11), sendo i o i -ésimo elemento do vetor de tamanho n .

Para cada geração g , se escolhe k , para selecionar o elemento x_k a ser otimizado mantendo todos os outros parâmetros fixos. Se minimiza o valor da função de

avaliação F para $F(x_k)$, como na Equação (12) onde p representa todos os valores possíveis para x_k dentro do seu espaço de busca.

Assim se percorre todos parâmetros $x_i^{(g)}$ para i de 0 a n , minimizando o valor da $F(x_i^{(g)})$, para cada iteração g , até que se alcance o critério de parada, como o limite máximo de iterações.

$$\mathbf{x}^{(g)} = (x_1^{(g)}, \dots, x_i^{(g)}, \dots, x_n^{(g)}) \quad (11)$$

$$F(x_k) < F(p) \quad (12)$$

2.2.5 Busca Aleatória

Algoritmo de busca aleatória, é uma heurística que faz escolhas para explorar o espaço de busca de forma aleatória. Segundo (KARNOPP, 1963), ao contrário de muitos outros algoritmos que seguem uma abordagem sistemática, a ideia principal por trás da busca aleatória é gerar amostras aleatórias no espaço de busca e avaliar o desempenho dessas amostras usando uma função de avaliação.

Essa abordagem torna-se vantajosa por ser simples a implementação e funcionamento, ao mesmo tempo que apresenta uma possível solução viável para problemas onde o espaço de busca é muito rugoso, contendo muitos vales que poderiam atraparilhar outras estratégias, como algoritmos de busca local que ficariam presos dentro dos vales locais e teriam muita dificuldade em encontrar uma solução ótima global, como explicado em (ANDERSSON; BLOOMFIELD, 1975). Como cada amostra é independente das demais o algoritmo é altamente paralelizável. No entanto muitas amostras devem ser testadas a fim de conseguir uma boa solução final, pois não há garantia de que a solução ótima será encontrada, ou até que a melhor solução encontrada seja uma ótima local e muito menos ótima global, como explicado em (PRICE, W. L., 1983).

2.3 VARREDURA SISTEMÁTICA DO ESPAÇO DE BUSCA

Como não há uma extensa pesquisa sobre os parâmetros das hélices de drones híbridos na literatura, se percebeu a necessidade de realizar um estudo no espaço de busca dentro dos limites dos parâmetros para explorar melhor o comportamento e influência destes nos resultados dos experimentos. Ao se conhecer as características do espaço de busca, é possível entender melhor o funcionamento das heurísticas sobre o problema.

Assim foi adotada a **varredura sistemática** sobre o espaço de busca dos parâmetros de forma semelhante a realizada em (PORUMBEL; HAO; KUNTZ, 2010). No entanto, é importante destacar que uma varredura sistemática tem limitações, visto que esta abordagem executa o número exponencial de combinações possíveis, e quando

aplicada a espaços de busca grandes ou complexos, se torna impraticável pelo tempo e recursos necessários. Como esse método será utilizado para um espaço de busca limitado e pequeno, com um número reduzido de parâmetros, a abordagem de realizar uma varredura sistemática se mostra adequada para uma avaliação preliminar das funções de avaliações e dos resultados obtidos pelas heurísticas estudadas. Sendo assim a varredura sobre o espaço de busca foi realizada apenas para um estudo preliminar com o intuito de melhorar a compreensão do problema e funcionamento dos algoritmos.

Neste trabalho, **somente para a varredura**, optou-se por discretizar e segmentar o espaço de cada parâmetro, em um número determinado de elementos criando uma malha com resolução definida do espaço de busca. O estudo neste tópico da varredura, tem apenas o objetivo de fornecer uma melhor compreensão do problema, foi feita a escolha do tamanho dos segmentos de forma que a execução fosse viável dentro do escopo deste trabalho em questões de tempo, que será melhor explicado na subseção 4.2.

3 TRABALHOS RELACIONADOS

Trabalhos de otimização paramétrica para hélices de drones híbridos, capazes de se locomover tanto no ar quanto na água, são limitados na literatura. Entretanto para drones convencionais que operam apenas no ar, diversos trabalhos sobre a otimização de hélices são encontrados. Dentre esses estudos é comum a utilização de algoritmos evolutivos, aplicados para encontrar boas configurações de parâmetros de hélices de um drone convencional, como em (KLIMCZYK, 2022).

A utilização de Estratégias Evolutivas (EE) como algoritmo de otimização para hélices de drones está presente na literatura, como em (CURRISTON; HARTFIELD; BURKHALTER, 2013) que utiliza cálculo da sustentação, resistência gerada pelas hélices no motor e potência gerada durante a rotação das pás, como parâmetros a serem otimizados. Enquanto a sustentação máxima gerada pela hélice como valor a ser maximizado pelo algoritmo de EE. Sendo utilizado o simulador *NACA 0012* (AIRFOILTOOLS. . . , s.d.) para realizar os testes de sustentação.

Algoritmos Genéticos são aplicados e analisados por (DUAN, 2022), para a otimização de hélices de um drone aéreo. São utilizados valores de inclinação em graus, afinamento, tamanho da corda raiz em metros, e a semi-envergadura em graus, como parâmetros a serem melhorados. Deste modo, cria-se um modelo 3D das pás, e então utilizado o simulador *ANSYS 2020* (ANSYS. . . , 2020) para gerar o valor de "Figure of Merit" (SFORZA, 2017), métrica que representa a eficiência global do propulsor, levando em consideração aspectos do desempenho, como a tração gerada, a potência consumida e a eficiência aerodinâmica. Sendo calculado como a razão entre a tração gerada e a potência consumida pelo propulsor, e quanto maior o valor mais eficiente é a conversão de energia fornecida em tração útil.

Para hélices de embarcações marítimas usualmente é utilizada a série "*Wageningen B-Screw*" (LAMMEREN; MANEN; OOSTERVELD, M., 1969) para a descrição de hélices aquáticas, sendo necessário no mínimo apenas os 4 parâmetros listados na Tabela 1. Segundo (HELMA, 2020) a praticidade na descrição da hélice e a grande quantidade de estudos realizados sobre a série "*Wageningen B-Screw*" a torna bem atrativa. Uma aplicação da série "*Wageningen B-Screw*" em otimização de hélices é em (DOIJODE *et al.*, 2022) com a utilização de métodos de aprendizado de máquina para encontrar uma configuração de hélice eficiente com pouca cavitação.

Sobre a otimização de hélices exclusiva para embarcações aquáticas, como apresentado em (MARQUES, Crístofer H.; BELCHIOR; CAPRACE, 2019) é aplicado o algoritmo Differential Evolutions (DE) para melhorar hélices para barcos, geradas a partir da série de "*Wageningen B-Screw*" que utiliza métricas para gerar a forma de uma hélice e testar seu desempenho na água, usando os parâmetros representados na tabela 1, e utiliza o valor de P_B para medir o desempenho da hélice testada, de forma

que os menores valores de P_B apresentam a melhor eficiência da hélice, sendo essa a métrica utilizada pelo DE para encontrar bons parâmetros da série de "Wageningen B-Screw" que descrevem as hélices.

Como pode ser observado na Tabela 2 a maioria dos estudos revisados adota abordagens baseadas em AE para otimizar o design de hélices de veículos não tripulados. Contudo, há uma falta de consenso quanto ao objetivo central da otimização. Alguns trabalhos concentram-se na busca por maior eficiência energética durante a operação da hélice como (KLIMCZYK, 2022) e (MARQUES, Crístofer H.; BELCHIOR; CAPRACE, 2019). Enquanto outros priorizam o empuxo máximo para um conjunto específico de parâmetros (DUAN, 2022) e (DOIJODE *et al.*, 2022). Outras pesquisas buscam eficiência considerando um conjunto abrangente de parâmetros, como coeficiente de sustentação, nível de ruído e velocidade de operação, (CURRISTON; HARTFIELD; BURKHALTER, 2013). Notavelmente, alguns estudos revisados impõem restrições à velocidade de operação. Neste trabalho, foi feito a fim de evitar tais limitações, visando ampliar a viabilidade da abordagem proposta na prática.

Tabela 2 – Conteúdo dos Trabalhos Relacionados.

| Trabalho | Utiliza Algoritmos Evolutivos | Otimização Eficiência Energética | Hélice para Meio Aquático | Sem Restrição na Velocidade de Operação |
|--|--------------------------------------|---|----------------------------------|--|
| (KLIMCZYK, 2022) | ✓ | ✓ | | ✓ |
| (CURRISTON; HARTFIELD; BURKHALTER, 2013) | ✓ | | | ✓ |
| (DUAN, 2022) | ✓ | | | |
| (DOIJODE <i>et al.</i> , 2022) | | | ✓ | ✓ |
| (MARQUES, Crístofer H.; BELCHIOR; CAPRACE, 2019) | ✓ | ✓ | ✓ | |
| Este Trabalho | ✓ | ✓ | ✓ | ✓ |

Fonte: Elaborada pelo autor.

4 METODOLOGIA

Essa seção é dedicada a explicar a implementação e execução dos algoritmos utilizados na melhora dos parâmetros das hélices de drones híbridos, afim de obter resultados que possuam uma melhor eficiência de propulsão. Primeiro será explicado a Abordagem Proposta para a realização dos experimentos em 4.1. Seguido pelos detalhes de como foi realizada a varredura no espaço de busca na seção 4.2. Em sequência como foram implementados os Experimentos Iniciais com Base na Literatura na seção 4.3. E por fim, são apresentados os detalhes da implementação dos algoritmos de Estratégias Evolutivas na seção 4.4.

4.1 ABORDAGEM PROPOSTA

Inicialmente, foram realizados experimentos exclusivamente na otimização de hélices para operação na água. Essa abordagem foi escolhida devido à escassez de estudos disponíveis na literatura sobre agentes autônomos submarinos. Portanto, optou-se por iniciar com uma análise e avaliação das hélices em ambiente aquático e somente após a confirmação do funcionamento com eficácia para hélices na água, para então ser dada continuação aos experimentos mais complexos.

Essa abordagem possibilita um melhor entendimento do funcionamento dos métodos de otimização específicos para cada meio, possibilitando ajustes e melhorias ao longo do processo. Dessa forma, é possível identificar e resolver eventuais problemas que possam surgir durante os experimentos, garantindo que a avaliação de desempenho funcione corretamente. A divisão em etapas distintas permite realizar ajustes refinados durante a operação de melhora da hélice híbrida, possibilitando um ajuste mais preciso do funcionamento do drone híbrido, caso se veja necessário um maior desempenho em apenas um dos meios.

Este trabalho se propõe a realizar uma análise comparativa entre diversos métodos de otimização encontrados na literatura no contexto de hélices aquáticas. Entre os métodos avaliados estão *Differential Evolution* (DE), *Covariance Matrix Adaptation Evolution Strategy* (CMA-ES), *OpenAI Evolution Strategy* (OpenAI-ES), *Coordinate Descent* e Busca Aleatória, tendo por finalidade realizar a comparação do desempenho desses métodos de otimização e a qualidade das soluções obtidas.

4.1.1 Protocolo Experimental para Água

Para realizar o cálculo de propulsão da hélice na água, foi utilizado dados de uma balsa projetada para operação em um lago no sudoeste do Brasil, apresentados em (MARQUES, Crístofer H.; BELCHIOR; CAPRACE, 2019). Todos os experimentos utilizaram dos valores na Tabela 3 como parâmetros constantes de ambiente, embar-

cação e sistema de propulsão.

O v_S , foi o único parâmetro alterado em diferentes execuções, visto que representa a velocidade em qual a embarcação irá trabalhar, podendo ser ajustado dependendo das exigências da aplicação. Valores mais altos de v_S exigem maior desempenho da hélice, e assim torna mais complexo encontrar configurações que respeitem as restrições, ao mesmo tempo que aumenta o valor de P_B da hélice.

Como a alteração do valor de v_S significa alterar umas das configurações do meio em que a hélice está sendo testada, comparações entre configurações de hélices e heurísticas devem ser feitas somente para valores idênticos de v_S .

Os algoritmos deste estudo utilizaram exclusivamente os parâmetros listados na Tabela 1 como as variáveis a serem otimizadas. Inicialmente, os valores de velocidade da água (v_S) utilizados foram de 7,0 e 7,5 nós. Posteriormente os valores de v_S de 8,0 e 8,5 nós também foram utilizados para testes mais rigorosos para avaliar o desempenho das heurísticas em cenários desafiadores, visto que não foram possíveis encontrar configurações de hélices válidas para essas velocidades em trabalhos como (MARQUES, Crístoper H.; BELCHIOR; CAPRACE, 2019).

Tabela 3 – Parâmetros de entrada constantes.

| Parâmetro | Valor |
|--------------------------------|--|
| <i>Ambiente</i> | |
| Aceleração devido à gravidade | 9.80665 m/s ² |
| Pressão atmosférica | 101.325 kPa |
| Densidade da água | 999.0 kg/m ³ |
| Viscosidade cinemática da água | 1.139×10^{-6} m ² /s |
| Pressão de vapor da água | 1704 Pa |
| <i>Embarcação</i> | |
| Comprimento de linha d'água | 22.00 m |
| Largura | 9.600 m |
| Calado da embarcação | 1.089 m |
| Deslocamento volumétrico | 204.1 m ³ |
| Coeficiente de meia-nau | 0.9761 |
| Coeficiente do plano d'água | 0.9964 |
| Parâmetro de forma da popa | 10 |
| Número de hélices | 2 |
| <i>Propulsão</i> | |
| Centro da hélice | 0.500 m |
| Tensão permitida da hélice | 134.8 MPa |
| Área de cavitação permitida | 5% |
| Eficiência do eixo | 99.00% |

Fonte: Adaptado de (MARQUES, Crístoper H.; BELCHIOR; CAPRACE, 2019). Traduzido pelo autor.

Os experimentos foram executados em uma máquina com processador Intel

Core i5-11320H 8 cores, 8GB memória RAM, Ubuntu 22, Python 3.10, Octave 6.4. Os códigos para o cálculo da performance em água da hélice foram obtidos de (MARQUES, Crístofer H.; BELCHIOR; CAPRACE, 2019) em Octave. A implementação dos algoritmos foram feitos na linguagem Python, sendo utilizada a biblioteca *Oct2Py* para permitir a troca de informações entre Octave e Python. Cada teste de desempenho de configurações de hélice, levaram em média 10 segundos. Assim a execução de cada algoritmo, para uma *seed* de geração aleatória, realizando 900 avaliações, levaram em média 2 horas.

4.2 VARREDURA DO ESPAÇO DE BUSCA

Para estudar o funcionamento dos algoritmos e verificar a existência de valores viáveis, foi realizada uma exploração no espaço de busca dos parâmetros. Para cada parâmetro contínuo listado na Tabela 1, seu intervalo foi dividido em 50 segmentos, quantidade definida arbitrariamente com o objetivo de obter uma alta precisão. Quanto ao número de pás da hélice Z , o único parâmetro discreto, todos os valores possíveis dentro do intervalo foram considerados.

Foi realizada uma varredura no espaço de busca para três valores de v_S : 7,0, 7,5 e 8,0 nós. O número total de avaliações para cada exploração pode ser calculado usando a Equação (13). Onde o valor 50 representa a divisão do intervalo de cada parâmetro contínuo, elevado a 3 pela quantidade de parâmetros contínuos. Enquanto o valor 6 representa o número de configurações diferentes para o número de pás da hélice. A varredura sistemática envolve explorar cada possível combinação de parâmetros, o que resulta em um tempo de execução exponencial. O expoente desse tempo de execução é determinado pelo número de parâmetros contínuos utilizados. Foi realizado um total de 750.000 avaliações para cada valor de v_S . A multiplicação por 3 na Equação (13) se refere a quantidade de valores de v_S utilizados, resultando em 2.250.000 avaliações no total. Cada avaliação consistiu na configuração de uma hélice, com os parâmetros indicados na Tabela 1, e no armazenamento do respectivo valor de P_B encontrado.

$$\text{Avaliações Totais} = 3 \times (6 \times 50^3) \quad (13)$$

Após a conclusão da exploração, os dados coletados foram utilizados para gerar imagens com o propósito de analisar o comportamento dos parâmetros em relação aos valores de P_B obtidos. Além de permitir a comparação com os resultados encontrados pelas heurísticas, a validação das configurações e resultados obtidos.

4.3 EXPERIMENTOS INICIAIS COM BASE NA LITERATURA

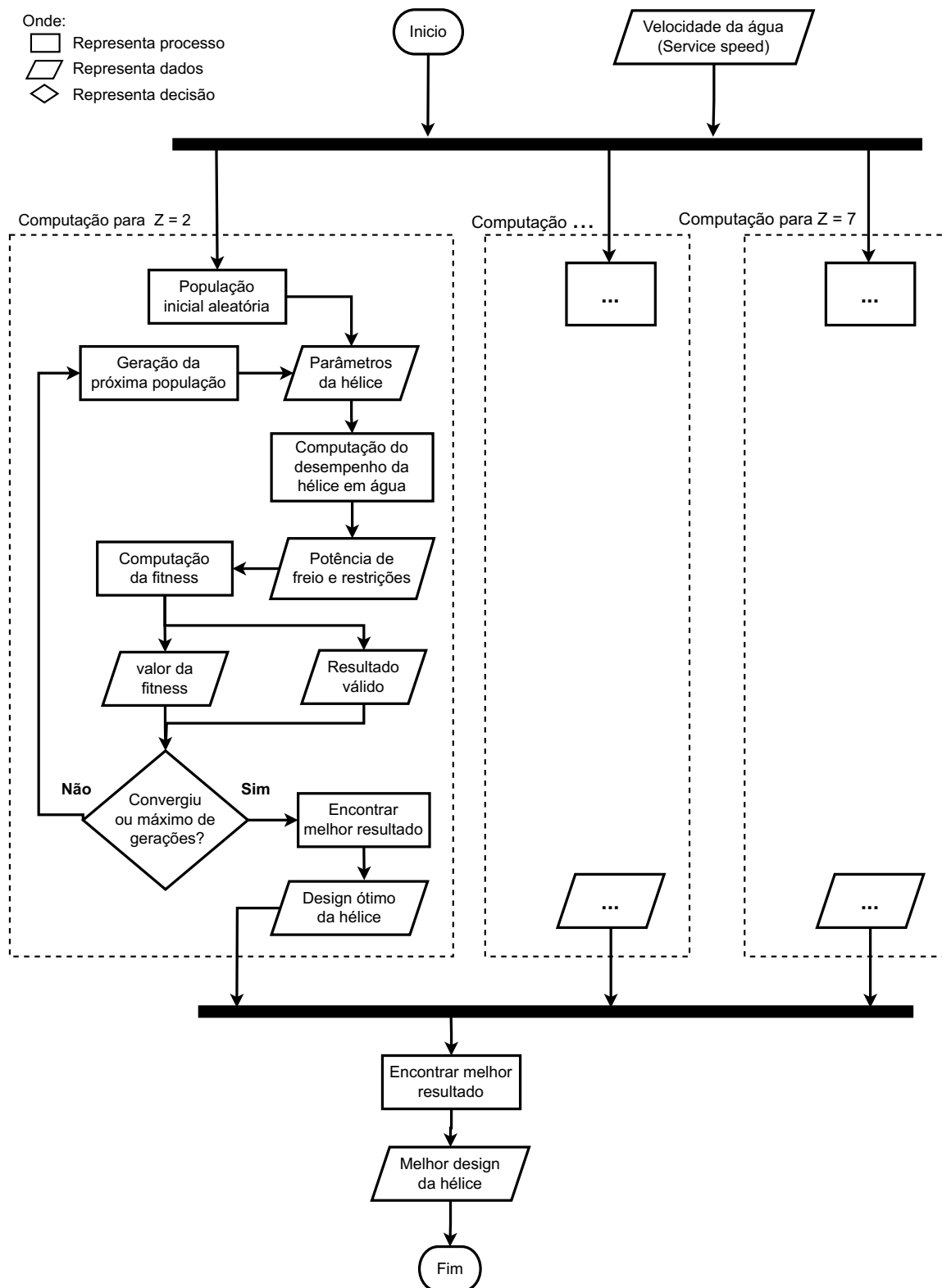
Inicialmente foi feita a réplica da execução do algoritmo DE utilizado em (MARGUES, Crístofer H.; BELCHIOR; CAPRACE, 2019) para validar a operação dos algoritmos. Essa implementação utilizou como função objetivo apenas o valor de P_B , não considerando hélices que estejam fora dos limites das restrições. O algoritmo DE foi executado com os parâmetros contidos na Tabela 4. Com o tamanho da população e quantidade de gerações, através da Equação (14) pode-se concluir que para cada execução do algoritmo são testadas 900 configurações de hélices, essa quantidade de avaliações foi considerada o orçamento computacional a ser aproximado por todos os algoritmos de otimização utilizados a fim de possibilitar uma comparação entre os resultados obtidos.

$$\text{Execuções} = \text{Tamanho da População} \times \text{Número de Gerações} \quad (14)$$

Neste trabalho, para os algoritmos CMA-ES, OpenAI-ES, *Coordinate Descent* e Busca Aleatória, para cada execução foram realizados 6 cenários distintos, considerando todos os possíveis valores da variável discreta Z , que corresponde ao número de pás da hélice. Em cada cenário o parâmetro Z possui um valor distinto no intervalo do parâmetro como consta na Tabela 1. Dessa forma foi possível reduzir o número de parâmetros utilizados pelas heurísticas, além de permitir a execução simultânea dos 6 cenários. Ao final de cada execução, a melhor solução encontrada dentre 6 cenários foi considerada como o resultado encontrado pela heurística para aquela execução. Esse funcionamento está representado pelo diagrama de fluxo na Figura 6.

Para conduzir os experimentos, foram realizadas 10 execuções independentes para cada heurística, em que cada execução utilizou uma semente distinta para a geração de números aleatórios, utilizada para a criação da população inicial e durante o funcionamento das heurísticas, a fim de adicionar variabilidade entre as execuções. Essa abordagem foi aplicada a cada valor distinto de v_S : 7,0 e 7,5.

Figura 6 – Diagrama de Fluxo de Execução das Heurísticas.



Fonte: Elaborada pelo autor.

4.3.1 Função Objetivo

A função objetivo busca encontrar os parâmetros de configuração da hélice que gerem o melhor desempenho, portanto é desejado encontrar o menor valor de P_B que gere uma hélice dentro das restrições explicadas em 2.1.2. Enquanto o DE descrito em (MARQUES, Crístofer H.; BELCHIOR; CAPRACE, 2019) utiliza apenas o valor de P_B e não considera as hélices geradas que estejam foras das restrições durante o processo de otimização, essa abordagem se demonstrou falha para encontrar consistentemente hélices com valores válidos para valores de v_S iguais ou acima de 8,0.

Este trabalho propõe adaptação da função objetivo, mantendo o objetivo principal encontrar o menor valor de P_B , mas levando em consideração o quanto a hélice ultrapassa as restrições, assim hélices que não são viáveis também participam do processo de otimização junto com as válidas, porém com uma penalidade para não serem candidatas tão desejáveis quanto as hélices dentro das restrições.

As restrições consideradas neste trabalho seguem as seguintes limitações:

- Para evitar a **Cavitação** é necessário que o coeficiente de carga de empuxo ($\sigma_{0.7R}$) não ultrapasse o coeficiente de carga de empuxo máximo ($\sigma_{c0.7R,max}$).
- Para evitar problemas a **Velocidade periférica** na ponta da lâmina (v_{tip}) não pode ultrapassar a velocidade máxima de 39 m/s.
- Para manter a **Resistência da lâmina da hélice**, é necessário que a espessura da lâmina a 75% do raio da hélice ($t_{0.75R}$) não seja inferior à espessura mínima exigida ($t_{0.75R,min}$).

Para que uma hélice seja considerada válida deve respeitar as Equações (15), (16) e (17) referentes a cavitação, velocidade periférica e resistência da lâmina respectivamente.

$$\sigma_{0.7R} \leq \sigma_{0.7R,max} \quad (15)$$

$$v_{tip} \leq 39 \quad (16)$$

$$t_{0.75R} \geq t_{0.75R,min} \quad (17)$$

A Equação (18) foi utilizada para gerar o valor da fitness, onde é feito o cálculo da razão da diferença entre o valor da restrição e seu limite, pelo limite restrição, dessa forma se obtém a razão do quanto cada restrição ultrapassa o seu limite. Quando uma restrição está dentro do limite, esse valor não é considerado, pois neste caso o valor da razão é um número negativo, e o cálculo do máximo entre esse valor e 0, faz com que 0 seja escolhido.

É realizada uma soma que inclui a razão de todas as restrições, somada a 1. Nos casos em que a configuração da hélice excede os limites das restrições, essa soma

ultrapassa 1, enquanto as hélices que permanecem dentro dos limites têm uma soma igual a 1. O resultado dessa soma é então multiplicado pelo valor de P_B . Para hélices válidas, o valor da fitness é igual ao valor do P_B da hélice, enquanto configurações que ultrapassam os limites das restrições recebem valores de fitness menos desejáveis.

Foi utilizado o valor da fitness negativo, considerando que a implementação dos algoritmos tem o objetivo de maximizar a função de fitness. Assim, como o objetivo é diminuir o valor de P_B da hélice, menores valores de fitness negativos para as heurísticas representam soluções com desempenho superior.

$$\begin{aligned} \text{fitness} = -P_B \times & \left(1 + \max \left(\frac{\sigma_{0.7R} - \sigma_{c0.7R,max}}{\sigma_{c0.7R,max}}, 0 \right) \right. \\ & + \max \left(\frac{v_{tip} - 39}{39}, 0 \right) \\ & \left. + \max \left(\frac{t_{0.75R,min} - t_{0.75R}}{t_{0.75R,min}}, 0 \right) \right) \end{aligned} \quad (18)$$

Esta abordagem proposta para o cálculo da fitness, visa permitir que as heurísticas explorem mais amplamente o espaço dos parâmetros, possibilitando que ao longo das gerações, as soluções se aproximem de configurações de hélice válidas e melhores.

4.3.2 Implementação dos Algoritmos

Essa próxima parte do trabalho é dedicada à explicação da implementação dos algoritmos. Foram realizadas 10 execuções para cada algoritmo, guardando o melhor resultado encontrado em cada execução, a fim de realizar uma análise comparativa posteriormente.

4.3.2.1 Implementação do DE Base

O algoritmo DE Base foi replicado com as mesmas configurações de (MARQUES, Crístofer H.; BELCHIOR; CAPRACE, 2019), utilizando os parâmetros contidos na Tabela 4. Essa implementação utilizou como função objetivo apenas o valor de P_B , não considerando hélices que estejam fora dos limites das restrições.

4.3.2.2 DE Modificado

O algoritmo DE Modificado utiliza os mesmos valores parâmetros contidos na Tabela 4, entretanto a função objetivo utiliza o cálculo da fitness da Equação (18). Desta forma utilizando hélices que estejam fora dos limites das restrições no processo de avaliação.

Tabela 4 – Parâmetros de entrada no DE

| Parâmetro | Valor |
|----------------------|-------|
| Tamanho da População | 30 |
| Gerações | 30 |
| Fator de Crossover | 0.5 |
| Fator de Mutação | 0.8 |

Fonte: Elaborada pelo autor.

4.3.2.3 Implementação do OpenAI-ES e CMA-ES

A implementação do CMA-ES foi baseada em (HANSEN, 2023) com a utilização da biblioteca de Python "pymcma". Gerando a população inicial com valores aleatórios dentro da faixa de cada parâmetro presente na Tabela 1.

A implementação do OpenAI-ES foi baseada em (HA, 2017b), que utiliza uma classe que implementa os métodos de geração da população. O algoritmo é iniciado com os valores presentes na tabela 5.

Em ambos os algoritmos a geração da população inicial foi feita com valores aleatórios dentro da faixa de cada parâmetro contínuo presente na Tabela 1. Foram realizadas 6 cenários, um para cada valor do intervalo de Z , assim realizando 150 avaliações para cada número de pás de hélices, totalizando 900 avaliações para cada execução. Ao final, de cada execução a melhor solução válida, i. e. dentro dos limites das restrições, encontrada entre os cenários, foi selecionada como o resultado da execução.

Tabela 5 – Parâmetros de descrição dos algoritmos OpenAI-ES e CMA-ES.

| Parâmetro | Significado | OpenAI-ES | CMA-ES |
|---------------------|-----------------------------------|-----------|--------|
| n params | Número de parâmetros melhorados | 3 | 3 |
| popsiz | Tamanho da população | 5 | 5 |
| max iteration | Gerações | 30 | 30 |
| weight decay | Decaimento dos pesos | 0.01 | 0.01 |
| sigma init | Desvio padrão inicial | 0.1 | 0.1 |
| sigma decay | Decaimento do desvio padrão | 0.99 | |
| antithetic sampling | Amostragem antitética | False | |
| forget best result | Esquecer melhor resultado | False | |
| learning rate | Taxa de aprendizado | 0.1 | |
| learning rate decay | Decaimento da taxa de aprendizado | 0.99 | |

Fonte: Elaborada pelo autor.

4.3.2.4 Implementação do *Coordinate Descent*

A implementação do *Coordinate Descent* segue a abordagem de (LI; OSHER, 2009), onde para cada valor do intervalo de Z foi criado um cenário. Em cada cenário

o algoritmo inicia com valores aleatórios dentro dos limites dos parâmetros contínuos da Tabela 1.

Em cada iteração, para cada um dos parâmetros a serem otimizados, o algoritmo percorre linearmente o espaço de busca do parâmetro selecionado, enquanto os demais parâmetros são mantidos fixos. Desta forma o algoritmo percorre linearmente o intervalo do parâmetro com passos do tamanho "step size" presente na Tabela 6. Após percorrer o intervalo do parâmetro selecionado, é escolhido o valor que encontrou o melhor resultado válido. Tendo cada nova iteração o valor de "step size" diminuído com base no "step decay" com a finalidade de convergir o espaço de busca. Assim o algoritmo avalia "params" parâmetros contínuos.

Esse processo é repetido até atingir o número máximo de iterações dado por "max iterations", calculado com base no número de parâmetros e quantidade de orçamento computacional.

Tabela 6 – Parâmetros de descrição *Coordinate Descent*.

| Parâmetro | Significado | Valor |
|-------------------|--|-------|
| params | Número de parâmetros a serem melhorados | 3 |
| step size inicial | Tamanho inicial do passo ao percorrer um parâmetro | 0.1 |
| step decay | Decaimento do passo a cada iteração | 0.1 |
| max iterations | Máximo de iterações | 30 |

Fonte: Elaborada pelo autor.

4.3.2.5 Implementação da busca aleatória

O algoritmo de busca aleatória executa a função avaliação para cada ponto selecionado aleatoriamente dentro dos limites de cada parâmetro (KARNOPP, 1963). Sendo realizadas 150 avaliações para cada cenário, correspondente a um valor do intervalo de Z , totalizando 900 avaliações. Ao final, a melhor solução válida encontrada, foi selecionada como o resultado final.

4.4 EXPERIMENTOS EXCLUSIVOS COM ESTRATEGIAS EVOLUTIVAS

Este trabalho tem como proposta a utilização de algoritmos de Estratégias Evolutivas (EE) na otimização dos parâmetros de hélices. Após a execução dos algoritmos no cenário inicial, foi realizado um estudo mais aprofundado, direcionado especificamente para os algoritmos de EE.

4.4.1 Estudo Sobre Tamanho da População e Número de Gerações

Dado o impacto significativo que o tamanho da população e o número máximo de gerações têm na execução de algoritmos de Estratégias Evolutivas, é essencial encontrar um equilíbrio entre esses parâmetros. O aumento no tamanho da população

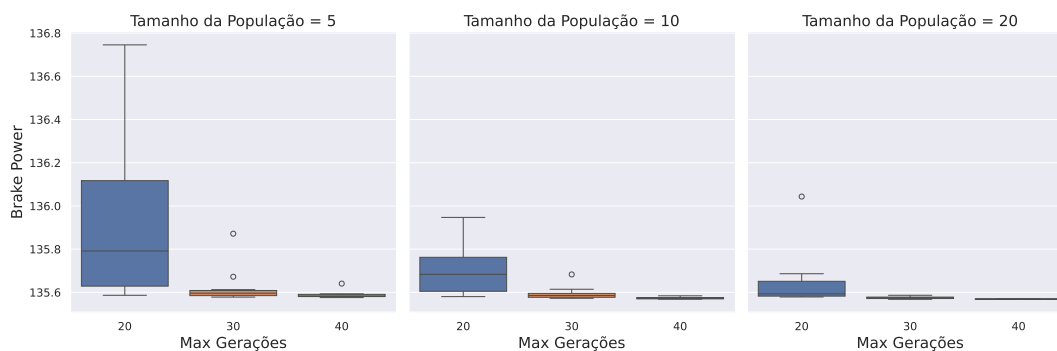
e o número de gerações podem trazer um melhor desempenho, mas também resulta em um aumento considerável no tempo de execução, devido ao maior número de casos a serem avaliados, como ilustrado na Equação (14).

Para encontrar valores ideais de tamanho da população e o número de gerações, específicos para o problema sendo testado, foi realizado um breve estudo para os algoritmos CMA-ES e OpenAI-ES. Para o algoritmo DE, esses aspectos já haviam sido testados e discutidos em estudos anteriores para o mesmo problema, em (MARQUES, Crístofer H.; BELCHIOR; CAPRACE, 2019). Foram realizados 10 execuções para cada conjunto de parâmetros com v_S de 8,0 nós. A Tabela 7 demonstra a quantidade de testes executados para cada configuração de parâmetros.

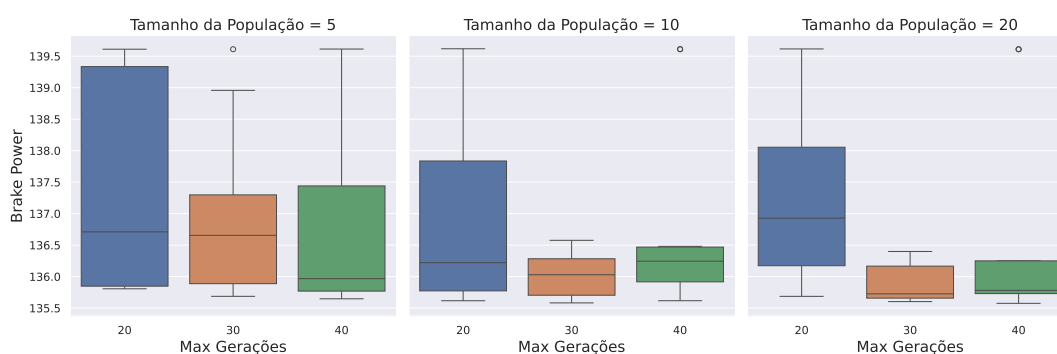
Tabela 7 – Quantidade de avaliações por Tamanho da População e Gerações.

| Máximo de Gerações | Tamanho da População | | |
|--------------------|----------------------|-----|-----|
| | 5 | 10 | 20 |
| 20 | 100 | 200 | 400 |
| 30 | 150 | 300 | 600 |
| 40 | 200 | 400 | 800 |

Fonte: Elaborada pelo autor.



(a) CMA-ES



(b) OpenAI-ES

Figura 7 – Estudo do Tamanho da População e Máximo de Gerações

Fonte: Elaborada pelo autor.

As Figuras 7a e 7b demonstram os resultados encontrados pelos algoritmos CMA-ES e OpenAI-ES respectivamente. Tamanhos maiores na população apresentam um aumento na uniformidade nos resultados, enquanto o aumento no máximo de gerações melhora a média dos resultados obtidos.

Considerando a Tabela 7 com quantidade de testes executados para cada conjunto de máximo de gerações com tamanho da população, foi escolhido prosseguir com tamanho da população de 10, e máximo de gerações de 30, por apresentarem uma menor variação entre os resultados quando comparados com o tamanho de população de 5 indivíduos. Como pode ser visto na Tabela 7 com a mudança de tamanho da população de 10 para 20, causa o dobro de testes de indivíduos a serem executados, enquanto que pode ser observado na Figura 7 que a qualidade dos resultados não teve um impacto significativo. Por isso a escolha de prosseguir com tamanho da população de 10 indivíduos.

Foi mantido o número de gerações em 30, visto que o aumento na quantidade das gerações gera um aumento no tempo de execução dos algoritmos, pois cada geração depende do resultado da análise de todos os indivíduos da geração anterior, e como o aumento na quantidade de gerações não apresentou impacto significativo na qualidade dos resultados, se manteve o máximo de gerações no valor de 30, para os testes seguintes.

4.4.2 Descrição dos Experimentos

Foram realizados mais experimentos com os algoritmos de DE Modificado, CMA-ES Modificado e OpenAI-ES Modificado. Com 10 execuções independentes para cada algoritmo, em que cada execução utilizou uma semente distinta para a geração de números aleatórios, utilizada para a criação da população inicial e durante o funcionamento das heurísticas. Cada execução com um cenário para cada um dos 6 valores do intervalo do parâmetro Z . Essa abordagem foi aplicada a cada valor distinto de v_S : 7,0, 7,5, 8,0 e 8,5.

4.4.2.1 OpenAI-ES e CMA-ES Modificados

Foi utilizada a mesma implementação dos algoritmos de OpenAI-ES e CMA-ES descritas em 4.3.2.3, porém com a modificação nos parâmetros de entrada como demonstra a Tabela 8, para o tamanho da população de 10 indivíduos.

Tabela 8 – Parâmetros modificados de descrição dos algoritmos OpenAI-ES e CMA-ES.

| Parâmetro | Significado | OpenAI-ES | CMA-ES |
|---------------------|-----------------------------------|------------------|---------------|
| n params | Número de parâmetros | 3 | 3 |
| popsize | Tamanho da população | 10 | 10 |
| max iteration | Gerações | 30 | 30 |
| weight decay | Decaimento dos pesos | 0.01 | 0.01 |
| sigma init | Desvio padrão inicial | 0.1 | 0.1 |
| sigma decay | Decaimento do desvio padrão | 0.99 | |
| antithetic sampling | Amostragem antitética | False | |
| forget best result | Esquecer melhor resultado | False | |
| learning rate | Taxa de aprendizado | 0.1 | |
| learning rate decay | Decaimento da taxa de aprendizado | 0.99 | |

Fonte: Elaborada pelo autor.

5 ANÁLISE DE RESULTADOS

Neste capítulo será discutido os resultados obtidos dos experimentos realizados. Primeiramente a seção 5.1 apresenta os resultados da Varredura do Espaço de Busca. Enquanto a seção 5.2 trás os resultados dos experimentos com as heurísticas de otimização. Por último a seção 5.3 apresenta os resultados dos experimentos feitos exclusivamente para Algoritmos Evolutivos.

5.1 RESULTADO DA VARREDURA DO ESPAÇO DE BUSCA

Após realizar a varredura sobre o espaço de busca dos parâmetros, foi feita a construção de gráficos para analisar as soluções viáveis encontradas, organizadas de acordo com o valor de v_S . Esse processo permitiu constatar que existem soluções viáveis para v_S de 8,0 nós, não detectadas pelos algoritmos de otimização em estudos anteriores como (MARQUES, Crístofer H.; BELCHIOR; CAPRACE, 2019). A discrepância nos resultados torna-se evidente quando é comparado a abrangência que soluções válidas aparecem no gráfico entre diferentes valores de v_S , na qual a quantidade de resultados diminui com o aumento em v_S .

As Figuras 8a, 8b e 9a apresentam gráficos de v_S 7,0, 7,5 e 8,0 nós respectivamente. Em que cada eixo representa um dos parâmetros contínuos da Tabela 1. As cores dos pontos nos gráficos indicam os valores de P_B , com cores mais claras correspondendo a valores menores, e portanto mais desejáveis. Pode-se observar que os melhores valores de P_B estão próximos do limite superior do diâmetro da hélice.

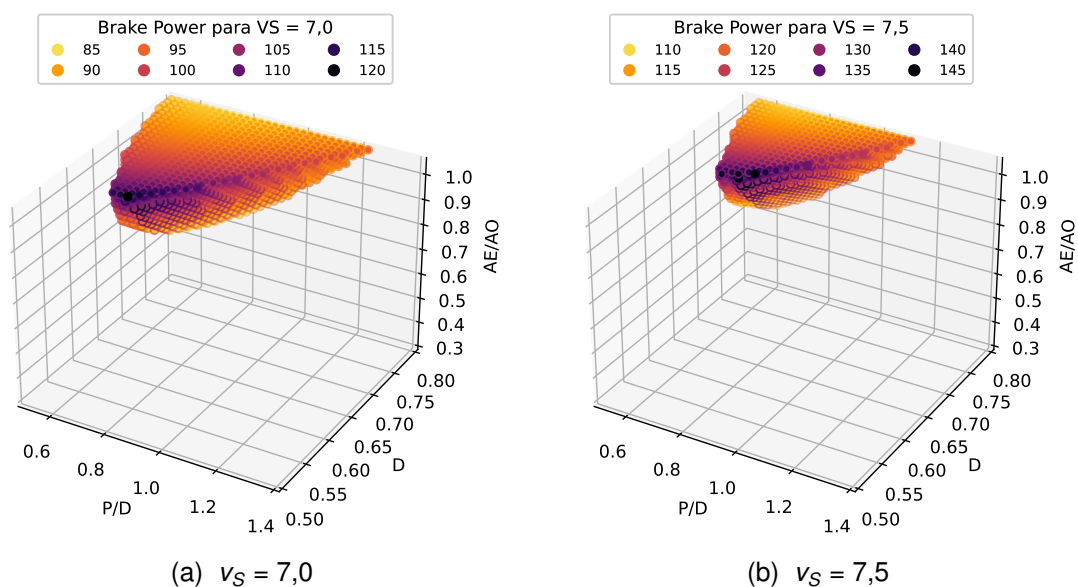


Figura 8 – P_B encontrados para os conjuntos de parâmetros testados.

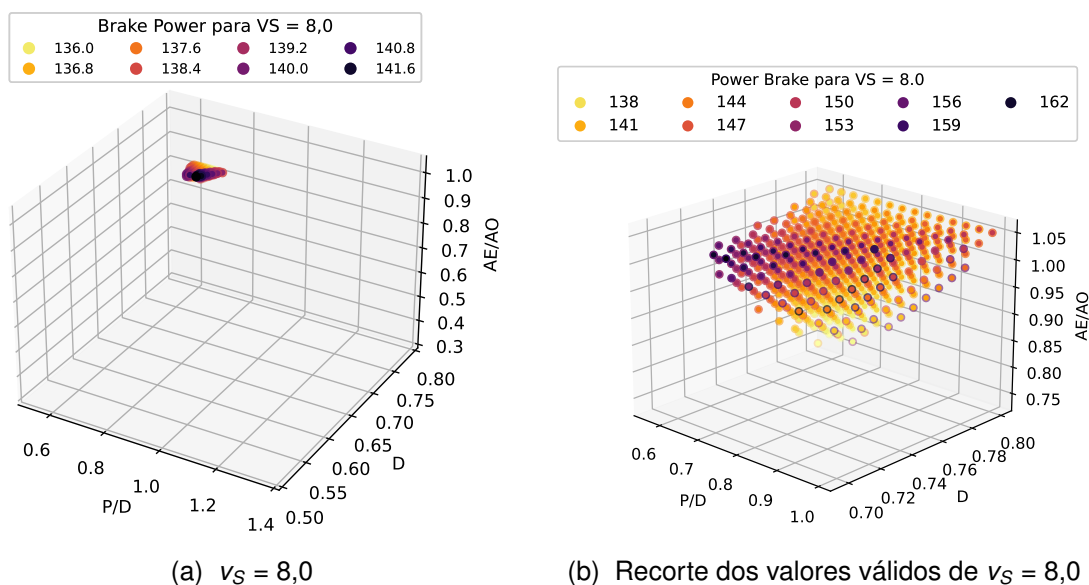


Figura 9 – P_B encontrados para os conjuntos de parâmetros testados.

As Figura 9a e 9b são referentes a v_S de 8,0 nós, com a Figura 9b sendo um recorte somente dos valores válidos da Figura 9a. Se pode observar uma disponibilidade limitada de parâmetros que levam a resultados válidos, sendo que estão concentradas em uma pequena região específica no espaço dos parâmetros.

5.2 RESULTADOS DAS HEURÍSTICAS

Após a execução dos algoritmos DE Base, DE Modificado, CMA-ES, OpenAI-ES, *Coordinate Descent* e Busca Aleatória, explicada em 4.3, foram criados as Figuras 10a e 10b para demonstrar os resultados para os valores de v_S de 7,0 e 7,5 nós respectivamente.

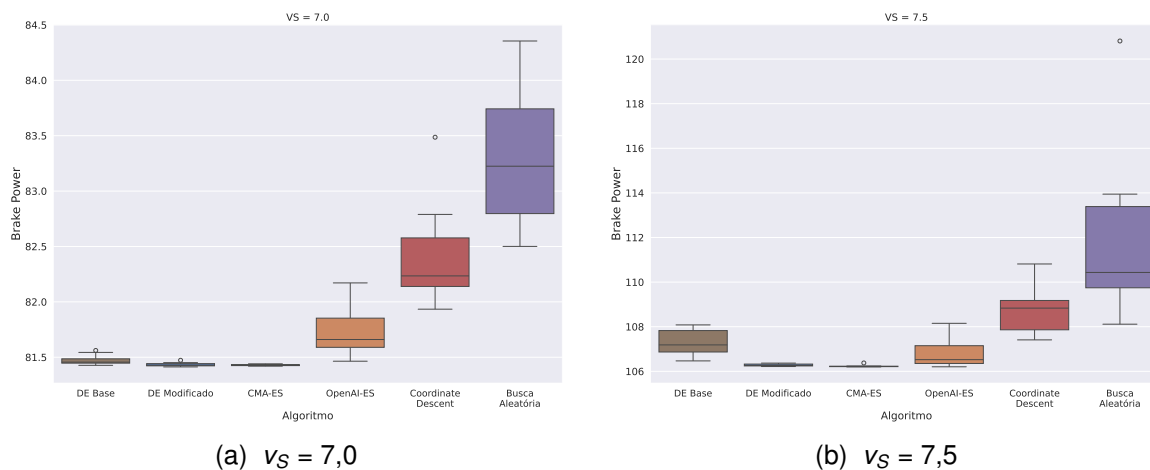


Figura 10 – Valores encontrados pelas execuções.

Fonte: Elaborada pelo autor.

Examinando a variabilidade dos resultados obtidos pelos algoritmos ao longo de dez execuções percebe-se que os resultados dos algoritmos evolutivos (AE) foram semelhantes, com o algoritmo CMA-ES consistentemente apresentando o desempenho mais conciso. Ambas implementações do *Differential Evolution* apresentaram resultados semelhantes, entretanto a versão modificada se mostrou melhor, possuindo uma menor variabilidade. Os algoritmos de *Coordinate Descent* e Busca Aleatória apresentaram o pior desempenho em ambas velocidades, tanto obtendo os piores valores, quanto a maior variabilidade entre os algoritmos.

Os melhores valores encontrados de P_B , em kilowatt (kW), dentre as dez execuções de cada algoritmo para cada v_S de 7,0 e 7,5 estão na Tabela 9, juntamente com os parâmetros da série de "Wageningen B-Screw" obtidos pelas heurísticas.

Tabela 9 – Parâmetros dos melhores resultados encontrados pelas heurísticas para valores de v_S de 7,0 e 7,5.

| v_S (nós) | Algoritmo | P_B | D | A_E/A_O | P/D | Z |
|-------------|--------------------|----------------|-------|-----------|-------|-----|
| 7,0 | DE Base | 81,427 | 0,8 | 0,641 | 0,674 | 5 |
| | DE Modificado | 81,413 | 0,8 | 0,632 | 0,669 | 5 |
| | CMA-ES | 81,419 | 0,8 | 0,631 | 0,667 | 5 |
| | OpenAI-ES | 81,464 | 0,8 | 0,640 | 0,658 | 5 |
| | Coordinate Descent | 81,933 | 0,8 | 0,680 | 0,692 | 6 |
| | Busca Aleatória | 82,501 | 0,8 | 0,737 | 0,749 | 6 |
| 7,5 | DE Base | 106,468 | 0,8 | 0,718 | 0,641 | 5 |
| | DE Modificado | 106,215 | 0,8 | 0,699 | 0,653 | 5 |
| | CMA-ES | 106,198 | 0,8 | 0,701 | 0,657 | 5 |
| | OpenAI-ES | 106,204 | 0,8 | 0,704 | 0,661 | 5 |
| | Coordinate Descent | 107,410 | 0,796 | 0,803 | 0,671 | 6 |
| | Busca Aleatória | 108,114 | 0,798 | 0,742 | 0,687 | 6 |

Fonte: Elaborada pelo autor.

Para v_S de 7,0 o algoritmo DE com o cálculo de fitness modificado encontrou o melhor resultado. Enquanto para v_S de 7,5 o CMA-ES obteve o melhor resultado.

Vale ressaltar que para a v_S de 7,0 todas as configurações alcançaram o mesmo valor de D . Todos os resultados em cada velocidade apresentam valores de P/D e A_E/A_O com pouca variação, assim como valores próximos de Z .

5.3 RESULTADOS DOS EXPERIMENTOS EXCLUSIVOS DE ESTRATÉGIAS EVOLUTIVAS

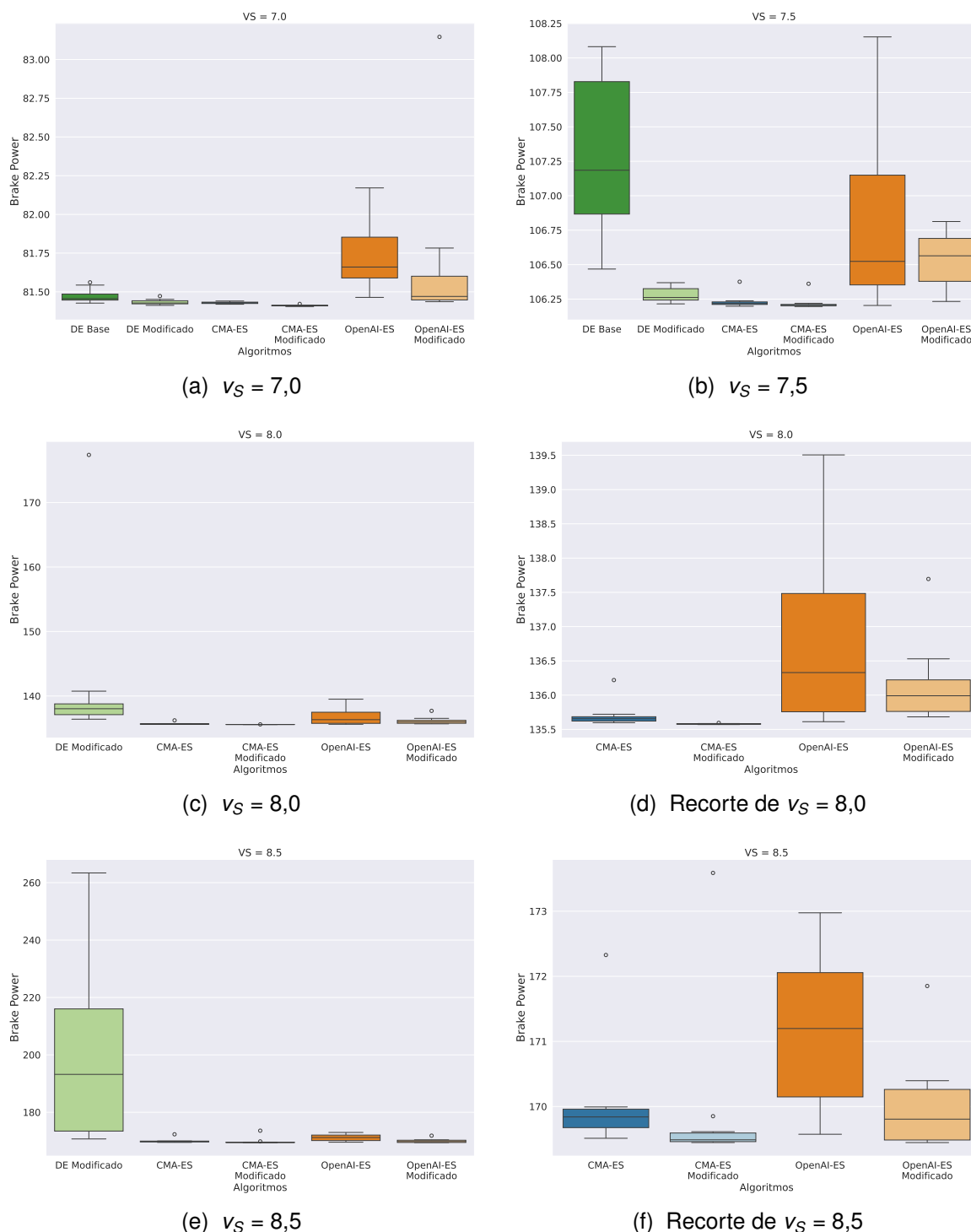


Figura 11 – Valores encontrados pelas execuções.

Fonte: Elaborada pelo autor.

A Figura 11 demonstra os resultados para os valores de v_S de 7,0, 7,5, 8,0 e 8,5 nós obtidos após a execução dos algoritmos DE Base, DE Modificado, CMA-ES,

CMA-ES Modificado, OpenAI-ES, OpenAI-ES Modificado, explicados em 4.4.

Examinando os resultados obtidos pelos algoritmos ao longo de dez execuções percebe-se que apesar dos desempenhos semelhantes os resultados dos algoritmos modificados geraram uma melhora na variabilidade e nos resultados. Especialmente o algoritmo OpenAI-ES modificado consistentemente apresentando o desempenho mais conciso do que a implementação com menor população. Notavelmente, exceto o DE Base, todos os demais algoritmos conseguiram encontrar resultados com hélices válidas, i. e. com as restrições dentro dos limites para todos os valores de v_S .

A Tabela 10 apresenta os melhores valores encontrados de P_B , em kilowatt (kW), dentre as dez execuções de cada algoritmo para cada v_S de 7,0, 7,5, 8,0 e 8,5 nós, juntamente com os parâmetros da série de "Wageningen B-Screw" obtidos pelos algoritmos otimizadores.

Tabela 10 – Parâmetros dos melhores resultados encontrados pelas heurísticas para todos os valores de v_S .

| v_S (nós) | Algoritmo | P_B | D | A_E/A_O | P/D | Z |
|-------------|----------------------|----------------|-------|-----------|-------|-----|
| 7,0 | DE Base | 81,427 | 0,8 | 0,641 | 0,675 | 5 |
| | DE Modificado | 81,413 | 0,8 | 0,632 | 0,669 | 5 |
| | CMA-ES | 81,419 | 0,8 | 0,631 | 0,667 | 5 |
| | CMA-ES Modificado | 81,405 | 0,8 | 0,631 | 0,670 | 5 |
| | OpenAI-ES | 81,464 | 0,8 | 0,640 | 0,658 | 5 |
| | OpenAI-ES Modificado | 81,436 | 0,8 | 0,642 | 0,675 | 5 |
| 7,5 | DE Base | 106,468 | 0,8 | 0,717 | 0,641 | 5 |
| | DE Modificado | 106,215 | 0,8 | 0,699 | 0,653 | 5 |
| | CMA-ES | 106,198 | 0,8 | 0,700 | 0,657 | 5 |
| | CMA-ES Modificado | 106,194 | 0,8 | 0,704 | 0,663 | 5 |
| | OpenAI-ES | 106,204 | 0,8 | 0,704 | 0,660 | 5 |
| | OpenAI-ES Modificado | 106,233 | 0,8 | 0,709 | 0,664 | 5 |
| 8,0 | DE Base | não encontrou | - | - | - | - |
| | DE Modificado | 136,402 | 0,8 | 0,881 | 0,743 | 6 |
| | CMA-ES | 135,624 | 0,8 | 0,791 | 0,667 | 5 |
| | CMA-ES Modificado | 135,597 | 0,8 | 0,809 | 0,680 | 6 |
| | OpenAI-ES | 135,612 | 0,8 | 0,788 | 0,656 | 5 |
| | OpenAI-ES Modificado | 135,683 | 0,8 | 0,823 | 0,689 | 6 |
| 8,5 | DE Base | não encontrou | - | - | - | - |
| | DE Modificado | 170,711 | 0,8 | 0,962 | 0,701 | 6 |
| | CMA-ES | 169,529 | 0,799 | 0,891 | 0,669 | 6 |
| | CMA-ES Modificado | 169,522 | 0,882 | 0,895 | 0,655 | 6 |
| | OpenAI-ES | 169,573 | 0,8 | 0,877 | 0,646 | 6 |
| | OpenAI-ES Modificado | 169,446 | 0,8 | 0,981 | 0,670 | 6 |

Fonte: Elaborada pelo autor.

O CMA-ES Modificado obteve o melhor resultado consistentemente para maioria

dos valores de v_S , porém OpenAI-ES Modificado encontrou o melhor resultado para a velocidade mais alta testada. Entretanto como pode ser observado na Tabela 10 todas as configurações apresentam valores similares de D e Z . Principalmente entre as mesmas velocidades os valores de P/D e A_E/A_O apresentam pouca variação entre algoritmos.

5.3.1 Convergência na Fitness

A Figura 12 apresenta o valor da fitness nos melhores resultados encontrados a cada geração, por cada algoritmo para as execuções que encontraram os melhores resultados, como constam na Tabela 10. Percebe-se que os algoritmos CMA-ES e OpenAI-ES não modificados começam nas primeiras gerações apresentando os piores resultados, em decorrência do menor número de indivíduos na população. Entretanto o DE Modificado é o único algoritmo que leva consideravelmente mais gerações do que os outros para convergir.

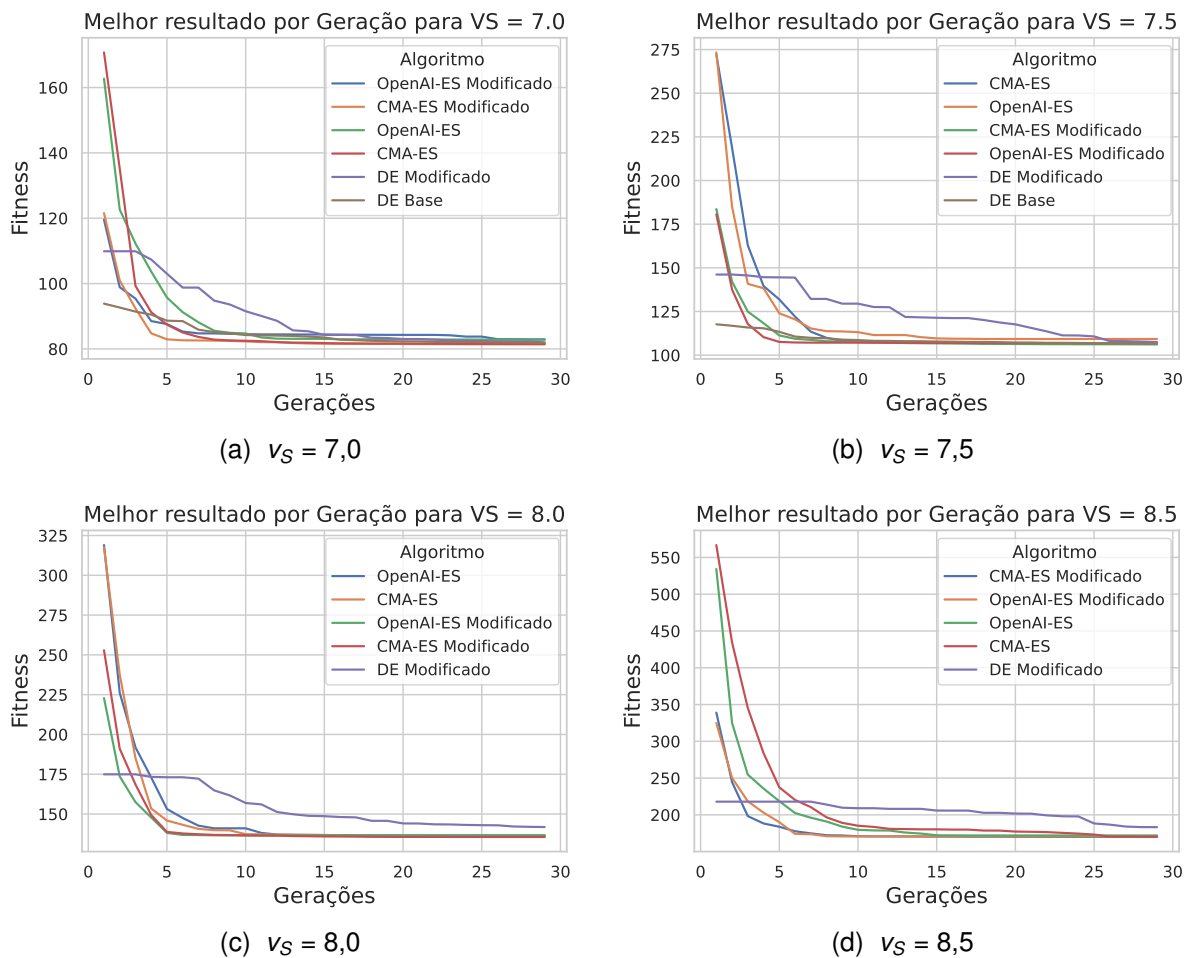


Figura 12 – Melhor Fitness encontrada por geração.

Fonte: Elaborada pelo autor.

5.3.2 Quantidade de indivíduos válidos por geração

A Figura 13 apresenta a quantidade de soluções candidatas válidas encontradas a cada geração para cada algoritmo, considerando pelo menos uma diferença maior do que 1% para cada parâmetro entre soluções candidatas da mesma geração. Para CMA-ES e OpenAI-ES, foi somado o número de soluções candidatas entre os 6 cenários, correspondentes aos diferentes quantidades de pás na hélice. Sendo assim, o CMA-ES Modificado e OpenAI-ES Modificado tem a possibilidade de alcançar no máximo 60 soluções válidas e diferentes em uma mesma geração devido ao dobro do tamanho da população por geração em comparação aos outros algoritmos, que podem alcançar no máximo 30 soluções válidas e diferentes em uma mesma geração.

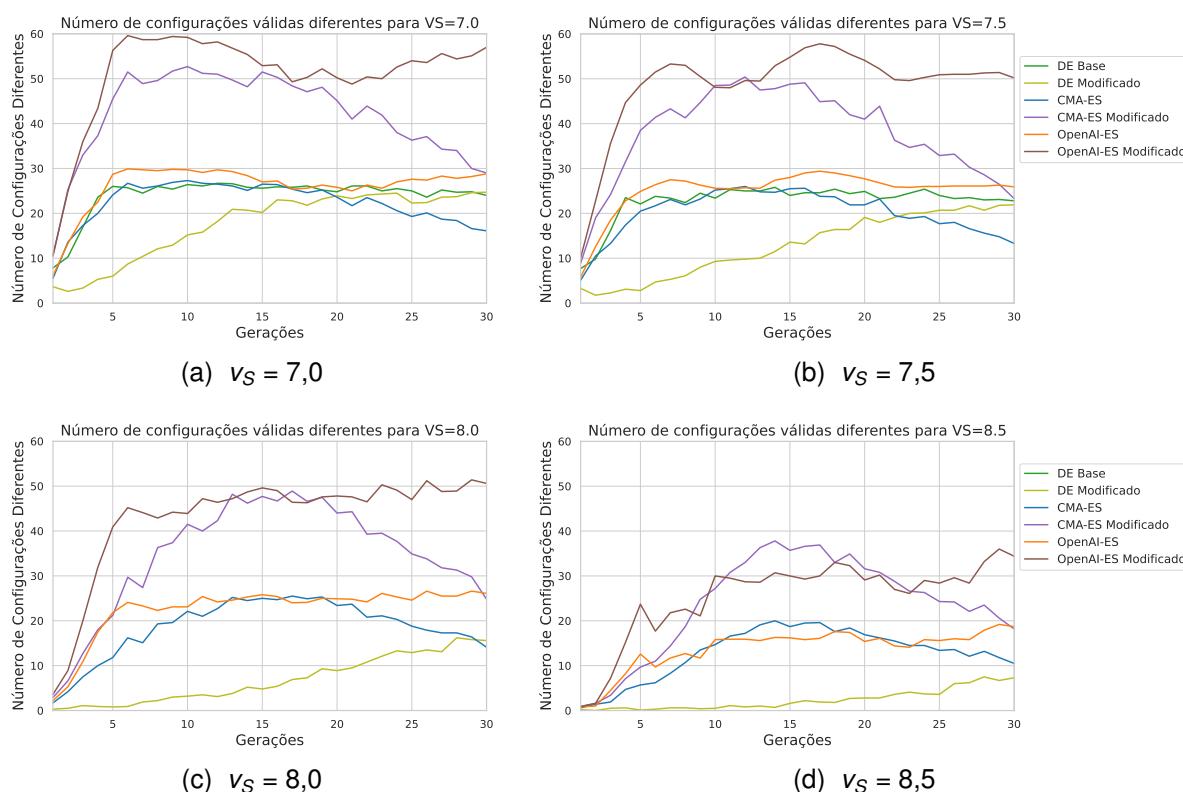


Figura 13 – Número de configurações válidas diferentes por geração.

Fonte: Elaborada pelo autor.

Como pode ser observado na Figura 13, os algoritmos começam uma pequena quantidade de soluções válidas encontradas, e este número tende a crescer à medida que as gerações avançam, eventualmente alcançando uma estabilização. A quantidade de gerações até se estabilizar a quantidade de soluções candidatas válidas difere principalmente por v_S , mas também por algoritmo. Assim é desejável algoritmos que sejam mais rápidos para encontrar soluções candidatas válidas.

Nas Figuras 13c e 13d são apresentados os valores de v_S não triviais, em que o DE Base não conseguiu encontrar soluções válidas. Assim se pode observar uma

menor quantidade de soluções candidatas válidas nas gerações em comparação com os outros algoritmos.

Para todos valores de v_S é possível observar na Figura 13 que o número de soluções candidatas válidas para o Para CMA-ES Modificado e OpenAI-ES Modificado são aproximadamente o dobro das versões com metade do tamanho da população.

Entretanto, na Figura 13 é perceptível que ambas as implementações de CMA-ES apresentam uma queda na quantidade de soluções encontradas diferentes durante a metade final das gerações. Sendo assim a Figura 14 apresenta em linhas contínuas as configurações diferentes, enquanto que as linhas tracejadas representam a quantidade total de soluções válidas, podendo ser muitos próximas ou até iguais entre a população.

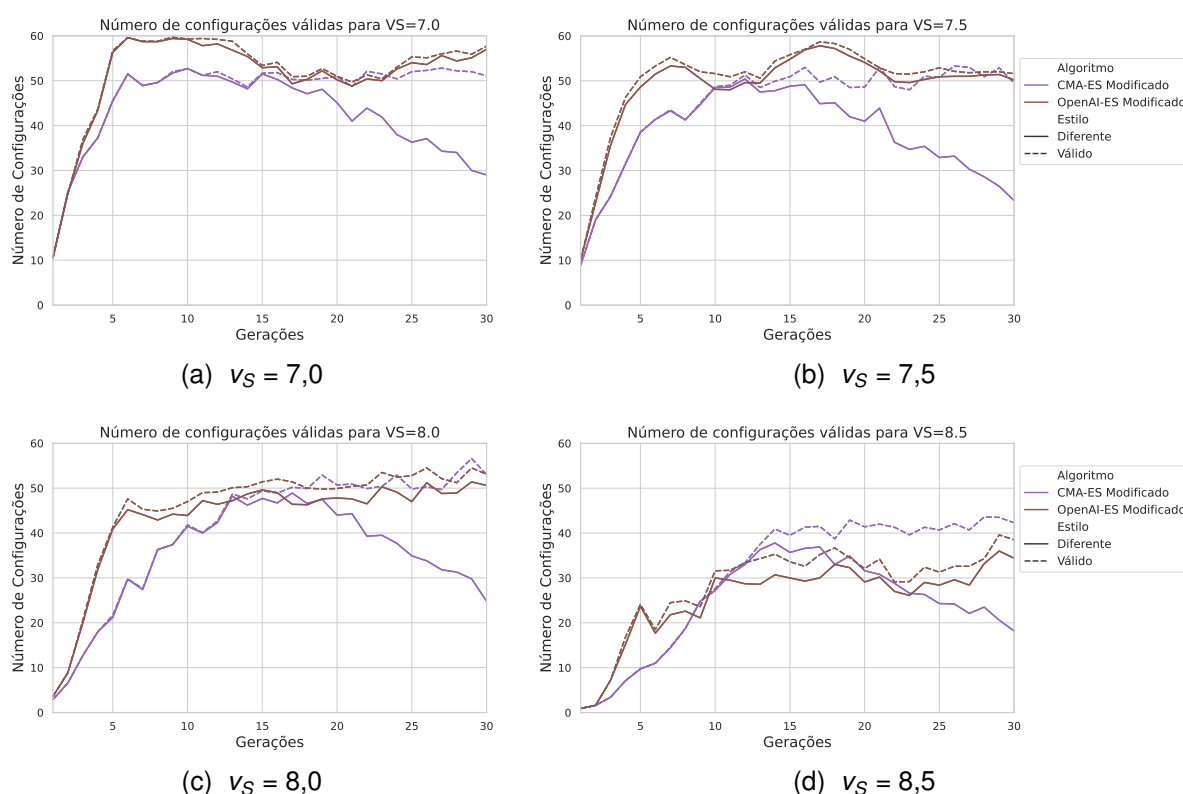


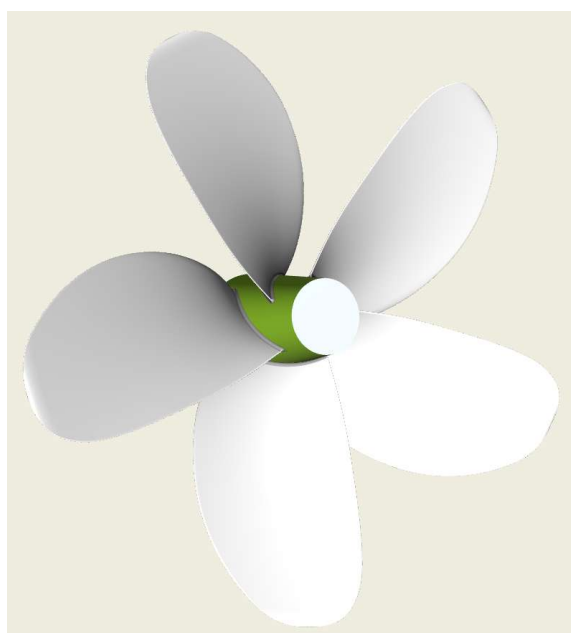
Figura 14 – Linhas contínuas representam número de configurações válidas diferentes por geração, enquanto que as linhas tracejadas representam as válidas podendo ser repetidas por geração.

Fonte: Elaborada pelo autor.

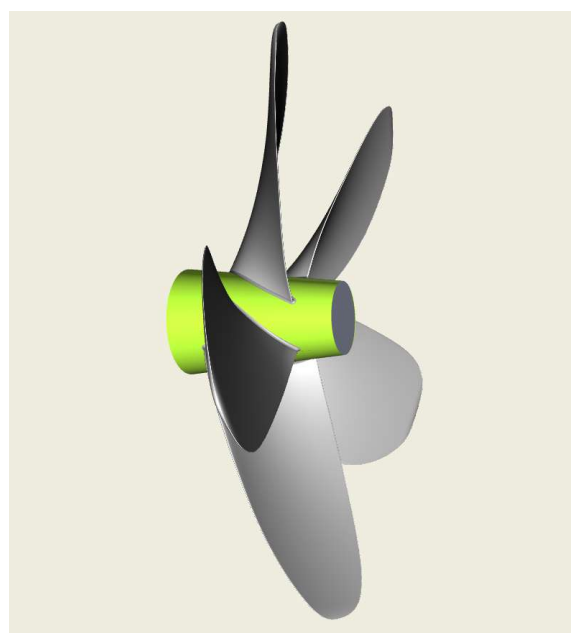
Assim se torna possível entender que os algoritmos continuam encontrando uma quantidade similar de resultados válidos, porém muito próximos. Isso significa que o algoritmo está convergindo para resultados comuns sobre a fitness, não garantidos que ótimos, mas que a área de busca está diminuindo. Logo um aumento na quantidade de gerações máximas provavelmente não teria grande impacto na melhora do resultado visto que o algoritmo está convergindo.

5.3.3 Visualização do resultado

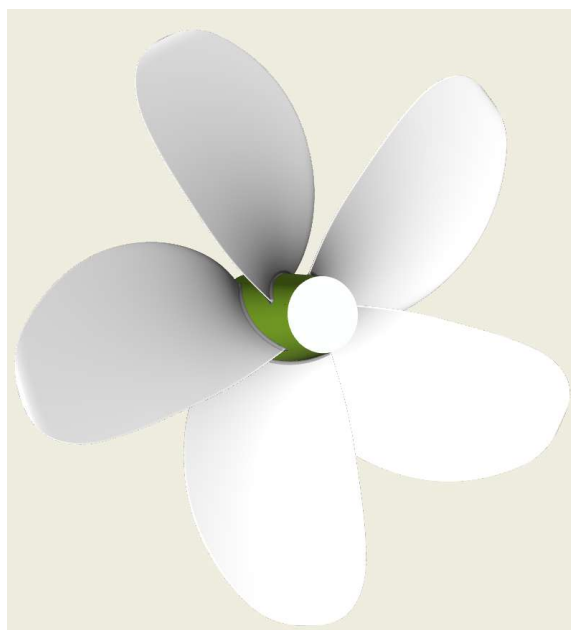
Com os melhores parâmetros encontrados pelas heurísticas foram geradas imagens da possível geometria das hélices, as Figuras 15a e 15b apresentam a geometria da hélice de frente e de lado respectivamente para os parâmetros encontrados pelo CMA-ES Modificado para v_S de 7,0 nós na Tabela 10. Enquanto que as Figuras 15c e 15d apresentam a geometria da hélice de frente e de lado respectivamente para os parâmetros encontrados pelo CMA-ES Modificado para v_S de 7,5 nós na Tabela 10.



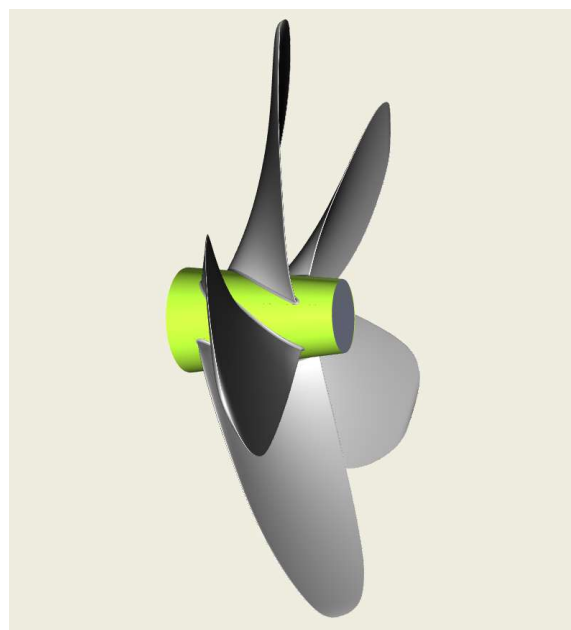
(a) Visão frontal para $v_S = 7,0$



(b) Visão lateral para $v_S = 7,0$



(c) Visão frontal para $v_S = 7,5$



(d) Visão lateral para $v_S = 7,5$

Figura 15 – Representações da Geometria 3D da Hélice.

Fonte: Elaborada pelo autor.

O software de geração de geometria 3D utilizado foi CAESES® fornecido pela Friendship Systems AG de forma gratuita, disponível em [<https://www.wageningen-b-series-propeller.com>]. Não foi possível criar imagens para todas as velocidades por limitações do software utilizado, entretanto pela proximidade dos parâmetros mesmo para diferentes valores v_S , podemos inferir que a geometria gerada seria similar as apresentadas neste trabalho.

6 CONCLUSÃO

Este trabalho se dedicou à busca de um processo capaz de gerar hélices melhoradas para drones híbridos para se locomover na água, com o foco central da pesquisa na aplicação de heurísticas nos parâmetros de descrição da morfologia da hélice, de forma a melhorar o desempenho do drone no aspecto de eficiência energética.

Foram executados experimentos de desempenho energético da hélice em água, entre algoritmos estado da arte em AEs, como CMA-ES e OpenAI-ES, também foram testadas outras abordagens mais convencionais com Algoritmos de Busca Local e Busca Aleatória. Ainda foram aplicadas avaliações de configuração de hiperparâmetros dos AEs para aprimorar os testes realizados.

Com os resultados obtidos a partir dos experimentos conduzidos, verificar-se evidências da viabilidade da utilização de algoritmos evolutivos, mais especificamente de Estratégias Evolutivas para encontrar configurações de hélices boas para desempenho na água. Esses algoritmos apresentaram desempenho melhor do que as abordagens puramente evolutivas ou de busca local, em encontrar parâmetros descritivos de morfologia de hélices que fornecessem melhor eficiência energética.

O novo cálculo de fitness permitiu as heurísticas explorarem mais soluções candidatas possíveis e apresentarem uma convergência mais rápida do que somente a utilização do P_B como avaliação. Além de possibilitar o funcionamento das heurísticas em valores de v_S mais altos, que trabalhos anteriores não tinham alcançado. Com o CMA-ES com 10 indivíduos de população, apresentando a menor variação entre execuções de diferentes *seeds*, encontrando os melhores resultados para os valores de v_S 7,0, 7,5 e 8,0, sendo eles 81,405, 106,194 e 135,597 respectivamente. Enquanto o OpenAI-ES com 10 indivíduos encontrou o melhor resultado com o valor de 169,446 para v_S de 8,5 nós.

Como todos os experimentos registraram as configurações de parâmetros e seus respectivos resultados de desempenho energético e as restrições geradas para aquele design de hélice, foi possível realizar diferentes análises para todo o processo. Dessa forma comparações entre resultados entre heurísticas diferentes foram realizadas de forma simples, apenas necessitando a implementação de ferramentas de visualização dos dados para cada análise diferente.

Por fim, para trabalhos futuros no campo de otimização de hélices para drones híbridos, se tem a possibilidade da utilização de métricas de drones reais como parte dos limites para os parâmetros da otimização. Utilizando medidas de drones reais para se obter valores como diâmetro máximo da hélice ou dados de resistência de materiais que constituem as hélices, podendo ser feito um ajuste nas restrições que impedem a viabilidade de diversas hélices em v_S mais altos. Ao incorporar essas métricas, as soluções obtidas pelos algoritmos poderiam se aproximar ainda mais das

configurações reais de hélices de drones híbridos.

Outra contribuição importante seria a implementação de testes para hélices no ar, com a expansão do escopo para incluir a avaliação de hélices não apenas em ambientes aquáticos, mas que também sejam capazes de voar. Testes estes com foco em drones híbridos com hélices que operam nos dois meios (água e ar), e que ainda são raros na literatura atual.

Por fim, realizar a validação de testes de potência teóricos em conjunto com testes práticos, para garantir que os resultados teóricos apresentem desempenho semelhantes nos drones reais. Assim conduzir testes práticos que validem as previsões teóricas, e ajustes a eventuais discrepâncias que surjam de forma iterativa, até que sejam obtidos testes teóricos que apresentem resultados similares aos encontrados na execução de hélices reais.

6.0.1 Código Fonte

O código fonte utilizado para experimentos, assim como os resultados obtidos e imagens geradas estão disponíveis publicamente no link: <https://github.com/pauloBFalmeida/hydrone>. Assim como arquivos "README.md" com explicações do funcionamento do código e como replicar os experimentos realizados neste trabalho.

REFERÊNCIAS

- ADBY, P. **Introduction to Optimization Methods**. [S.l.]: Springer Netherlands, 2013. (Chapman and Hall Mathematics Series). ISBN 9789400957053.
- AIRFOILTOOLS. Acesso em: 21 de julho de 2023, 14:00. Disponível em: <http://airfoiltools.com/airfoil/details?airfoil=n0012-il>.
- ANDERSSON, R. S.; BLOOMFIELD, P. Properties of the random search in global optimization. **Journal of Optimization Theory and Applications**, v. 16, n. 5, p. 383–398, set. 1975. ISSN 1573-2878.
- ANSYS Workbench 2020 R1. [S.l.: s.n.], 2020. Acesso em: 18 de julho de 2023, 17:30. Disponível em: <https://www.ansys.com/blog/ansys-2020-r1>.
- AYAMGA, Matthew; AKABA, Selorm; NYAABA, Albert Apotele. Multifaceted applicability of drones: A review. **Technological Forecasting and Social Change**, v. 167, p. 120677, 2021. ISSN 0040-1625.
- BARTZ-BEIELSTEIN, Thomas; BRANKE, Jürgen; MEHNEN, Jörn; MERSMANN, Olaf. Evolutionary Algorithms. **WIREs Data Mining and Knowledge Discovery**, v. 4, n. 3, p. 178–195, 2014. eprint: <https://wires.onlinelibrary.wiley.com/doi/pdf/10.1002/widm.1124>.
- BENINI, Ernesto. Multiobjective Design Optimization of B-Screw Series Propellers Using Evolutionary Algorithms. **Marine Technology and SNAME News**, v. 40, n. 04, p. 229–238, out. 2003. ISSN 0025-3316. eprint: <https://onepetro.org/MTSN/article-pdf/40/04/229/2200571/sname-mtsn-2003-40-4-229.pdf>.
- CARLTON, J.S. Chapter 12 - Resistance and Propulsion. *In*: CARLTON, J.S. (Ed.). **Marine Propellers and Propulsion (Fourth Edition)**. Fourth Edition. Oxford, UK: Butterworth-Heinemann, 2019. P. 313–365. ISBN 978-0-08-100366-4.
- CARLTON, JS. 19 - Propeller blade strength. *In*: CARLTON, JS (Ed.). **Marine Propellers and Propulsion (Second Edition)**. Second Edition. Oxford: Butterworth-Heinemann, 2007. P. 396–415. ISBN 978-0-7506-8150-6.

CAUPIN, Frédéric; HERBERT, Eric. Cavitation in water: a review. **Comptes Rendus Physique**, v. 7, n. 9, p. 1000–1017, 2006. Nucleation. ISSN 1631-0705.

CORTEZ, Paulo. Algoritmos Genéticos e Redes Neurais na Previsão de Séries Temporais, jan. 1997.

CUI, Weicheng. An Overview on Aquatic Unmanned Aerial Vehicles. **Annals of Reviews & Research**, 2019.

CURRISTON, Drew; HARTFIELD, Roy; BURKHALTER, John. Power Turbine Blade Aerodynamic Optimization Using Non-Restrictive Evolution Strategies. *In*.

DAS, Swagatam; SUGANTHAN, Ponnuthurai Nagaratnam. Differential Evolution: A Survey of the State-of-the-Art. **IEEE Transactions on Evolutionary Computation**, 2011.

DINELLI, Chris *et al.* Configurations and Applications of Multi-Agent Hybrid Drone/Unmanned Ground Vehicle for Underground Environments: A Review. **Drones**, v. 7, n. 2, 2023. ISSN 2504-446X.

DOERR, B.; NEUMANN, F. **Theory of Evolutionary Computation: Recent Developments in Discrete Optimization**. [S.l.]: Springer International Publishing, 2019. (Natural Computing Series). ISBN 9783030294144.

DOIJODE, Pranav Sumanth; HICKEL, Stefan; VAN TERWISGA, Tom; VISSER, Klaas. A machine learning approach for propeller design and optimization: Part I. **Applied Ocean Research**, v. 124, 2022. ISSN 0141-1187.

DREWS-JR, Paulo; NETO, Armando; CAMPOS, Mario. A Survey on Aerial Submersible Vehicles. *In*.

DUAN, Xu. **Multidisciplinary Design Optimization of Coaxial Drone Propellers**. 2022. F. 61. Tese (Doutorado). ISBN 978-1-60086-983-9.

EIBEN, A. E.; SMITH, J. E. What Is an Evolutionary Algorithm? *In*: INTRODUCTION to Evolutionary Computing. Berlin, Heidelberg: Springer Berlin Heidelberg, 2015. P. 25–48. ISBN 978-3-662-44874-8.

ELMENSHAWY, Adham; ALSHWAILY, Yahya. Optimization of Propeller Performance for a Quadcopter Drone by Applying Aerodynamic Propeller-Ducts. *In: [S.l.: s.n.]*, jan. 2022. P. 186–197. ISBN 978-3-030-96195-4.

FRIEDMAN, Jerome; HASTIE, Trevor; HÖFLING, Holger; TIBSHIRANI, Robert. Pathwise coordinate optimization. **The Annals of Applied Statistics**, Institute of Mathematical Statistics, v. 1, n. 2, p. 302–332, 2007.

FRIENDSHIP SYSTEMS AG. **B-Series Propeller Generator**. [S.l.: s.n.], 2023. <https://www.wageningen-b-series-propeller.com/>. Acesso em: 26 de outubro de 2023, 20:15.

GEWEKE, John. Antithetic acceleration of Monte Carlo integration in Bayesian inference. **Journal of Econometrics**, v. 38, n. 1, p. 73–89, 1988. ISSN 0304-4076.

HA, David. A Visual Guide to Evolution Strategies. **blog.otoro.net**, 2017. Acesso em: 15 de junho de 2023, 16:45.

HA, David. Evolving Stable Strategies. **blog.otoro.net**, 2017. Acesso em: 15 de junho de 2023, 15:30.

HAFEEZ, Abdul; HUSAIN, Mohammed Aslam; SINGH, S.P.; CHAUHAN, Anurag; KHAN, Mohd. Tauseef; KUMAR, Navneet; CHAUHAN, Abhishek; SONI, S.K. Implementation of drone technology for farm monitoring & pesticide spraying: A review. **Information Processing in Agriculture**, v. 10, n. 2, p. 192–203, 2023. ISSN 2214-3173.

HANSEN, Nikolaus. CMA-ES: A Function Value Free Second Order Optimization Method. *In: PGMO COPI 2014 - GASPARD MONGE PROGRAM FOR OPTIMIZATION AND OPERATIONS RESEARCH CONFERENCE ON OPTIMIZATION AND PRACTICES IN INDUSTRY*. Saclay, France: [s.n.], out. 2014.

HANSEN, Nikolaus. **The CMA Evolution Strategy: A Tutorial**. [S.l.: s.n.], 2023. arXiv: 1604.00772 [cs.LG].

HANSEN, Nikolaus; ARNOLD, Dirk V.; AUGER, Anne. Evolution Strategies. *In: Springer Handbook of Computational Intelligence*. Edição: Janusz Kacprzyk e Witold Pedrycz. Berlin, Heidelberg: Springer Berlin Heidelberg, 2015. P. 871–898. ISBN 978-3-662-43505-2.

HANSEN, Nikolaus; OSTERMEIER, Andreas. Adapting arbitrary normal mutation distributions in evolution strategies: The covariance matrix adaptation. *In*: p. 312–317.

HANSEN, Nikolaus; OSTERMEIER, Andreas; GAWELCZYK, Andreas. On the Adaptation of Arbitrary Normal Mutation Distributions in Evolution Strategies: The Generating Set Adaptation. **Proceedings of the Sixth International Conference on Genetic Algorithms**, jan. 1997.

HELMA, Stephan. Surprising Behaviour of the Wageningen B-Screw Series Polynomials. **Journal of Marine Science and Engineering**, v. 6, n. 211, p. 1–63, 2020.

HOLDEN, Kjell Olav; FAGERJORD, Odd; FROSTAD, Ragnar. Early design-stage approach to reducing hull surface forces due to propeller cavitation. **Society of Naval Architects and Marine Engineers**, v. 14, n. 00323480, 1980.

HORN, Alexandre C.; PINHEIRO, Pedro M.; SILVA, Cesar B.; NETO, Armando Alves; DREWS-JR, Paulo L. J. A Study on Configuration of Propellers for Multicopter-like Hybrid Aerial-Aquatic Vehicles. *In*: 2019 19th International Conference on Advanced Robotics (ICAR). [S.l.: s.n.], 2019. P. 173–178.

IRIZARRY, Javier; GHEISARI, Masoud; WALKER, Bruce N. Usability assessment of drone technology as safety inspection tools. **J. Inf. Technol. Constr.**, v. 17, p. 194–212, 2012.

KARNOPP, Dean C. Random search techniques for optimization problems. **Automatica**, v. 1, n. 2, p. 111–121, 1963. ISSN 0005-1098.

KASNO, Mohammad 'Afif; MOHD ARAS, Mohd Shahrieel; KEONG, Yeoh Eik; AB AZIS, Fadilah; MOHAMED KASSIM, Anuar; HARUN, Mohamad Haniff; MOHD ZAMBRI, Mohd Khairi; KHAMIS, Alias. Design Analysis of Remotely Amphibian Vehicle (RAV) –Underwater Drone. **Journal of Telecommunication, Electronic and Computer Engineering (JTEC)**, v. 9, n. 3-2, p. 35–38, out. 2017.

KLIMCZYK, Witold Artur. Aerodynamic design and optimization of propellers for multicopter. **Aircraft Engineering and Aerospace Technology**, Emerald Publishing Limited, v. 94, n. 1, p. 21–30, jan. 2022. ISSN 1748-8842.

KUIPER, G. Cavitation Research and Ship Propeller Design. **Applied Scientific Research**, v. 58, n. 1, p. 33–50, mar. 1997. ISSN 1573-1987.

LAMMEREN, Wpa van; MANEN, J.D. van; OOSTERVELD, Mwc. THE WAGENINGEN B-SCREW SERIES, 1969.

LAND, Mark William Shannon. **Evolutionary algorithms with local search for combinatorial optimization**. [S.l.]: University of California, San Diego, 1998.

LEE, Jeung-Hoon. Acoustic localization of incipient cavitation in marine propeller using greedy-type compressive sensing. **Ocean Engineering**, v. 197, p. 106894, 2020. ISSN 0029-8018.

LI, Yingying; OSHER, Stanley. Coordinate descent optimization for l1 minimization with application to compressed sensing; a greedy algorithm. **Inverse Problems and Imaging**, v. 3, n. 3, p. 487–503, 2009. ISSN 1930-8337.

MAKI, Atsuo; AKIMOTO, Youhei; NAOYA, Umeda. Application of optimal control theory based on the evolution strategy (CMA-ES) to automatic berthing (part: 2). **Journal of Marine Science and Technology**, v. 26, n. 3, p. 835–845, set. 2021. ISSN 1437-8213.

MALAPUR, Harsh; SINGH, Yashraj; SHENDKAR, Mayuresh; BARVE, Atharva; BEDEKAR, Mangesh. Diffused casing of drone propeller for reduced operational noise and optimized energy consumption. **Materials Today: Proceedings**, v. 63, p. 136–140, 2022. 4th International Conference on Advances in Mechanical Engineering and Nanotechnology. ISSN 2214-7853.

MARQUES, Crístofer H.; BELCHIOR, Carlos R. P.; CAPRACE, J.-D. Marine propeller parametric optimisation and matching to electric motor. **Journal of the Brazilian Society of Mechanical Sciences and Engineering**, v. 41, n. 3, p. 119, fev. 2019. ISSN 1806-3691.

MARQUES, Crístofer Hood; ORDONEZ, Juan C.; SOUZA, Jeferson Avila; CAPRACE, Jean-David. Genetic Optimization of Propeller-Motor Matching for All-Electric Ships. *In: 2023 IEEE Electric Ship Technologies Symposium (ESTS)*. [S.l.: s.n.], 2023. P. 129–138.

MELO, M. G. d. **Adaptação local da matriz de covariância guiada por mecanismos de exploração em estratégias de evolução**. 2019. Tese (Doutorado) – Universidade Federal de Pernambuco.

MOHD DAUD, Sharifah Mastura Syed; MOHD YUSOF, Mohd Yusmialdil Putera; HEO, Chong Chin; KHOO, Lay See; CHAINCHEL SINGH, Mansharan Kaur; MAHMOOD, Mohd Shah; NAWAWI, Hapizah. Applications of drone in disaster management: A scoping review. **Science & Justice**, v. 62, n. 1, p. 30–42, 2022. ISSN 1355-0306.

OOSTERVELD, Marinus Willem Cornelis; OOSSANEN, Peter van. Further computer-analyzed data of the Wageningen B-screw series. **International shipbuilding progress**, IOS Press, v. 22, n. 251, p. 251–262, 1975.

PACHECO, Marco Aurélio Cavalcanti *et al.* Algoritmos Genéticos: Princípios e Aplicações. **ICA: Laboratório de Inteligência Computacional Aplicada. Departamento de Engenharia Elétrica. Pontifícia Universidade Católica do Rio de Janeiro. Fonte desconhecida**, v. 28, 1999.

PATEL, Vyom; NIKAM, Keval; DIKSHIT, Shantanu; AGARWALLA, Manav; ZAGADE, Chinmay. Propeller Design and Optimization for Drones. *In*: SACHDEVA, Anish; KUMAR, Pradeep; YADAV, O. P.; TYAGI, Mohit (Ed.). **Recent Advances in Operations Management Applications**. Singapore: Springer Nature Singapore, 2022. P. 411–424.

PEREZ, Ruben; LIU, Hugh; BEHDINAN, Kamran. Evaluation of Multidisciplinary Optimization Approaches for Aircraft Conceptual Design. **Collection of Technical Papers - 10th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference**, v. 4, ago. 2004.

PORUMBEL, Daniel Cosmin; HAO, Jin-Kao; KUNTZ, Pascale. A search space “cartography” for guiding graph coloring heuristics. **Computers & Operations Research**, v. 37, n. 4, p. 769–778, 2010. ISSN 0305-0548.

PRICE, W. L. Global optimization by controlled random search. **Journal of Optimization Theory and Applications**, v. 40, n. 3, p. 333–348, jul. 1983. ISSN 1573-2878.

ROY, B. Problems and methods with multiple objective functions. **Mathematical Programming**, v. 1, n. 1, p. 239–266, dez. 1971. ISSN 1436-4646.

SALIMANS, Tim; HO, Jonathan; CHEN, Xi; SIDOR, Szymon; SUTSKEVER, Ilya. **Evolution Strategies as a Scalable Alternative to Reinforcement Learning**. [S.l.: s.n.], set. 2017. arXiv: 1703.03864. Disponível em: <http://arxiv.org/abs/1703.03864>.

SFORZA, Pasquale M. Chapter 10 - Propellers. *In*: SFORZA, Pasquale M. (Ed.). **Theory of Aerospace Propulsion (Second Edition)**. Second Edition. [S.l.]: Butterworth-Heinemann, 2017. (Aerospace Engineering). P. 487–524. ISBN 978-0-12-809326-9.

SLAVÍK, Petr. Improved performance of the greedy algorithm for partial cover. **Information Processing Letters**, v. 64, n. 5, p. 251–254, 1997. ISSN 0020-0190.

SLOWIK, Adam; KWASNICKA, Halina. Evolutionary algorithms and their applications to engineering problems. **Neural Computing and Applications**, v. 32, n. 16, p. 12363–12379, ago. 2020. ISSN 1433-3058.

STAPERSMA, D; WOULD, Hk. Matching propulsion engine with propulsor. **Journal of Marine Engineering & Technology**, Taylor & Francis, v. 4, n. 2, p. 25–32, 2005.

STORN, Rainer; PRICE, Kenneth. Differential Evolution - A Simple and Efficient Heuristic for global Optimization over Continuous Spaces. **Journal of Global Optimization**, v. 11, n. 4, p. 341–359, nov. 1997. ISSN 1573-2916.

STORN, Rainer; PRICE, Kenneth. Differential Evolution – A Simple and Efficient Heuristic for Global Optimization over Continuous Spaces. **Journal of Global Optimization**, 1997.

VOSS, Stefan; MARTELLO, Silvano; OSMAN, Ibrahim H; ROUCAIROL, Catherine. **Meta-heuristics: Advances and trends in local search paradigms for optimization**. [S.l.]: Springer Science & Business Media, 2012.

WANG, Feng-Sheng; CHEN, Li-Hsunan. Heuristic Optimization. *In*: **Encyclopedia of Systems Biology**. Edição: Werner Dubitzky, Olaf Wolkenhauer, Kwang-Hyun Cho e Hiroki Yokota. New York, NY: Springer New York, 2013. P. 885–885. ISBN 978-1-4419-9863-7.

WEBB, A.; EAMES, C.; TUFFREY, A. Factors Affecting Design Stresses in Marine Propellers. *In*.

WIERSTRA, Daan; SCHAUL, Tom; PETERS, Jan; SCHMIDHUBER, Juergen. Natural Evolution Strategies. English (US). *In*: 2008 IEEE Congress on Evolutionary Computation, CEC 2008. [S.l.: s.n.], nov. 2008. P. 3381–3387. Generated from Scopus record by KAUST IRTS on 2022-09-14.

WOUD, H. K.; STAPERSMA, D. **Design of Propulsion and Electric Power Generation Systems**. London, UK: IMarEST, 2013.

APÊNDICE A – ARTIGO DO TCC

Neste apêndice é apresentado o artigo referente a este trabalho seguindo o padrão da Sociedade Brasileira de Computação.

Otimização Paramétrica de Hélices de Drone Híbrido para Desempenho na Água por Meio de Algoritmos Evolutivos

Paulo B. F. Almeida¹, Jônata Tyska Carvalho¹

¹ Departamento de Informática e Estatística
Universidade Federal de Santa Catarina (UFSC)
Florianópolis – SC – Brasil

paulo.b.f.almeida@grad.ufsc.br, jonata.tyska@ufsc.br

Abstract. *The drone's performance relies on efficient propellers that convert energy into motion. For drones capable of moving in both air and water using the same propellers, enhancing the propeller design for greater energy efficiency involves generating and testing different morphological parameters. This study focuses on describing morphology with continuous and discrete parameters, aiming to enhance the propeller configuration to reduce energy consumption. Comparing Evolutionary Algorithms (CMA-ES, OpenAI-ES) with local and random search at different service speeds, the results demonstrate effectiveness at speeds not achieved by previous approaches and superiority in cases with pre-existing results.*

Resumo. *O desempenho do drone depende de hélices que sejam eficientes em convertem energia em movimento. Para drones capazes de se movimentar na água e ar com as mesmas hélices, o aprimoramento do design das hélices, visando maior eficiência energética, envolve a geração e teste de diferentes parâmetros morfológicos. Este estudo foca na descrição da morfologia com parâmetros contínuos e discretos, buscando melhorar a configuração da hélice para reduzir o consumo de energia. Comparando Algoritmos Evolutivos (CMA-ES, OpenAI-ES) com busca local e aleatória em diferentes velocidades de serviço. Os resultados mostram eficácia em velocidades não alcançadas por abordagens anteriores e superioridade em casos com resultados pré-existentes.*

1. Introdução

O uso crescente de drones tem transformado várias áreas, proporcionando uma gama diversificada de aplicações. Segundo [Ayamga et al. 2021], essas aeronaves não tripuladas têm revolucionado operações, como monitoramento ambiental, inspeções de segurança em áreas perigosas [Irizarry et al. 2012], mapeamento aéreo e exploração [Mohd Daud et al. 2022]. Essas inovações oferecem benefícios consideráveis, incluindo maior praticidade, redução de custos e eficiência operacional.

No entanto, drones convencionais enfrentam limitações significativas, operando exclusivamente no ar, o que inviabiliza ações na água [Drews-Jr et al. 2009]. Já agentes autônomos submarinos têm mobilidade restrita à água [Cui 2019]. A proposta de drones híbridos, capazes de operar tanto no ar quanto na água, surge como solução, apresentando simplicidade, eficiência operacional e redução de custos [Kasno et al. 2017].

Esses drones podem desempenhar tarefas diversas e complexas em ambientes variados, em numerosas possibilidades de aplicações desde pesquisas ambientais, inspeção e manutenção de infraestruturas até operações de busca e salvamento, como expresso em [Dinelli et al. 2023].

A otimização da morfologia das hélices é crucial para melhorar o desempenho e a eficiência dos drones, visto que desempenham papel fundamental, convertendo energia em movimento [Woud and Stapersma 2013]. A configuração das hélices influencia diretamente a capacidade de voo e eficiência energética [Perez et al. 2004], sendo assim a escolha dos parâmetros de morfologia da hélice é crucial para a otimização. Este trabalho adotou parâmetros da série "Wageningen B-screw" [van Lammeren et al. 1969], simplificando a descrição da morfologia da hélice, de forma análoga a [Malapur et al. 2022]. Os testes de desempenho foram realizados calculando a "Potência de Frenagem" (P_B) da hélice, medindo a energia mínima necessária para movimentar o drone na água, de forma semelhante a realizada em [Marques et al. 2019].

Encontrar configurações de hélice eficientes para drones híbridos representa um desafio significativo para modelos atuais. Devido a falta de pesquisas específicas nesta área, os modelos empregados por este trabalho são inspirados em literatura de otimização de hélices para drones convencionais, como a utilização de Algoritmos Evolutivos (AE) na escolha dos parâmetros de descrição da morfologia da hélice em [Klimczyk 2022]. AE, como Estratégias Evolutivas da OpenAI (OpenAI-ES) [Salimans et al. 2017] e Estratégias Evolutivas com Adaptação da Matriz de Covariância (CMA-ES) [Hansen 2014], foram escolhidos para a otimização, comparando-os com abordagens de busca local e aleatória. Os resultados indicam que os AEs superam abordagens convencionais, encontrando configurações de hélices mais eficientes em diferentes velocidades de serviço, inclusive superando resultados já conhecidos em alguns casos.

Dada a complexidade da otimização com parâmetros contínuos e discretos simultaneamente, este trabalho optou por realizar a otimização de parâmetros contínuos para cada valor discreto do número de pás da hélice. Utilizando dados de uma balsa projetada para operação em um lago no sudoeste do Brasil, assemelhando-se ao desafio abordado em [Marques et al. 2019]. Experimentos foram conduzidos com diferentes heurísticas em diferentes "service speeds" (v_S), visando identificar configurações de hélice que minimizassem P_B . Os AEs mostraram eficácia em alcançar resultados válidos em v_S não alcançadas por abordagens anteriores, e superando resultados conhecidos com até 0,26% de redução em P_B .

2. Fundamentação Teórica

2.1. Geração da Morfologia

A abordagem adotada para este trabalho é a série de "Wageningen B-Screw" [van Lammeren et al. 1969] para a geração da morfologia da hélice, por apresentar variados estudos na literatura e ser amplamente utilizada. É feito o uso de equações polinomiais para determinar os atributos da hélice e realizar cálculos de desempenho hidrodinâmicos. Uma grande vantagem desta abordagem é o uso de quantidade reduzida de parâmetros necessários para descrever a morfologia de uma hélice, apresentados na Tabela 1. Sendo **Diâmetro da hélice** (D), **Área expandida dividida pela área original** (A_E/A_O), **Relação passo pelo diâmetro** (P/D) e **Número de lâminas** (Z) o parâmetro

discreto referente ao número de pás em uma hélice. Esses parâmetros são melhorados pelas heurísticas de otimização, respeitando seus respectivos intervalos, de forma similar ao realizado em [Marques et al. 2019].

Tabela 1. Parâmetros para a geração da hélice.

| Parâmetro | Significado | Intervalo | Tipo |
|-----------|--|-----------|----------|
| D | Diâmetro externo da hélice (metros) | 0,5–0,8 | Contínuo |
| A_E/A_O | Razão da área da pá | 0,3–1,05 | Contínuo |
| P/D | Razão do passo da hélice pelo diâmetro | 0,5–1,4 | Contínuo |
| Z | Número de pás na hélice | 2–7 | Discreto |

2.2. Restrições da Hélice

As hélices estão sujeitas a restrições físicas que limitam a viabilidade de certas configurações de morfologia, conforme discutido em [Webb et al. 1975]. Assim, na avaliação das hélices, foram consideradas as seguintes restrições, detalhadas em [Marques et al. 2019] e [Marques et al. 2023].

Resistência da lâmina da hélice: De acordo com [Carlton 2007], caso a lâmina não suporte as forças envolvidas durante a operação, pode ocorrer deformação ou quebra, resultando em perda de eficiência ou inoperabilidade. A espessura da lâmina a 75% do raio ($t_{0.75R}$) não deve ser inferior à espessura mínima exigida ($t_{0.75R,min}$). Utilizaram-se as abordagens de $t_{0.75R}$ e $t_{0.75R,min}$ de [Oosterveld and van Oossanen 1975].

Cavitação: Este fenômeno, descrito em [Caupin and Herbert 2006], ocorre quando a diminuição de pressão forma bolhas de ar próximas à hélice, podendo causar problemas como redução do desempenho e danos estruturais. A restrição da cavitação na parte traseira da lâmina seguiu o diagrama de *cavitação de Burrill* para fluxo uniforme [Carlton 2019], garantindo que o coeficiente de carga de empuxo ($\sigma_{0.7R}$) seja inferior ou igual ao coeficiente de carga de empuxo máximo ($\sigma_{c0.7R,max}$).

Velocidade periférica da hélice: Elevadas velocidades periféricas podem resultar em ruído, vibração e redução da eficiência. A velocidade periférica na ponta da lâmina (v_{tip}) deve ser mantida baixa, não ultrapassando 39 m/s, conforme [Holden et al. 1980].

2.3. Cálculo de Propulsão da Hélice

Uma hélice projetada para operar em ambientes aquáticos deve levar em consideração a "service speed" (v_S), que representa a velocidade de serviço em nós na qual a embarcação opera, conforme definido em [Carlton 2019]. A escolha do valor de v_S ao projetar a lâmina depende da aplicação e das condições ambientais, permitindo estudar diferentes valores durante o *design*. Porém, como essa métrica é utilizada no cálculo de propulsão, obriga que comparações entre diferentes designs de hélices devam ser feitos somente entre valores iguais de v_S .

O desempenho da hélice em água é quantificado pela métrica "Brake Power" (P_B), que representa a potência do motor necessária para mover a embarcação [Stapersma and Woud 2005]. Em veículos aquáticos, a métrica de P_B é particularmente relevante, pois reflete a potência necessária para superar a velocidade da água (v_S) durante a operação. Portanto, uma hélice eficiente é aquela que demanda menos potência

do motor, indicada pelo menor valor de P_B para uma determinada v_S , como discutido em [Marques et al. 2019]. Assim os métodos de otimização devem buscar uma configuração de parâmetros de hélice que minimizem o valor de P_B para uma v_S escolhida.

No processo de otimização, é crucial considerar as restrições operacionais, conforme abordado em [Benini 2003]. A hélice deve atender a esses limites para ser viável. Assim, a otimização busca não apenas o menor valor de P_B , mas sim o menor valor que resulta em uma configuração de hélice válida, onde todas as restrições estão dentro dos limites estabelecidos.

2.4. Heurísticas e Algoritmos de Otimização

Os algoritmos exatos de otimização buscam a solução ótima para um problema, avaliando candidatos por meio de uma função objetivo, conforme descrito em [Roy 1971]. Eles visam encontrar valores de parâmetros que otimizem essa função, maximizando o desempenho do sistema, como detalhado em [Adby 2013].

Heurísticas são abordagens computacionais que aprimoram iterativamente soluções candidatas com base em uma medida de qualidade, geralmente fornecida pela função objetivo. Esses métodos exploram diversas soluções candidatas, buscando ótimos locais ou globais, com ênfase na eficiência computacional para soluções próximas às melhores possíveis. São particularmente úteis em problemas nos quais garantir a solução ótima é muito difícil, como esclarecido em [Wang and Chen 2013].

2.5. Função de Avaliação

Como explicado em [Bartz-Beielstein et al. 2014], a função de avaliação ou função fitness, é uma maneira de avaliar a qualidade ou o desempenho dos parâmetros, através da atribuição de um valor numérico, uma forma de pontuação, representando o quão adequada a solução candidata é em relação ao objetivo do problema.

2.6. Algoritmos Evolutivos

Conforme [Doerr and Neumann 2019], os algoritmos evolutivos (AEs) constituem uma classe de métodos de otimização inspirados na evolução biológica, refletindo o processo de seleção natural. Conforme [Slowik and Kwasnicka 2020], AEs são versáteis e adequados para lidar com problemas de otimização multiobjetivo, restrições complexas, alta dimensionalidade, o que os torna cada vez mais aplicáveis em áreas como a engenharia.

Essencialmente, esses métodos empregam populações de soluções candidatas, representadas como **indivíduos** ou **cromossomos**, é composto por **genes** que são os parâmetros a serem melhorados pelo algoritmo. Cada indivíduo oferece um conjunto de valores que constitui uma candidata solução para o problema em questão, que são avaliados por meio de uma **função de avaliação**, que mensura seu desempenho em relação ao objetivo definido. Com base nessa avaliação, os melhores indivíduos são selecionados para gerar uma nova população, utilizando técnicas como mutação e recombinação entre indivíduos, conforme discutido em [Bartz-Beielstein et al. 2014].

A operação de **mutação** introduz pequenas alterações aleatórias nos indivíduos, possibilitando diversificação nos atributos e a criação de características ausentes na população. Já o **cruzamento** ou **crossover**, combina características dos indivíduos existentes, gerando novos indivíduos com atributos distintos, mas mantendo características presentes na população [Doerr and Neumann 2019].

Após mutação e recombinação, ocorre a **seleção**, na qual os melhores indivíduos, conforme critérios estabelecidos, formam a próxima geração. Algoritmos variados aplicam diferentes critérios de seleção, sendo possível incorporar elitismo para privilegiar os mais adaptados, como discutido em [Duan 2022].

Ao longo das gerações, o algoritmo visa aprimorar as soluções, buscando a **convergência** para uma solução que melhore a função objetivo. Dessa forma, a população evolui sucessivamente, buscando soluções que resultem nos parâmetros melhorados para o problema, conforme explicado em [Eiben and Smith 2015]. A Figura 1 oferece uma visão geral do funcionamento de um Algoritmo Evolutivo.

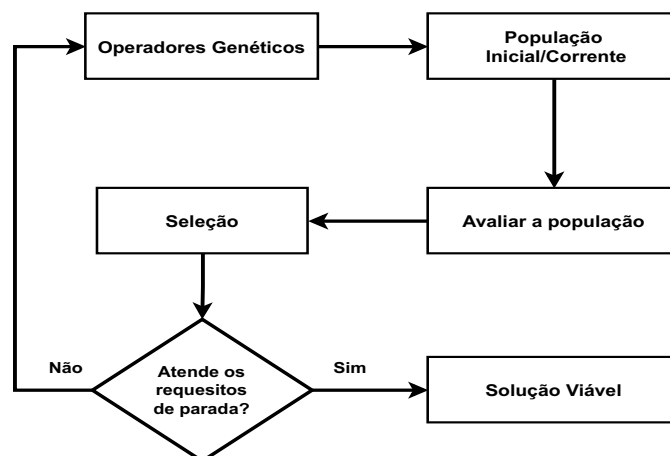


Figura 1. Visão geral do funcionamento de um Algoritmo Evolutivo. Adaptado de [Pacheco et al. 1999].

2.7. Differential Evolution (DE)

O *Differential Evolution* (DE) é um AE eficaz para otimização em espaços de busca complexos, propondo uma abordagem independente do gradiente, adequada para problemas ruidosos ou não contínuos [Storn and Price 1997b]. Operando com uma população de soluções candidatas, o algoritmo gera novas soluções através de mutação e recombinação, explorando eficientemente o espaço de busca. Ao longo das gerações, a solução com melhor desempenho é preservada [Das and Suganthan 2011].

O processo iterativo, de criação de novos indivíduos gerados pelos operadores genéticos, utilizando o melhor encontrado até então, continua até atingir o critério de parada, como o número de gerações ou estabilidade da fitness. Dessa forma, o DE busca progressivamente uma solução aprimorada, como descrito em [Storn and Price 1997a].

2.8. Estratégias Evolutivas (EE)

Os algoritmos de Estratégias Evolutivas (Estratégias Evolutivas (EE)) pertencem a classe de AE, e realizam o processo de busca e aprimoramento gradual de soluções, de maneira a imitar o que acontece na natureza com a seleção natural [Hansen et al. 2015].

Uma vantagem das EE é a capacidade de avaliar simultaneamente a população de soluções candidatas, acelerando o processo de busca, uma vez que a avaliação de cada indivíduo é independente dos outros [Salimans et al. 2017]. Diferentes algoritmos de EE empregam estratégias diversas de busca e atualização de soluções durante a evolução.

2.9. CMA-ES

O CMA-ES [Hansen 2014] é uma EE que utiliza a Adaptação da Matriz de Covariância (CMA) [Hansen and Ostermeier 1996], ajustando dinamicamente o espaço de busca com base nos resultados de cada geração. A matriz de covariância reflete as relações entre os parâmetros, sendo recalculada a cada geração para orientar a busca. Para isso, o algoritmo utiliza uma distribuição normal multivariada para gerar amostras de soluções. Os melhores indivíduos influenciam na atualização da matriz de covariância, permitindo a expansão ou redução iterativa do espaço de busca [Hansen 2023, Melo 2019].

2.10. OpenAI-ES

O OpenAI-ES [Salimans et al. 2017] é uma EE que utiliza uma abordagem de Estratégias de Evolução Natural [Wierstra et al. 2008], empregando um estimador da função de pontuação e ascensão de gradiente estocástico para maximizar a média da função de avaliação. Em cada iteração, o algoritmo gera uma população com perturbações nos parâmetros por meio da adição de ruído gaussiano. A "amostragem antitética" é aplicada para reduzir a variância, gerando pares de perturbações opostas. O "ajuste de avaliação" emprega uma transformação de classificação e "decaimento de peso" para mitigar influências atípicas e evitar crescimento excessivo dos parâmetros em relação às mutações, contribuindo para uma convergência mais eficiente [Geweke 1988, Ha 2017b].

2.11. Busca Local

Os métodos de busca local são heurísticas que selecionam a melhor solução em cada iteração, percorrendo o espaço de busca a partir de um ponto inicial. Diferentemente dos algoritmos de busca global, esses métodos concentram-se em encontrar uma solução localmente ótima dentro de uma vizinhança da solução atual [Voß et al. 2012]. No entanto, essa abordagem não garante a descoberta da melhor solução global, podendo ficar presa em mínimos locais que podem não representar a solução ótima global [Land 1998]. A Figura 2 ilustra como os mínimos locais, embora sejam os menores valores em sua vizinhança, podem não ser os menores da função global, destacando a possibilidade de encontrar soluções sub-ótimas ao utilizar busca local.

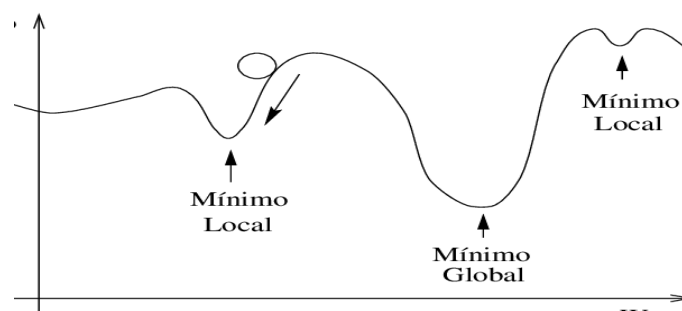


Figura 2. Diferença entre mínimo local e global. Retirado de [Cortez 1997].

2.12. Coordinate Descent

Para de problemas de minimização com múltiplas variáveis, o algoritmo *Coordinate Descent* [Friedman et al. 2007] simplifica o processo, ajustando uma coordenada por vez, mantendo as outras fixas. O algoritmo inicia com valores aleatórios dentro dos limites

de cada coordenada. Em cada iteração, uma coordenada é escolhida para ser melhorada, percorrendo valores dentro dos limites para encontrar o que gera o melhor resultado, com explicado em [Slavík 1997]. A busca linear é empregada, percorrendo um intervalo definido em passos, avaliando cada ponto pela função de avaliação. Neste trabalho o processo é repetido até atingir o critério de parada de número máximo de iterações.

2.13. Busca Aleatória

O algoritmo de busca aleatória é uma heurística que explora o espaço de busca fazendo escolhas aleatórias. Diferentemente de abordagens sistemáticas, esse método gera amostras aleatórias no espaço de busca e avalia seu desempenho. Essa abordagem é simples de implementar e pode ser eficaz em espaços de busca rugosos, evitando ficar preso em vales locais, como explicado em [Anderssen and Bloomfield 1975, Karnopp 1963]. Embora seja altamente paralelizável, exige testar muitas amostras para encontrar uma solução adequada, sem garantia de encontrar uma solução ótima global ou local, como descrito em [Price 1983].

3. Varredura Sistemática do Espaço de Busca

Sem uma extensa pesquisa na literatura sobre o este tópico, foi realizada uma varredura sistemática no espaço de busca dos parâmetros, permitindo explorar o comportamento e a influência dos parâmetros nos resultados dos experimentos. Apesar das limitações inerentes à varredura sistemática, essa abordagem foi adotada devido ao espaço de busca limitado e ao número reduzido de parâmetros [Porumbel et al. 2010]. A varredura foi realizada de forma discretizada e segmentada para uma avaliação preliminar das funções e dos resultados das heurísticas estudadas. A escolha do tamanho dos segmentos foi feita visando a viabilidade de execução dentro do escopo do trabalho em termos de tempo.

4. Trabalhos Relacionados

A otimização paramétrica de hélices para drones híbridos, capazes de operar tanto no ar quanto na água, é uma área com limitadas contribuições na literatura. Em contraste, há diversos estudos focados na otimização de hélices para drones aéreos convencionais, onde AEs são comumente aplicados para encontrar configurações eficientes, como evidenciado em [Klimczyk 2022].

O uso de EE em hélices de drones é abordado em [Curriston et al. 2013], que emprega cálculos de sustentação, resistência gerada pelas hélices e potência durante a rotação das pás como parâmetros de otimização. Sendo a sustentação máxima gerada pela hélice é o principal objetivo de otimização. Outro estudo [Duan 2022] utiliza Algoritmos Genéticos para a melhorar hélices de drones aéreos, sendo os parâmetros utilizados inclinação, afinamento, tamanho da corda raiz e semi-envergadura. A eficiência global do propulsor é avaliada usando o simulador *ANSYS 2020*, considerando aspectos como tração, potência consumida e eficiência aerodinâmica.

A série "*Wageningen B-Screw*" [van Lammeren et al. 1969] é amplamente adotada na descrição de hélices para embarcações aquáticas, como destacado em [Helma 2020]. Em [Marques et al. 2019], o algoritmo DE é empregado para otimizar hélices para barcos, geradas a partir dessa série. Utilizando P_B como métrica de desempenho, visando a eficiência da hélice.

Na Tabela 2, observa-se que os estudos revisados empregam principalmente abordagens baseadas em AEs para otimizar hélices de veículos não tripulados. No entanto, as metas de otimização variam, com alguns focando na eficiência energética [Klimczyk 2022] e [Marques et al. 2019], enquanto outros priorizam o empuxo máximo [Duan 2022] e [Doijode et al. 2022]. Algumas pesquisas consideram uma ampla gama de parâmetros, como coeficiente de sustentação e nível de ruído [Curriston et al. 2013]. Notavelmente, alguns estudos impõem restrições à velocidade de operação, enquanto este trabalho busca evitar tais limitações para ampliar a aplicabilidade da abordagem proposta.

Tabela 2. Conteúdo dos Trabalhos Relacionados.

| Trabalho | Utiliza Algoritmos Evolutivos | Otimização Eficiência Energética | Hélice para Meio Aquático | Sem Restrição na Velocidade de Operação |
|-------------------------|--------------------------------------|---|----------------------------------|--|
| [Klimczyk 2022] | X | X | | X |
| [Curriston et al. 2013] | X | | | X |
| [Duan 2022] | X | | | |
| [Doijode et al. 2022] | | | X | X |
| [Marques et al. 2019] | X | X | X | |
| Este Trabalho | X | X | X | X |

5. Metodologia

Inicialmente, realizaram-se experimentos exclusivamente na otimização de hélices para operação na água. Essa abordagem foi escolhida devido à escassez de estudos disponíveis na literatura sobre agentes autônomos submarinos. Optou-se por iniciar com uma análise e avaliação das hélices em ambiente aquático, confirmando sua eficácia antes de prosseguir para experimentos mais complexos.

Essa abordagem possibilita um melhor entendimento do funcionamento dos métodos de otimização específicos para cada meio, permitindo ajustes e melhorias ao longo do processo, garantindo que a avaliação de desempenho funcione corretamente. A divisão em etapas distintas permite ajustes refinados durante o processo de melhora da hélice híbrida, possibilitando um ajuste mais preciso do funcionamento do drone híbrido, caso seja necessário um maior desempenho em apenas um dos meios.

Este trabalho propõe uma análise comparativa entre métodos de otimização encontrados na literatura no contexto de hélices aquáticas, incluindo DE, CMA-ES, OpenAI-ES, *Coordinate Descent* e Busca Aleatória. Com o objetivo de comparar o desempenho desses métodos e a qualidade das soluções obtidas.

5.1. Protocolo Experimental para Água

Para calcular a propulsão da hélice na água, utilizaram-se dados de uma balsa projetada para operação em um lago no sudoeste do Brasil, apresentados em [Marques et al. 2019]. Todos os experimentos usaram os valores na Tabela 3 como parâmetros constantes de ambiente, embarcação e sistema de propulsão.

O v_S foi o único parâmetro alterado em diferentes execuções. Valores mais altos de v_S exigem maior desempenho da hélice, tornando mais complexo encontrar

configurações que respeitem as restrições. Comparações entre configurações de hélices e heurísticas foram feitas somente para valores idênticos de v_S .

Os algoritmos deste estudo otimizaram exclusivamente os parâmetros listados na Tabela 1. Inicialmente, os valores de velocidade da água (v_S) utilizados foram 7,0 e 7,5 nós. Posteriormente, também foram testados valores de v_S de 8,0 e 8,5 nós para avaliar o desempenho das heurísticas em cenários desafiadores. Visto que não foram possíveis encontrar configurações de hélices válidas para essas velocidades em trabalhos anteriores como [Marques et al. 2019].

Tabela 3. Parâmetros de entrada constantes. Adaptado de [Marques et al. 2019]. Traduzido pelo autor.

| Parâmetro | Valor |
|--------------------------------|--|
| <i>Ambiente</i> | |
| Aceleração devido à gravidade | 9.80665 m/s ² |
| Pressão atmosférica | 101.325 kPa |
| Densidade da água | 999.0 kg/m ³ |
| Viscosidade cinemática da água | 1.139×10^{-6} m ² /s |
| Pressão de vapor da água | 1704 Pa |
| <i>Embarcação</i> | |
| Comprimento de linha d'água | 22.00 m |
| Largura | 9.600 m |
| Calado da embarcação | 1.089 m |
| Deslocamento volumétrico | 204.1 m ³ |
| Coefficiente de meia-nau | 0.9761 |
| Coefficiente do plano d'água | 0.9964 |
| Parâmetro de forma da popa | 10 |
| Número de hélices | 2 |
| <i>Propulsão</i> | |
| Centro da hélice | 0.500 m |
| Tensão permitida da hélice | 134.8 MPa |
| Área de cavitação permitida | 5% |
| Eficiência do eixo | 99.00% |

5.2. Varredura do Espaço de Busca

Para estudar o funcionamento dos algoritmos e verificar a existência de valores viáveis, foi realizada uma exploração no espaço de busca dos parâmetros. Cada parâmetro contínuo foi dividido em 50 segmentos, e todos os valores possíveis para Z foram considerados. Uma varredura foi feita para três valores de v_S : 7,0, 7,5 e 8,0 nós. Como para cada possível combinação de parâmetros, resulta em quantidade de avaliações exponencial, determinado pelo número de segmentos de parâmetros contínuos utilizados. Foi realizado um total de 750.000 avaliações para cada valor de v_S . A multiplicação por 3 na Equação 1 se refere à quantidade de valores de v_S utilizados, enquanto o valor 6 representa a quantidade de valores discretos de Z , resultando em 2.250.000 avaliações no total.

$$\text{Avaliações Totais} = 3 \times (6 \times 50^3) \quad (1)$$

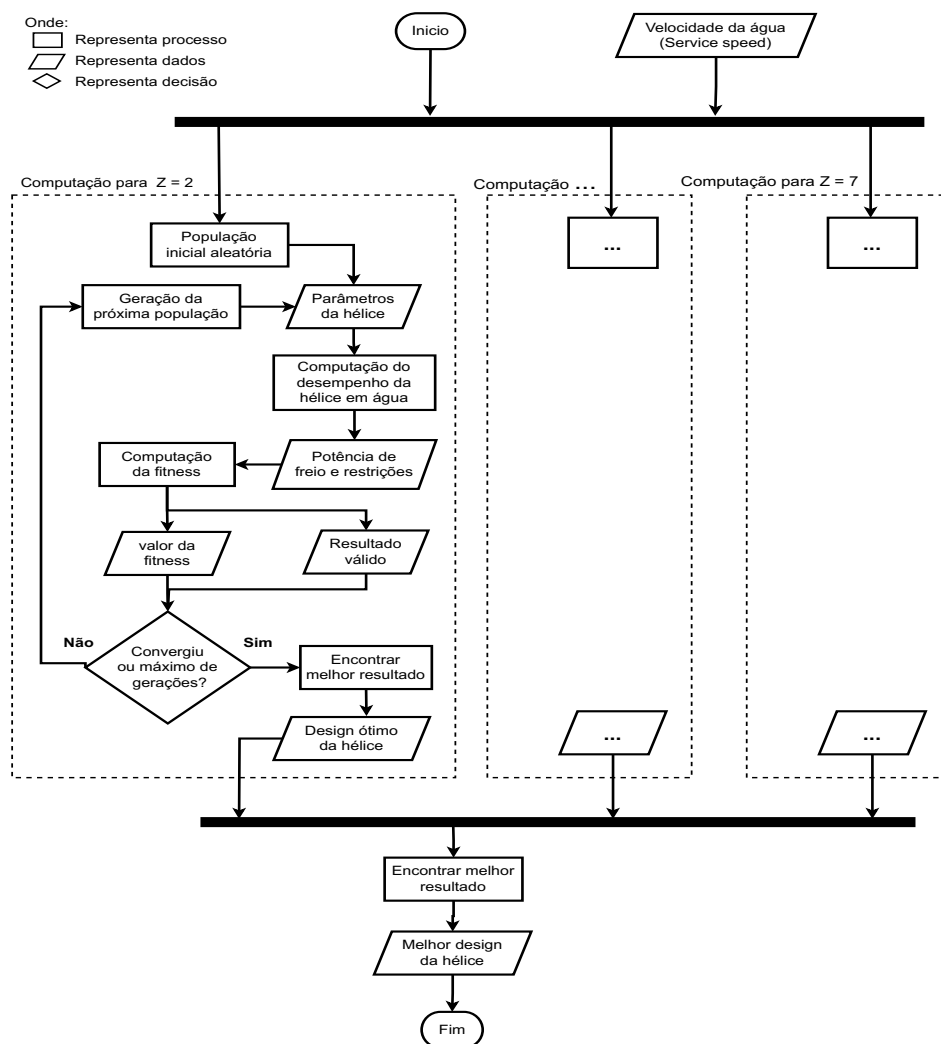
Cada avaliação configurou uma hélice e armazenou o valor de P_B correspondente. Os dados coletados foram usados para gerar imagens, analisar o comportamento dos parâmetros em relação a P_B e validar as configurações obtidas em comparação com os resultados das heurísticas.

5.3. Experimentos Iniciais com Base na Literatura

Inicialmente, foi replicada a execução do algoritmo DE de [Marques et al. 2019] para validar os algoritmos. A implementação considerou apenas P_B como função objetivo, executando DE com os parâmetros da Tabela 4. Cada execução testou 900 configurações de hélices, calculo realizado através da Equação 2, estabelecendo um orçamento computacional para comparação com outros algoritmos.

Para CMA-ES, OpenAI-ES, *Coordinate Descent* e Busca Aleatória, cada execução abrangeu 6 cenários distintos, variando Z conforme a Tabela 1. Isso reduziu os parâmetros e permitiu a execução simultânea de 6 cenários. A melhor solução entre os cenários foi considerada como resultado encontrado pela heurística para aquela execução, conforme o diagrama de fluxo na Figura 3.

Figura 3. Diagrama de Fluxo de Execução das Heurísticas.



$$\text{Execuções} = \text{Tamanho da População} \times \text{Número de Gerações} \quad (2)$$

Foram realizadas 10 execuções independentes para cada heurística, usando sementes distintas para a geração de números aleatórios para variabilidade. Essa abordagem foi aplicada para os valores distintos de v_S : 7,0 e 7,5.

5.4. Função Objetivo

A função objetivo busca minimizar o valor de P_B para encontrar configurações de hélice com melhor desempenho, mas considerando restrições. Em contraste com o DE de [Marques et al. 2019], que não considera hélices inviáveis. Assim a abordagem proposta ajusta a função objetivo para incluir penalidades para hélices fora das restrições, permitindo a participação delas durante o processo. Mantendo o objetivo principal encontrar o menor valor de P_B , mas levando em consideração o quanto a hélice ultrapassa as restrições como penalidade para não serem candidatas tão desejáveis quanto as hélices dentro das restrições.

As restrições consideradas neste trabalho seguem as seguintes limitações:

- Para evitar a **Cavitação** é necessário que o coeficiente de carga de empuxo ($\sigma_{0.7R}$) não ultrapasse o coeficiente de carga de empuxo máximo ($\sigma_{c0.7R,max}$).
- Para evitar problemas a **Velocidade periférica** na ponta da lâmina (v_{tip}) não pode ultrapassar a velocidade máxima de 39 m/s.
- Para manter a **Resistência da lâmina da hélice**, é necessário que a espessura da lâmina a 75% do raio da hélice ($t_{0.75R}$) não seja inferior à espessura mínima exigida ($t_{0.75R,min}$).

Para que uma hélice seja considerada válida deve respeitar as Equações 3, 4 e 5 referentes a cavitação, velocidade periférica e resistência da lâmina respectivamente.

$$\sigma_{0.7R} \leq \sigma_{0.7R,max} \quad (3)$$

$$v_{tip} \leq 39 \quad (4)$$

$$t_{0.75R} \geq t_{0.75R,min} \quad (5)$$

A Equação 6 calcula a fitness usando a razão da diferença entre o valor da restrição e seu limite, dividida pelo limite da restrição. Se a restrição estiver dentro do limite, o valor da razão é negativo, sendo ignorado pelo cálculo do máximo entre esse valor e 0.

A soma ponderada dessas razões, somada a 1, nos casos em que a configuração da hélice excede os limites das restrições, essa soma ultrapassa 1, enquanto as hélices que permanecem dentro dos limites têm uma soma igual a 1. O resultado dessa soma é então multiplicado pelo valor de P_B . Para hélices válidas, o valor da fitness é igual ao valor do P_B da hélice, enquanto configurações que ultrapassam os limites das restrições recebem valores de fitness menos desejáveis.

$$\text{fitness} = -P_B \times \left(1 + \max \left(\frac{\sigma_{0.7R} - \sigma_{c0.7R,max}}{\sigma_{c0.7R,max}}, 0 \right) + \max \left(\frac{v_{tip} - 39}{39}, 0 \right) + \max \left(\frac{t_{0.75R,min} - t_{0.75R}}{t_{0.75R,min}}, 0 \right) \right) \quad (6)$$

A fitness é negativa, pois os algoritmos buscam maximizá-la, enquanto que valores menores de P_B indicam desempenho superior da hélice.

Esta abordagem proposta para o cálculo da fitness, visa permitir que as heurísticas explorem mais amplamente o espaço dos parâmetros, possibilitando que ao longo das gerações, as soluções se aproximem de configurações de hélice válidas e melhores.

5.5. Implementação dos Algoritmos

Essa parte do trabalho é dedicada à explicação da implementação dos algoritmos. Foram realizadas 10 execuções para cada algoritmo, guardando o melhor resultado encontrado em cada execução, a fim de realizar uma análise comparativa posteriormente.

5.5.1. Implementação do DE Base

O algoritmo DE Base foi replicado com as mesmas configurações de [Marques et al. 2019], utilizando os parâmetros contidos na tabela 4. Essa implementação utilizou como função objetivo apenas o valor de P_B , não considerando hélices que estejam fora dos limites das restrições.

Tabela 4. Parâmetros de entrada no DE

| Parâmetro | Valor |
|----------------------|-------|
| Tamanho da População | 30 |
| Gerações | 30 |
| Fator de Crossover | 0.5 |
| Fator de Mutação | 0.8 |

5.5.2. DE Modificado

O algoritmo DE Modificado utiliza os mesmos valores parâmetros contidos na Tabela 4, entretanto a função objetivo utiliza o cálculo da fitness da Equação 6. Desta forma utilizando hélices que estejam fora dos limites das restrições no processo de avaliação.

5.5.3. Implementação do OpenAI-ES e CMA-ES

A implementação do CMA-ES foi baseada em [Hansen 2023] com a utilização da biblioteca de Python "pymcma". Gerando a população inicial com valores aleatórios dentro da faixa de cada parâmetro presente na Tabela 1.

A implementação do OpenAI-ES foi baseada em [Ha 2017a], que utiliza uma classe que implementa os métodos de geração da população. O algoritmo é iniciado com os valores presentes na tabela 5.

Em ambos os algoritmos a geração da população inicial foi feita com valores aleatórios dentro da faixa de cada parâmetro contínuo presente na Tabela 1. Foram realizadas 6 cenários, um para cada valor do intervalo de Z , assim realizando 150 avaliações para cada número de pás de hélices, totalizando 900 avaliações para cada execução. Ao final, de cada execução a melhor solução válida, i. e. dentro dos limites das restrições, encontrada entre os cenários, foi selecionada como o resultado da execução.

Tabela 5. Parâmetros de descrição dos algoritmos OpenAI-ES e CMA-ES.

| Parâmetro | Significado | OpenAI-ES | CMA-ES |
|---------------------|-----------------------------------|-----------|--------|
| n params | Número de parâmetros | 3 | 3 |
| popsiz | Tamanho da população | 5 | 5 |
| max iteration | Gerações | 30 | 30 |
| weight decay | Decaimento dos pesos | 0.01 | 0.01 |
| sigma init | Desvio padrão inicial | 0.1 | 0.1 |
| sigma decay | Decaimento do desvio padrão | 0.99 | |
| antithetic sampling | Amostragem antitética | False | |
| forget best result | Esquecer melhor resultado | False | |
| learning rate | Taxa de aprendizado | 0.1 | |
| learning rate decay | Decaimento da taxa de aprendizado | 0.99 | |

5.5.4. Implementação do *Coordinate Descent*

A implementação do *Coordinate Descent* segue a abordagem de [Li and Osher 2009]. Cada valor do intervalo de Z é tratado como um cenário único. Em cada cenário o algoritmo inicia com valores aleatórios dentro dos limites dos parâmetros contínuos presentes na Tabela 1.

A cada iteração, para melhorar cada parâmetro, o algoritmo percorre linearmente o espaço de busca do parâmetro selecionado, mantendo os demais fixos. O "step size" da Tabela 6 determina tamanho do passo. Ao final o valor que proporciona o melhor resultado válido é escolhido. A cada nova iteração, o "step size" é reduzido conforme o "step decay" para convergir o espaço de busca. O algoritmo avalia "params" parâmetros contínuos. Esse processo é repetido até atingir o número máximo de iterações "max iterations", calculado com base no número de parâmetros e orçamento computacional disponível.

Tabela 6. Parâmetros de descrição *Coordinate Descent*.

| Parâmetro | Significado | Valor |
|-------------------|--|--------------|
| params | Número de parâmetros a serem melhorados | 3 |
| step size inicial | Tamanho inicial do passo ao percorrer um parâmetro | 0.1 |
| step decay | Decaimento do passo a cada iteração | 0.1 |
| max iterations | Máximo de iterações | 30 |

5.5.5. Implementação da busca aleatória

O algoritmo de busca aleatória executa a função avaliação para cada ponto selecionado aleatoriamente dentro dos limites de cada parâmetro [Karnopp 1963]. Sendo realizadas 150 avaliações para cada cenário, correspondente a um valor do intervalo de Z , totalizando 900 avaliações. Ao final, a melhor solução válida encontrada, foi selecionada como o resultado final.

6. Experimentos exclusivos com Estratégias Evolutivas

Este trabalho tem como proposta a utilização de algoritmos de Estratégias Evolutivas (EE) na otimização dos parâmetros de hélices. Após a execução dos algoritmos no cenário inicial, foi realizado um estudo mais aprofundado, direcionado especificamente para os algoritmos de EE.

6.1. Estudo Sobre Tamanho da População e Número de Gerações

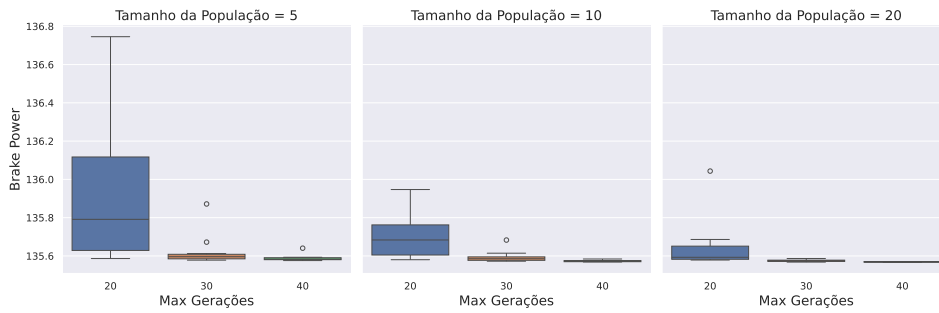
Dado o impacto significativo do tamanho da população e do número máximo de gerações na execução de algoritmos de Estratégias Evolutivas, é crucial encontrar um equilíbrio entre esses parâmetros. O aumento desses valores pode melhorar o desempenho, mas também resulta em um considerável aumento no tempo de execução, conforme ilustrado na Equação 2.

Para determinar valores ideais de tamanho da população e número de gerações, específicos para o problema em questão, foi realizado um estudo breve para os algoritmos CMA-ES e OpenAI-ES. No caso do algoritmo DE, esses aspectos já foram discutidos em estudos anteriores para o mesmo problema em [Marques et al. 2019]. Foram conduzidas 10 execuções para cada conjunto de parâmetros, considerando v_S de 8,0 nós. A Tabela 7 resume a quantidade de testes executados para cada configuração de parâmetros.

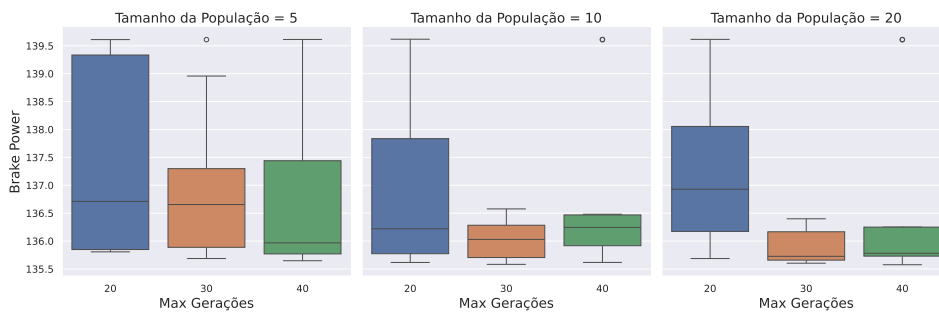
Tabela 7. Quantidade de avaliações por Tamanho da População e Máximo de Gerações.

| Máximo de Gerações | Tamanho da População | | |
|--------------------|----------------------|-----|-----|
| | 5 | 10 | 20 |
| 20 | 100 | 200 | 400 |
| 30 | 150 | 300 | 600 |
| 40 | 200 | 400 | 800 |

As Figuras 4a e 4b exibem os resultados dos algoritmos CMA-ES e OpenAI-ES, respectivamente. Tamanhos maiores na população geram uma maior uniformidade



(a) CMA-ES



(b) OpenAI-ES

Figura 4. Estudo do Tamanho da População e Máximo de Gerações

nos resultados, enquanto um aumento no máximo de gerações melhora a média dos resultados obtidos. Considerando a Tabela 7 que indica a quantidade de testes para cada configuração de máximo de gerações e tamanho da população, foi escolhido prosseguir com uma população de tamanho 10 e um máximo de 30 gerações. Essa escolha se deve à menor variação entre os resultados em comparação com a população de tamanho 5. Nota-se na Tabela 7 que aumentar o tamanho da população para 20 resultaria em dobro de testes, mas a Figura 4 indica que isso não afetaria significativamente a qualidade dos resultados. Portanto, a escolha de manter a população de 10 indivíduos.

Optou-se por manter o número de gerações em 30 devido ao aumento no tempo de execução associado a cada geração. Cada geração depende dos resultados de todos os indivíduos da geração anterior, e um aumento na quantidade de gerações não apresentou um impacto significativo na qualidade dos resultados. Dessa forma, foi mantido o máximo de gerações em 30 para os testes seguintes.

6.2. Descrição dos Experimentos

Foram conduzidos experimentos adicionais com os algoritmos de DE Modificado, CMA-ES Modificado e OpenAI-ES Modificado. Cada algoritmo teve 10 execuções independentes, com uma semente única para a geração de números aleatórios em cada execução. Essa abordagem foi aplicada para os valores de v_S : 7,0, 7,5, 8,0 e 8,5.

A implementação dos algoritmos de OpenAI-ES e CMA-ES permaneceu a mesma descrita anteriormente na Tabela 5, com apenas modificações nos parâmetros de entrada para o tamanho de população, "popsiz", de 10 indivíduos.

7. Análise de Resultados

Neste parte será discutido os resultados obtidos dos experimentos realizados.

7.1. Resultado da Varredura do Espaço de Busca

Após a varredura do espaço de busca dos parâmetros, gráficos foram gerados para analisar as soluções viáveis organizadas por valores de v_S . Surpreendentemente, soluções viáveis para v_S de 8,0 nós, não identificadas em estudos anteriores, foram encontradas. A análise revela uma diminuição na quantidade de resultados viáveis à medida que v_S aumenta.

As Figuras 5a, 5b, e 5c apresentam os gráficos para v_S de 7,0, 7,5 e 8,0 nós, respectivamente. Cada eixo representa um parâmetro contínuo da Tabela 1. As cores indicam os valores de P_B , sendo cores mais claras mais desejáveis. Destaca-se que os melhores P_B estão próximos do limite superior do diâmetro da hélice D . A Figura 5c revela uma concentração limitada de parâmetros que levam a resultados válidos, especialmente em uma pequena região específica do espaço de parâmetros.

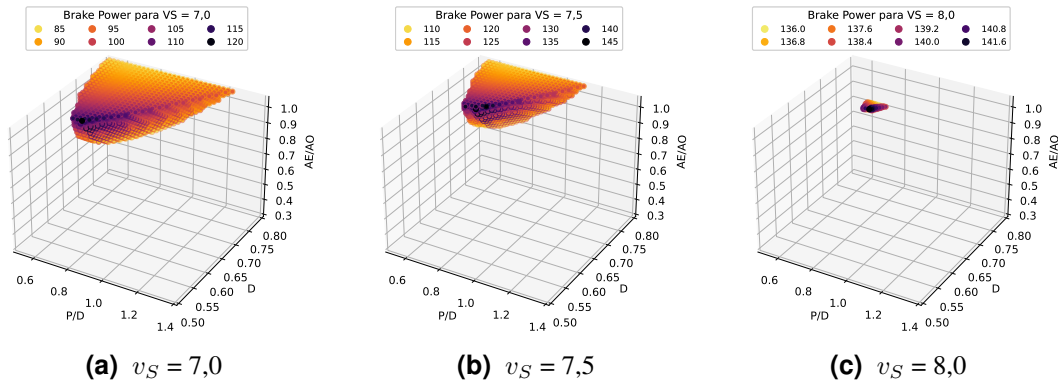


Figura 5. P_B encontrados para os conjuntos de parâmetros testados.

7.2. Resultados das Heurísticas

Após a execução das heurísticas (DE Base, DE Modificado, CMA-ES, OpenAI-ES, *Coordinate Descent* e Busca Aleatória), as Figuras 6a e 6b representam os resultados para os valores de v_S de 7,0 e 7,5 nós, respectivamente.

Ao analisar a variabilidade dos resultados ao longo de dez execuções, destaca-se a semelhança nos resultados dos AE, com o CMA-ES consistentemente apresentando desempenho mais coeso. Ambas implementações do DE forneceram resultados semelhantes, sendo a versão modificada superior, com menor variabilidade. Os algoritmos de *Coordinate Descent* e Busca Aleatória mostraram o pior desempenho em ambas as velocidades, apresentando a maior variabilidade entre os algoritmos.

Os melhores valores de P_B em kilowatt (kW), dentre as dez execuções de cada algoritmo para v_S de 7,0 e 7,5, estão na Tabela 8, juntamente com os parâmetros da série de "Wageningen B-Screw" obtidos pelas heurísticas.

Para v_S de 7,0 o algoritmo DE com o cálculo de fitness modificado encontrou o melhor resultado e todas as heurísticas alcançaram o mesmo valor de D . Enquanto para v_S de 7,5 o CMA-ES obteve o melhor resultado. Vale ressaltar que para todas velocidades os valores de P/D e A_E/A_O apresentaram pouca variação, e valores de Z próximos.

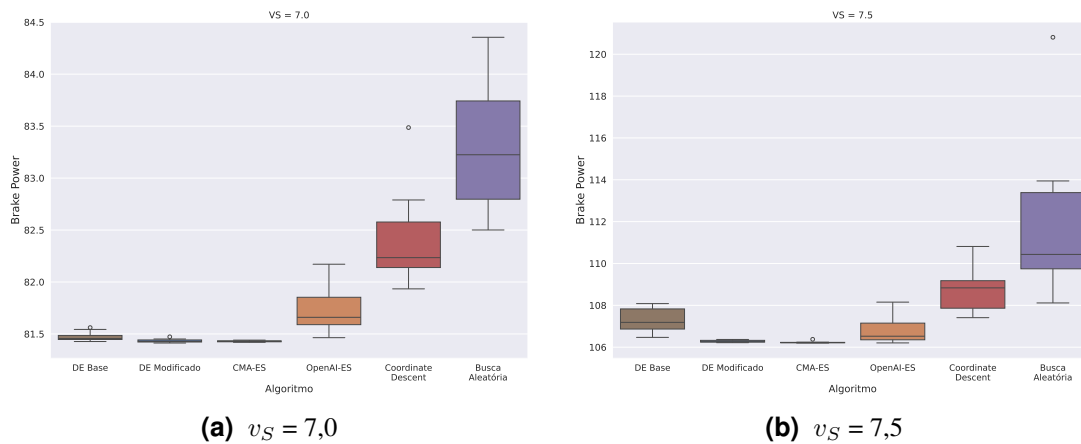


Figura 6. Valores encontrados pelas execuções.

Tabela 8. Parâmetros dos melhores resultados encontrados pelos algoritmos para valores de v_S de 7,0 e 7,5.

| v_S (nós) | Algoritmo | P_B | D | A_E/A_O | P/D | Z |
|-------------|--------------------|----------------|-------|-----------|-------|-----|
| 7,0 | DE Base | 81,427 | 0,8 | 0,641 | 0,674 | 5 |
| | DE Modificado | 81,413 | 0,8 | 0,632 | 0,669 | 5 |
| | CMA-ES | 81,419 | 0,8 | 0,631 | 0,667 | 5 |
| | OpenAI-ES | 81,464 | 0,8 | 0,640 | 0,658 | 5 |
| | Coordinate Descent | 81,933 | 0,8 | 0,680 | 0,692 | 6 |
| | Busca Aleatória | 82,501 | 0,8 | 0,737 | 0,749 | 6 |
| 7,5 | DE Base | 106,468 | 0,8 | 0,718 | 0,641 | 5 |
| | DE Modificado | 106,215 | 0,8 | 0,699 | 0,653 | 5 |
| | CMA-ES | 106,198 | 0,8 | 0,701 | 0,657 | 5 |
| | OpenAI-ES | 106,204 | 0,8 | 0,704 | 0,661 | 5 |
| | Coordinate Descent | 107,410 | 0,796 | 0,803 | 0,671 | 6 |
| | Busca Aleatória | 108,114 | 0,798 | 0,742 | 0,687 | 6 |

7.3. Resultados dos Experimentos Exclusivos de Estratégias Evolutivas

A Figura 7 exibe resultados para v_S de 7,0, 7,5, 8,0 e 8,5 nós após a execução dos heurísticas (DE Base, DE Modificado, CMA-ES, CMA-ES Modificado, OpenAI-ES e OpenAI-ES Modificado). Ao examinar os resultados em dez execuções, percebe-se que os algoritmos modificados melhoraram a variabilidade e resultados, com destaque para OpenAI-ES Modificado, consistentemente mais uniforme. Exceto pelo DE Base, todos os algoritmos encontraram hélices válidas para todos os v_S .

A Tabela 9 apresenta os melhores valores de P_B em kW entre dez execuções de cada algoritmo para v_S de 7,0, 7,5, 8,0 e 8,5 nós, com parâmetros da série "Wageningen B-Screw". O CMA-ES Modificado obteve consistentemente o melhor resultado para a maioria dos v_S , enquanto o OpenAI-ES Modificado alcançou o melhor para a velocidade mais alta. Contudo, na Tabela 9, todas as configurações apresentam valores similares de D e Z , principalmente entre as mesmas velocidades, com pouca variação nos valores de P/D e A_E/A_O entre os algoritmos.

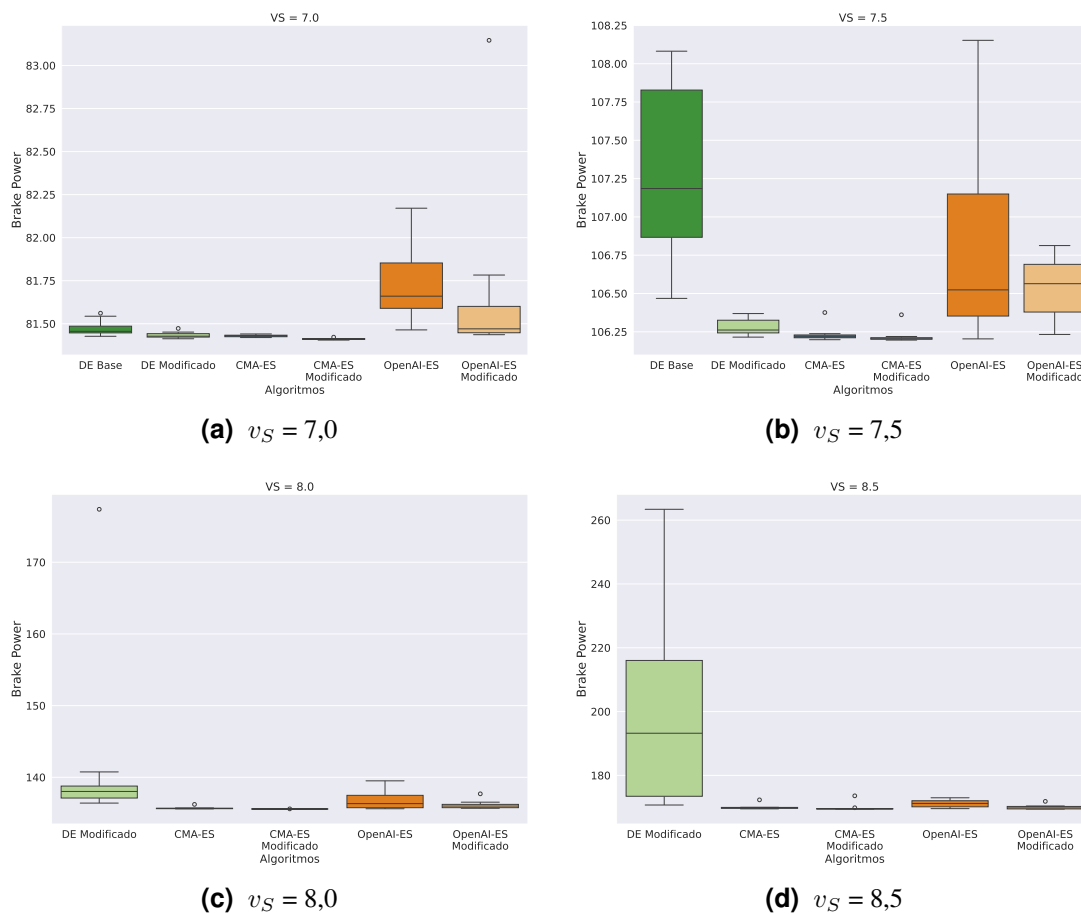


Figura 7. Valores encontrados pelas execuções.

7.4. Convergência na Fitness

A Figura 8 exibe a evolução da fitness nos melhores resultados de cada geração para os algoritmos listados na Tabela 9. Observa-se que os algoritmos CMA-ES e OpenAI-ES não modificados começam com os piores resultados nas primeiras gerações devido ao menor número de indivíduos na população. No entanto, o DE Modificado requer consideravelmente mais gerações para convergir em comparação aos outros algoritmos.

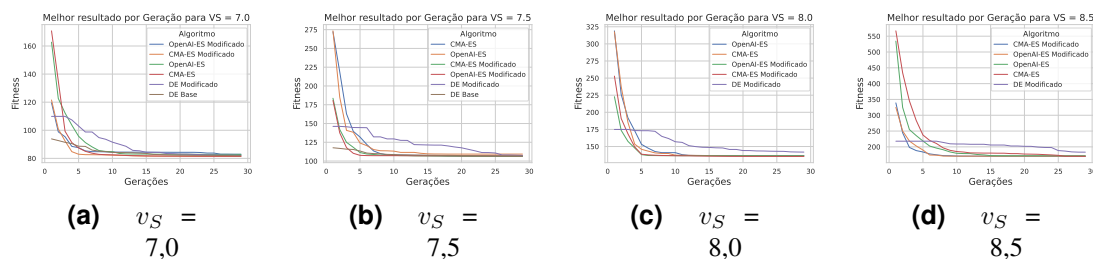


Figura 8. Melhor Fitness encontrada por geração.

Tabela 9. Parâmetros dos melhores resultados encontrados pelas heurísticas para todos os valores de v_S .

| v_S (nós) | Algoritmo | P_B | D | A_E/A_O | P/D | Z |
|-------------|----------------------|----------------|-------|-----------|-------|-----|
| 7,0 | DE Base | 81,427 | 0,8 | 0,641 | 0,675 | 5 |
| | DE Modificado | 81,413 | 0,8 | 0,632 | 0,669 | 5 |
| | CMA-ES | 81,419 | 0,8 | 0,631 | 0,667 | 5 |
| | CMA-ES Modificado | 81,405 | 0,8 | 0,631 | 0,670 | 5 |
| | OpenAI-ES | 81,464 | 0,8 | 0,640 | 0,658 | 5 |
| | OpenAI-ES Modificado | 81,436 | 0,8 | 0,642 | 0,675 | 5 |
| 7,5 | DE Base | 106,468 | 0,8 | 0,717 | 0,641 | 5 |
| | DE Modificado | 106,215 | 0,8 | 0,699 | 0,653 | 5 |
| | CMA-ES | 106,198 | 0,8 | 0,700 | 0,657 | 5 |
| | CMA-ES Modificado | 106,194 | 0,8 | 0,704 | 0,663 | 5 |
| | OpenAI-ES | 106,204 | 0,8 | 0,704 | 0,660 | 5 |
| | OpenAI-ES Modificado | 106,233 | 0,8 | 0,709 | 0,664 | 5 |
| 8,0 | DE Base | não encontrou | - | - | - | - |
| | DE Modificado | 136,402 | 0,8 | 0,881 | 0,743 | 6 |
| | CMA-ES | 135,624 | 0,8 | 0,791 | 0,667 | 5 |
| | CMA-ES Modificado | 135,597 | 0,8 | 0,809 | 0,680 | 6 |
| | OpenAI-ES | 135,612 | 0,8 | 0,788 | 0,656 | 5 |
| | OpenAI-ES Modificado | 135,683 | 0,8 | 0,823 | 0,689 | 6 |
| 8,5 | DE Base | não encontrou | - | - | - | - |
| | DE Modificado | 170,711 | 0,8 | 0,962 | 0,701 | 6 |
| | CMA-ES | 169,529 | 0,799 | 0,891 | 0,669 | 6 |
| | CMA-ES Modificado | 169,522 | 0,882 | 0,895 | 0,655 | 6 |
| | OpenAI-ES | 169,573 | 0,8 | 0,877 | 0,646 | 6 |
| | OpenAI-ES Modificado | 169,446 | 0,8 | 0,981 | 0,670 | 6 |

7.5. Quantidade de indivíduos válidos por geração

A Figura 9 ilustra a evolução da quantidade de soluções válidas ao longo das gerações para diferentes algoritmos, considerando variações superiores a 1% nos parâmetros entre indivíduos de uma mesma geração. Para CMA-ES e OpenAI-ES, os resultados dos cenários de Z distintos foram somados. Isso implica que CMA-ES Modificado e OpenAI-ES Modificado têm potencial para atingir até 60 soluções válidas e distintas em uma mesma geração, devido ao dobro do tamanho da população em comparação com outros algoritmos, que podem alcançar no máximo 30 soluções em uma geração.

Observa-se, na Figura 9, que os algoritmos iniciam com poucas soluções válidas, aumentando à medida que as gerações progridem e eventualmente estabilizando. O tempo para atingir a estabilização varia principalmente com a v_S e entre os algoritmos. Assim, é desejável algoritmos que encontrem rapidamente soluções válidas. As Figuras 9c e 9d destacam os valores de v_S não triviais, nos quais o DE Base não encontrou soluções válidas, e exibem menores quantidades de soluções encontradas em comparação com outros algoritmos.

Ao analisar a Figura 9, nota-se que CMA-ES Modificado e OpenAI-ES Modifi-

cado alcançam aproximadamente o dobro de soluções candidatas em comparação com versões com metade do tamanho da população. Entretanto, ambas as implementações de CMA-ES apresentam uma queda na quantidade de soluções diferentes nas gerações finais.

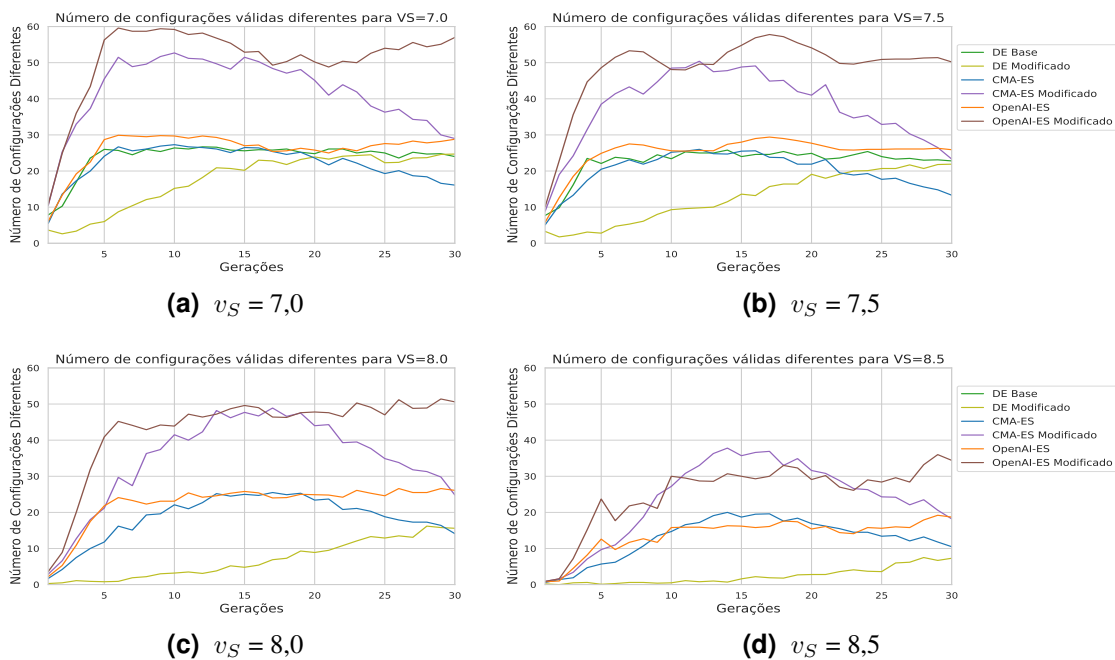


Figura 9. Número de configurações válidas diferentes por geração.

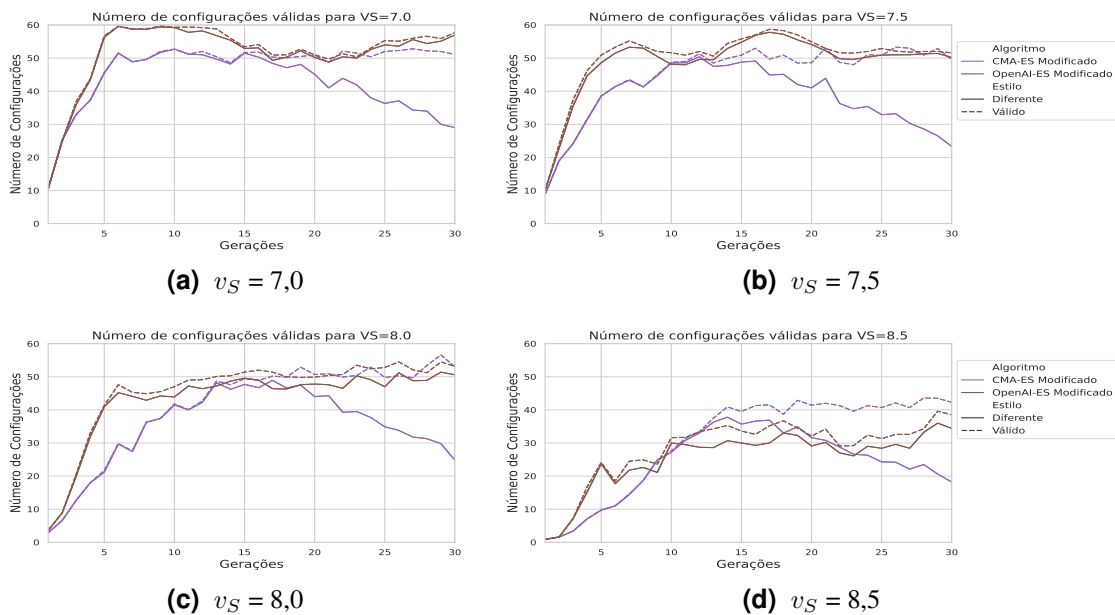


Figura 10. Linhas contínuas representam número de configurações válidas diferentes por geração, enquanto que as linhas tracejadas representam as válidas podendo ser repetidas por geração.

A Figura 10 destaca as configurações diferentes (linhas contínuas) e a quantidade total de soluções válidas (linhas tracejadas), indicando que os algoritmos continuam a encontrar quantidades semelhantes de resultados válidos, sugerindo convergência para soluções comuns. A diminuição na área de busca indica que um aumento na quantidade máxima de gerações provavelmente não impactaria significativamente os resultados, dado que os algoritmos estão convergindo.

8. Visualização do resultado

Com base nos melhores parâmetros encontrados, foram geradas imagens representando a possível geometria das hélices. As Figuras 11a e 11b mostram a geometria da hélice de frente e de lado, respectivamente, para os parâmetros obtidos pelo CMA-ES Modificado na v_S de 7,0 nós (conforme Tabela 9). Já as Figuras 11c e 11d apresentam a geometria correspondente para uma v_S de 7,5 nós, também encontradas pelo CMA-ES Modificado (conforme Tabela 9).

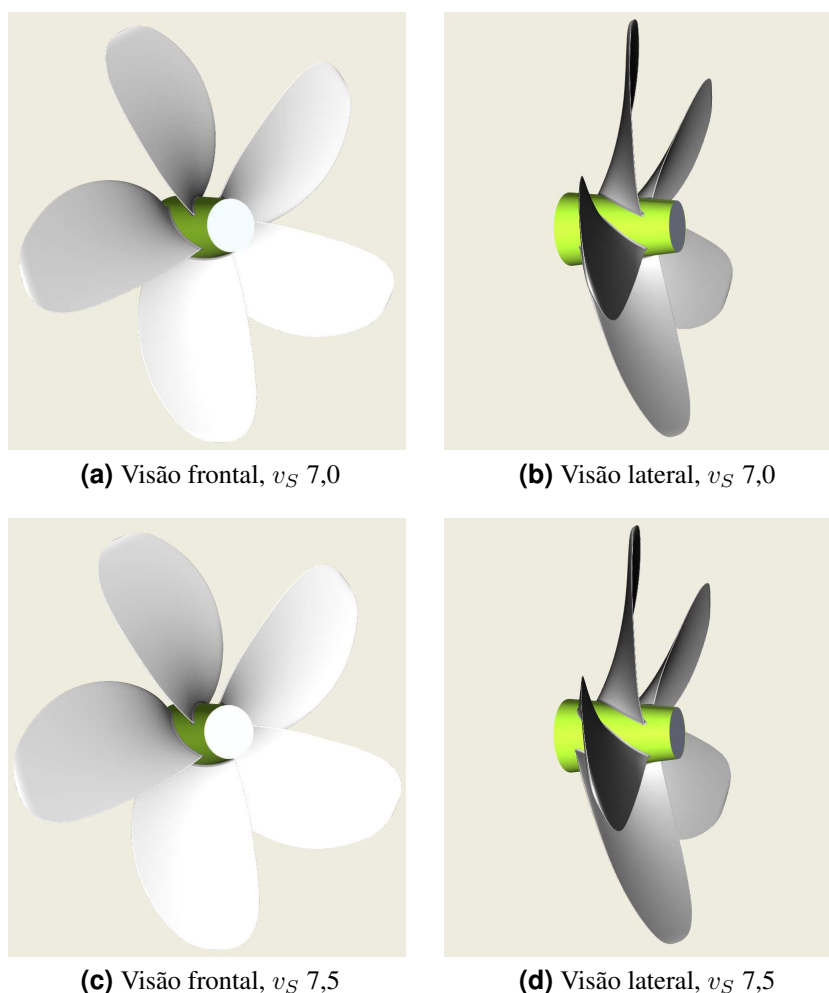


Figura 11. Representações da Geometria 3D da Hélice.

O software de geração de geometria 3D utilizado foi o CAESES®, fornecido gratuitamente pela Friendship Systems AG e disponível em [<https://www.wageningen-b-series-propeller.com>] . Não foi possível criar imagens para todas as velocidades devido

a limitações do software. No entanto, dada a proximidade dos parâmetros mesmo para diferentes valores de v_S , é razoável inferir que a geometria gerada seria semelhante à apresentada neste trabalho.

9. Conclusão

Este trabalho buscou melhorar hélices para drones híbridos em ambientes aquáticos, focalizando na eficiência energética por meio da aplicação de heurísticas nos parâmetros de morfologia da hélice. Experimentos compararam algoritmos, destacando Estratégias Evolutivas (CMA-ES e OpenAI-ES), superando abordagens convencionais. Ajustes de hiperparâmetros em AEs foram explorados para aprimorar os resultados.

Resultados indicaram a eficácia de Estratégias Evolutivas na descoberta de configurações de hélices eficientes. A introdução de um novo cálculo de fitness permitiu rápida convergência e exploração mais ampla de soluções candidatas, especialmente em maiores valores de v_S . CMA-ES com 10 indivíduos obteve os melhores resultados para v_S 7,0, 7,5 e 8,0 nós, enquanto OpenAI-ES atingiu o máximo desempenho a 8,5 nós.

O registro de parâmetros e restrições permitiu análises comparativas entre algoritmos. Sugere-se para futuras pesquisas a incorporação de métricas de drones reais nas restrições para aproximar as soluções dos designs reais. Além disso, a expansão do escopo para hélices operando em ambientes aéreos, focando em drones híbridos como proposta. Assim como validar testes teóricos com experimentos práticos e ajustar iterativamente, para assegurar resultados consistentes entre teoria e prática no desempenho de hélices reais.

9.1. Código Fonte

O código fonte utilizado para experimentos, assim como os resultados obtidos e imagens geradas estão disponíveis publicamente no link: <https://github.com/pauloBFalmeida/hydrone>. Assim como arquivos "README.md" com explicações do funcionamento do código e como replicar os experimentos realizados neste trabalho.

Referências

- Adby, P. (2013). *Introduction to Optimization Methods*. Chapman and Hall Mathematics Series. Springer Netherlands.
- Anderssen, R. S. and Bloomfield, P. (1975). Properties of the random search in global optimization. *Journal of Optimization Theory and Applications*, 16(5):383–398.
- Ayamga, M., Akaba, S., and Nyaaba, A. A. (2021). Multifaceted applicability of drones: A review. *Technological Forecasting and Social Change*, 167:120677.
- Bartz-Beielstein, T., Branke, J., Mehnen, J., and Mersmann, O. (2014). Evolutionary algorithms. *WIREs Data Mining and Knowledge Discovery*, 4(3):178–195.
- Benini, E. (2003). Multiobjective Design Optimization of B-Screw Series Propellers Using Evolutionary Algorithms. *Marine Technology and SNAME News*, 40(04):229–238.

- Carlton, J. (2007). 19 - propeller blade strength. In Carlton, J., editor, *Marine Propellers and Propulsion (Second Edition)*, pages 396–415. Butterworth-Heinemann, Oxford, second edition edition.
- Carlton, J. (2019). Chapter 12 - resistance and propulsion. In Carlton, J., editor, *Marine Propellers and Propulsion (Fourth Edition)*, pages 313–365. Butterworth-Heinemann, Oxford, UK, fourth edition edition.
- Caupin, F. and Herbert, E. (2006). Cavitation in water: a review. *Comptes Rendus Physique*, 7(9):1000–1017. Nucleation.
- Cortez, P. (1997). Algoritmos genéticos e redes neuronais na previsão de séries temporais.
- Cui, W. (2019). An overview on aquatic unmanned aerial vehicles. *Annals of Reviews & Research*.
- Currison, D., Hartfield, R., and Burkhalter, J. (2013). Power turbine blade aerodynamic optimization using non-restrictive evolution strategies.
- Das, S. and Suganthan, P. N. (2011). Differential evolution: A survey of the state-of-the-art. *IEEE Transactions on Evolutionary Computation*.
- Dinelli, C., Racette, J., Escarcega, M., Lotero, S., Gordon, J., Montoya, J., Dunaway, C., Androulakis, V., Khaniani, H., Shao, S., Roghanchi, P., and Hassanalain, M. (2023). Configurations and applications of multi-agent hybrid drone/unmanned ground vehicle for underground environments: A review. *Drones*, 7(2).
- Doerr, B. and Neumann, F. (2019). *Theory of Evolutionary Computation: Recent Developments in Discrete Optimization*. Natural Computing Series. Springer International Publishing.
- Doijode, P. S., Hickel, S., van Terwisga, T., and Visser, K. (2022). A machine learning approach for propeller design and optimization: Part i. *Applied Ocean Research*, 124.
- Drews-Jr, P., Neto, A., and Campos, M. (2009). A survey on aerial submersible vehicles.
- Duan, X. (2022). *Multidisciplinary Design Optimization of Coaxial Drone Propellers*. PhD thesis.
- Eiben, A. E. and Smith, J. E. (2015). *What Is an Evolutionary Algorithm?*, pages 25–48. Springer Berlin Heidelberg, Berlin, Heidelberg.
- Friedman, J., Hastie, T., Höfling, H., and Tibshirani, R. (2007). Pathwise coordinate optimization. *The Annals of Applied Statistics*, 1(2):302 – 332.
- Friendship Systems AG (2023). B-series propeller generator. <https://www.wageningen-b-series-propeller.com/>. Acesso em: 26 de outubro de 2023, 20:15.
- Geweke, J. (1988). Antithetic acceleration of monte carlo integration in bayesian inference. *Journal of Econometrics*, 38(1):73–89.
- Ha, D. (2017a). Evolving stable strategies. *blog.otoro.net*. Acesso em: 15 de junho de 2023, 15:30.
- Ha, D. (2017b). A visual guide to evolution strategies. *blog.otoro.net*. Acesso em: 15 de junho de 2023, 16:45.

- Hansen, N. (2014). CMA-ES: A Function Value Free Second Order Optimization Method. In *PGMO COPI 2014 - Gaspard Monge Program for Optimization and operations research Conference on Optimization and Practices in Industry*, Saclay, France.
- Hansen, N. (2023). The cma evolution strategy: A tutorial.
- Hansen, N., Arnold, D. V., and Auger, A. (2015). *Evolution Strategies*, pages 871–898. Springer Berlin Heidelberg, Berlin, Heidelberg.
- Hansen, N. and Ostermeier, A. (1996). Adapting arbitrary normal mutation distributions in evolution strategies: The covariance matrix adaptation. pages 312 – 317.
- Helma, S. (2020). Surprising behaviour of the wageningen b-screw series polynomials. *Journal of Marine Science and Engineering*, 6(211):1–63.
- Holden, K. O., Fagerjord, O., and Frostad, R. (1980). Early design-stage approach to reducing hull surface forces due to propeller cavitation. *Society of Naval Architects and Marine Engineers*, 14(00323480).
- Irizarry, J., Gheisari, M., and Walker, B. N. (2012). Usability assessment of drone technology as safety inspection tools. *J. Inf. Technol. Constr.*, 17:194–212.
- Karnopp, D. C. (1963). Random search techniques for optimization problems. *Automatica*, 1(2):111–121.
- Kasno, M. A., Mohd Aras, M. S., Keong, Y. E., Ab Azis, F., Mohamed Kassim, A., Harun, M. H., Mohd Zambri, M. K., and Khamis, A. (2017). Design analysis of remotely amphibian vehicle (rav) –underwater drone. *Journal of Telecommunication, Electronic and Computer Engineering (JTEC)*, 9(3-2):35–38.
- Klimczyk, W. A. (2022). Aerodynamic design and optimization of propellers for multi-rotor. *Aircraft Engineering and Aerospace Technology*, 94(1):21–30.
- Land, M. W. S. (1998). *Evolutionary algorithms with local search for combinatorial optimization*. University of California, San Diego.
- Li, Y. and Osher, S. (2009). Coordinate descent optimization for l1 minimization with application to compressed sensing; a greedy algorithm. *Inverse Problems and Imaging*, 3(3):487–503.
- Malapur, H., Singh, Y., Shendkar, M., Barve, A., and Bedekar, M. (2022). Diffused casing of drone propeller for reduced operational noise and optimized energy consumption. *Materials Today: Proceedings*, 63:136–140. 4th International Conference on Advances in Mechanical Engineering and Nanotechnology.
- Marques, C. H., Belchior, C. R. P., and Caprace, J.-D. (2019). Marine propeller parametric optimisation and matching to electric motor. *Journal of the Brazilian Society of Mechanical Sciences and Engineering*, 41(3):119.
- Marques, C. H., Ordonez, J. C., Souza, J. A., and Caprace, J.-D. (2023). Genetic optimization of propeller-motor matching for all-electric ships. In *2023 IEEE Electric Ship Technologies Symposium (ESTS)*, pages 129–138.

- Melo, M. G. d. (2019). *Adaptação local da matriz de covariância guiada por mecanismos de exploração em estratégias de evolução*. PhD thesis, Universidade Federal de Pernambuco.
- Mohd Daud, S. M. S., Mohd Yusof, M. Y. P., Heo, C. C., Khoo, L. S., Chainchel Singh, M. K., Mahmood, M. S., and Nawawi, H. (2022). Applications of drone in disaster management: A scoping review. *Science & Justice*, 62(1):30–42.
- Oosterveld, M. W. C. and van Oossanen, P. (1975). Further computer-analyzed data of the wageningen b-screw series. *International shipbuilding progress*, 22(251):251–262.
- Pacheco, M. A. C. et al. (1999). Algoritmos genéticos: Princípios e aplicações. *ICA: Laboratório de Inteligência Computacional Aplicada. Departamento de Engenharia Elétrica. Pontifícia Universidade Católica do Rio de Janeiro. Fonte desconhecida*, 28.
- Perez, R., Liu, H., and Behdinan, K. (2004). Evaluation of multidisciplinary optimization approaches for aircraft conceptual design. *Collection of Technical Papers - 10th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference*, 4.
- Porumbel, D. C., Hao, J.-K., and Kuntz, P. (2010). A search space “cartography” for guiding graph coloring heuristics. *Computers & Operations Research*, 37(4):769–778.
- Price, W. L. (1983). Global optimization by controlled random search. *Journal of Optimization Theory and Applications*, 40(3):333–348.
- Roy, B. (1971). Problems and methods with multiple objective functions. *Mathematical Programming*, 1(1):239–266.
- Salimans, T., Ho, J., Chen, X., Sidor, S., and Sutskever, I. (2017). Evolution Strategies as a Scalable Alternative to Reinforcement Learning.
- Slavík, P. (1997). Improved performance of the greedy algorithm for partial cover. *Information Processing Letters*, 64(5):251–254.
- Slowik, A. and Kwasnicka, H. (2020). Evolutionary algorithms and their applications to engineering problems. *Neural Computing and Applications*, 32(16):12363–12379.
- Stapersma, D. and Woud, H. (2005). Matching propulsion engine with propulsor. *Journal of Marine Engineering & Technology*, 4(2):25–32.
- Storn, R. and Price, K. (1997a). Differential evolution - a simple and efficient heuristic for global optimization over continuous spaces. *Journal of Global Optimization*, 11(4):341–359.
- Storn, R. and Price, K. (1997b). Differential evolution – a simple and efficient heuristic for global optimization over continuous spaces. *Journal of Global Optimization*.
- van Lammeren, W., van Manen, J., and Oosterveld, M. (1969). The wageningen b-screw series.
- Voß, S., Martello, S., Osman, I. H., and Roucairol, C. (2012). *Meta-heuristics: Advances and trends in local search paradigms for optimization*. Springer Science & Business Media.
- Wang, F.-S. and Chen, L.-H. (2013). *Heuristic Optimization*, pages 885–885. Springer New York, New York, NY.

- Webb, A., Eames, C., and Tuffrey, A. (1975). Factors affecting design stresses in marine propellers.
- Wierstra, D., Schaul, T., Peters, J., and Schmidhuber, J. (2008). Natural evolution strategies. In *2008 IEEE Congress on Evolutionary Computation, CEC 2008*, pages 3381–3387. Generated from Scopus record by KAUST IRTS on 2022-09-14.
- Woud, H. K. and Stapersma, D. (2013). *Design of Propulsion and Electric Power Generation Systems*. IMarEST, London, UK.